DEVELOPMENT OF A DIGITAL TERRAIN SIMULATOR
FOR SHORT-TERM
FOREST RESOURCE PLANNING

by

DANIEL ZWI LEMKOW
B.S.F, University of British Columbia, 1975

A THESIS SUBMITTED IN PARTIAL PULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(The Faculty of Forestry)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
July, 1977
© Daniel Zwi Lemkow, 1977
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Forestry

The University of British Columbia
2075 Westbrook Place
Vancouver, Canada
V6T 1W5

Date July 29, 1977
Abstract

This study describes the development of a desktop computer model (digital terrain simulator) for short-term forest planning.

An overview of the applications currently developed for the terrain simulator is presented:

1) Collection of the required terrain elevations and forest inventory data;
2) Determination and production of map overlays of the topographic features: slope, aspect and elevation;
3) Design of logging settings for cable systems and placement of yarding roads;
4) Location of forest-access roads;
5) Delineation of viewable areas and production of three dimensional representations of the terrain on a two dimensional surface;
6) Estimation of harvesting costs and wood volume production.

The study then presents the theory required to implement each of the above components. Where possible, several different
approaches are developed and compared.

The elevation data base, describing the study area, is represented by a regular grid of elevations. The slope, aspect and elevation are computed for each of these grid units from a geometric plane fitted to the ground surface using a least-squares procedure. An algorithm for producing map overlays of these attributes, in varying combinations, is given.

The major emphasis in the discussion of the setting design module is on the prediction of loadpaths of cable yarding systems. The road design module concentrates on route projection. An algorithm is proposed that automatically locates a trial route between any two map locations.

For producing three dimensional representations both orthographic and perspective projections are developed along with an algorithm to remove the 'hidden areas' in these three dimensional plots.

The costing module presents methods for input of forest inventory data, using the same regular grid as for the terrain elevations. An algorithm deterministically simulates the yarding of the wood in each grid unit contained by the logging setting boundaries.

Finally, the limitations of the model, due to computer technology and the quality and quantity of the data, are
examined. Slow execution speed, in several instances, dictated the use of the least accurate approach; this occurred in the input, yarding design and costing modules. This also made it necessary to use orthographic instead of the true perspective projection for producing three dimensional representations. The determination of earthwork volumes, for road construction, was deemed inappropriate, due to the generally poor precision of available maps. Soils information, although important for estimating costs of yarding and road construction, was not included because of its limited availability. The map overlays of slope, aspect and elevation could be easily extended to include soils and land-use data, thereby offering a more complete and useful retrieval system.

Although full implementation of the terrain simulator has not been possible, this study demonstrates the feasibility of implementing a comprehensive short-term forest planning model, designed for a desktop computer.
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>List of Plates</td>
<td>xiv</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>xv</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Part I : An Overview of the Terrain Simulator</td>
<td></td>
</tr>
<tr>
<td>and Its Applications</td>
<td></td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1.0 Initial Preparation</td>
<td>8</td>
</tr>
<tr>
<td>2.0 Input of the Elevation Data Base</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Transect Method</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Contour Method</td>
<td>14</td>
</tr>
<tr>
<td>3.0 Analysis of Topographic Features</td>
<td>16</td>
</tr>
<tr>
<td>4.0 Yarding Location and Setting Design</td>
<td>20</td>
</tr>
<tr>
<td>5.0 Road Location</td>
<td>28</td>
</tr>
<tr>
<td>6.0 Aesthetic Analysis</td>
<td>33</td>
</tr>
<tr>
<td>6.1 Viewable Area</td>
<td>33</td>
</tr>
<tr>
<td>6.2 Three Dimensional Representations</td>
<td>33</td>
</tr>
<tr>
<td>7.0 Harvesting Costs and Wood Volume Production</td>
<td>40</td>
</tr>
<tr>
<td>Part II : Terrain Simulator Theory</td>
<td>45</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1.0 Elevation Data Base</td>
<td>46</td>
</tr>
<tr>
<td>1.1 Retention of Contour Locations</td>
<td>47</td>
</tr>
<tr>
<td>1.2 Conversion of Contours into a Regular</td>
<td></td>
</tr>
<tr>
<td>Grid of Elevations</td>
<td>49</td>
</tr>
</tbody>
</table>
1.2.1 Interpolation Method for Finding Elevations .......... 52

1.2.2 Surface Equation Method For Finding Elevations .......... 61

1.3 Implementation ........................................ 64

2.0 Input of the Elevation Data Base ............................. 67

2.1 Transect Method ........................................... 67

2.2 Contour Method ............................................. 70

2.2.1 Entry of Contour Lines ................................. 71

2.2.2 Sorting of Data Points to Facilitate Searching .......... 71

2.2.3 Generating the Grid of Elevations ...................... 76

2.3 Comparison of the Transect and Contour Methods .............. 80

2.4 Implementation ............................................. 82

2.5 A New Method for Data Entry ................................ 83

3.0 Topographic Features ......................................... 84

3.1 Determining the Topographic Features ......................... 85

3.1.1 Solving for Average Elevation ......................... 85

3.1.2 Solving for the Maximum Surface Slope .................. 85

3.1.3 Solving for Aspect ...................................... 87

3.2 Storage of the Slope, Aspect and Elevation ................... 91

3.3 Output of the Topographic Features ......................... 91

3.4 Implementation ............................................. 92
CHAPTER PAGE

4.0 Yarding Location and Setting Design ................. 95

4.1 Parabolic Model for Approximating
the Loadpath ........................................ 96

4.1.1 Comparison to the Catenary Model .......... 102

4.1.2 Improvement of the
Iterative Method ...................................... 102

4.2 Percent Deflection Rule for Approximating
the Loadpath ........................................ 105

4.3 Implementation ....................................... 111

5.0 Road Location ...................................... 114

5.1 Manual Road Projection ............................ 114

5.2 Automatic Road Projection ......................... 115

5.3 Road Location Output ............................... 120

5.4 Grade Line Location and
Earthwork Volumes .................................. 120

5.5 Implementation ....................................... 120

6.0 Aesthetic Analysis .................................. 122

6.1 Viewable Area Assessment ......................... 122

6.1.1 Determining the Viewable Cells
on a Sight Line ....................................... 122

6.1.2 Implementation ................................... 126

6.2 Three Dimensional Representations of the
Elevation Data Base ................................. 127

6.2.1 Transforming the Elevation
Data Base ........................................... 128

6.2.2 Plotting the Three Dimensional
Representation ........................................ 136
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.3 Plotting the Locations of Planometric Details onto the 3D Plot</td>
<td>143</td>
</tr>
<tr>
<td>6.2.4 Implementation</td>
<td>146</td>
</tr>
<tr>
<td>7.0 Harvesting Costs and Wood Volume Production</td>
<td>147</td>
</tr>
<tr>
<td>7.1 Creation of the Wood Volume and Soils Data Base</td>
<td>147</td>
</tr>
<tr>
<td>7.2 Setting Analysis</td>
<td>149</td>
</tr>
<tr>
<td>7.3 Road Costing</td>
<td>151</td>
</tr>
<tr>
<td>7.4 Implementation</td>
<td>152</td>
</tr>
<tr>
<td>Discussion</td>
<td>153</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Economic Justification of the Terrain Simulator</td>
<td>153</td>
</tr>
<tr>
<td>2.0 Additions to the Current Model</td>
<td>156</td>
</tr>
<tr>
<td>2.1 A Hydrology Module</td>
<td>156</td>
</tr>
<tr>
<td>2.2 A Graphical Information Retrieval Module</td>
<td>157</td>
</tr>
<tr>
<td>2.3 Summary</td>
<td>158</td>
</tr>
<tr>
<td>Conclusion</td>
<td>160</td>
</tr>
<tr>
<td>Literature Cited</td>
<td>168</td>
</tr>
<tr>
<td>Appendices</td>
<td>170</td>
</tr>
<tr>
<td>Appendix A. Point-in-Polygon Algorithm</td>
<td>171</td>
</tr>
<tr>
<td>Appendix B. Menu Operation</td>
<td>175</td>
</tr>
<tr>
<td>Appendix C. Cable Mechanics Theory</td>
<td>178</td>
</tr>
<tr>
<td>Appendix D. Perspective Plotting</td>
<td>203</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. System Specifications of the Hewlett-Packard 9830A Computing System</td>
<td>6</td>
</tr>
<tr>
<td>II. Coordinates and Elevations of the Four Cell Corners</td>
<td>52</td>
</tr>
<tr>
<td>III. Comparison of the Elevation Errors from Maps of Different Sizes</td>
<td>59</td>
</tr>
<tr>
<td>IV. Comparison of Operation Counts</td>
<td>60</td>
</tr>
<tr>
<td>V. Relative Execution Times</td>
<td>60</td>
</tr>
<tr>
<td>VI. Definition of Aspect Indices</td>
<td>90</td>
</tr>
<tr>
<td>VII. Errors from the Parabolic Model with the Carriage at Midspan for a Madill 052 Tension Skidder</td>
<td>103</td>
</tr>
<tr>
<td>VIII. Difference Table for Sag (y)</td>
<td>104</td>
</tr>
<tr>
<td>IX. Deflection at Midspan for a Madill 052 Tension Skidder (Chord Definition)</td>
<td>107</td>
</tr>
<tr>
<td>X. Deflection at Midspan for a Madill 052 Tension Skidder (Span Definition)</td>
<td>109</td>
</tr>
<tr>
<td>XI. Percent Rule Error at Quarter Span for a Madill 052 Tension Skidder</td>
<td>112</td>
</tr>
<tr>
<td>XII. Line Tensions with the Carriage at Midspan for a Madill 052 Tension Skidder using the Parabolic Model</td>
<td>195</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Map of the Study Area with the Defining Rectangle</td>
</tr>
<tr>
<td>2</td>
<td>Entering the Contour Locations for a Transect Line</td>
</tr>
<tr>
<td>3</td>
<td>Menu for the Transect Method of Entering Contours</td>
</tr>
<tr>
<td>4</td>
<td>Menu for the Contour Method of Entering Contours</td>
</tr>
<tr>
<td>5</td>
<td>Map Overlay for the 40 Percent Slope Classes</td>
</tr>
<tr>
<td>6</td>
<td>Combination Overlay for Areas Between 40 and 79% slope, 500 and 999 Feet Elevation and Southeast and Southwest Aspects</td>
</tr>
<tr>
<td>7</td>
<td>A Cable Yarding System</td>
</tr>
<tr>
<td>8</td>
<td>Plot of Ground Profile</td>
</tr>
<tr>
<td>9</td>
<td>Ground Profile and Loadpath Plots for Projected Yarding Roads</td>
</tr>
<tr>
<td>10</td>
<td>Plan View of Projected Yarding Roads</td>
</tr>
<tr>
<td>11</td>
<td>Plot of Ground Profile and Loadpath for a Single Yarding Road</td>
</tr>
<tr>
<td>12</td>
<td>Plan View Plot of Road Network</td>
</tr>
<tr>
<td>13</td>
<td>Road Profiles</td>
</tr>
<tr>
<td>14</td>
<td>Road Analysis Output</td>
</tr>
<tr>
<td>15</td>
<td>Map Overlay for Viewable Areas</td>
</tr>
<tr>
<td>16</td>
<td>Determining the Initial Viewing Angle</td>
</tr>
<tr>
<td>17</td>
<td>Orthogonal Plot of Cypress Bowl Provincial Park (looking north)</td>
</tr>
<tr>
<td>FIGURE</td>
<td>PAGE</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>18</td>
<td>Planametric Map Showing the Hidden Areas for the Orthogonal Projection of Cypress Bowl Provincial Park</td>
</tr>
<tr>
<td>19</td>
<td>Map of Study Area Showing Covertype Boundaries and Associated Grid Overlay</td>
</tr>
<tr>
<td>20</td>
<td>Covertype Map and Associated Species Listing</td>
</tr>
<tr>
<td>21</td>
<td>Costing Analyses for Roads and Settings</td>
</tr>
<tr>
<td>22</td>
<td>Definition of Coordinate System</td>
</tr>
<tr>
<td>23</td>
<td>Computing the Elevation for the Point ((x,y)) Using Three-step Linear Interpolation and Inverse Distance-squared Interpolation</td>
</tr>
<tr>
<td>24</td>
<td>Accuracy Test Comparing the Three-step Linear and Inverse Distance-squared Techniques for Computing Elevations</td>
</tr>
<tr>
<td>25</td>
<td>Graph of the Elevation Error as a Function of Map Area</td>
</tr>
<tr>
<td>26</td>
<td>Accuracy Test for the Quadratic Interpolation Technique</td>
</tr>
<tr>
<td>27</td>
<td>Double Transect Method for Generating Elevations</td>
</tr>
<tr>
<td>28</td>
<td>Relationship of Bands, Contour Points and Grid Points</td>
</tr>
<tr>
<td>29</td>
<td>Using Four Regions to find an Elevation</td>
</tr>
<tr>
<td>30</td>
<td>Comparison of the Accuracy of the Contour and Transect Methods</td>
</tr>
<tr>
<td>31</td>
<td>The Surface Plane and its Coordinate System</td>
</tr>
<tr>
<td>32</td>
<td>Definition of the Aspect Classes</td>
</tr>
</tbody>
</table>
FIGURE

33 Flowchart for Plotting Map Overlays .................. 93
34 Schematic Drawing of a Running Skyline System .......... 97
35 Loadpath Solution for a Running Skyline System
   Using the Parabolic Model and Newton Iteration .......... 101
36 Using an Interpolating Polynomial to
   Predict the Loadpath ................................ 106
37 Planometric View of 'Automatic' Road Location .......... 117
38 The Effect of the Controlling Parameters
   on 'Automatic' Road Location ........................ 119
39 Locating Grid Cells Along the Sight Line ............. 124
40 Determining the Viewable Area Along a Sight Line ...... 125
41 Rotation of One Box of the Elevation Matrix .......... 129
42 Flowchart for the Rotation of the Elevation Matrix ... 130
43 Viewing Controls for Orthographic Projection .......... 133
44 Orthographic Projection using One Set of Profile
   Lines and an Angle of Rotation of 0 degrees .......... 137
45a Orthographic Projection using Two Sets of Profile
   Lines and an Angle of Rotation of 0 degrees .......... 138
45b Orthographic Projection using Two Sets of Profile
   Lines and an Angle of Rotation of 45 degrees ....... 139
45c Orthographic Projection using Two Sets of Profile
   Lines and an Angle of Rotation of 80 degrees ....... 140
46 Removing Hidden Areas .............................. 142
47 Determining Whether a Point is Within A Polygon .... 172
<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>173</td>
</tr>
<tr>
<td>49</td>
<td>176</td>
</tr>
<tr>
<td>50</td>
<td>180</td>
</tr>
<tr>
<td>51</td>
<td>181</td>
</tr>
<tr>
<td>52</td>
<td>182</td>
</tr>
<tr>
<td>53</td>
<td>184</td>
</tr>
<tr>
<td>54</td>
<td>185</td>
</tr>
<tr>
<td>55</td>
<td>187</td>
</tr>
<tr>
<td>56</td>
<td>189</td>
</tr>
<tr>
<td>57</td>
<td>191</td>
</tr>
<tr>
<td>58</td>
<td>200</td>
</tr>
<tr>
<td>59</td>
<td>204</td>
</tr>
<tr>
<td>60</td>
<td>206</td>
</tr>
</tbody>
</table>
**LIST OF PLATES**

<table>
<thead>
<tr>
<th>PLATE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The Hewlett-Packard 9830A Computing System</td>
</tr>
<tr>
<td>2.</td>
<td>Digitizing a Transect Line</td>
</tr>
<tr>
<td>3.</td>
<td>Cypress Bowl Provincial Park (looking north)</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

I am greatly indebted to Mr. G.G. Young, my supervisor, for providing his invaluable assistance for all aspects of my graduate program.

My gratitude is also extended to the National Research Council of Canada for generously providing financial support.

I would like to thank Dr. P.L. Cottell and Dr. D.H. Williams for being on my committee and reviewing the text.

Finally, I wish to thank Dr. D. Reimer and Mr. J. Marlow of MacMillan Bloedel LTD. for the opportunity to test the Digital Terrain Simulator in an operational environment. I would also like to express my gratitude to the staff, especially Mr. A. Kelly, of the Company's Franklin River Logging Division for their cooperation and valuable suggestions.
Introduction

Computer models for forest resource planning have been available since the early 1960's. During the past few years their use has accelerated due, in part, to increasing resource conflicts and wood harvesting costs. The major emphasis in their development has been towards long-range planning. Consequently there has been an absence of suitable models designed for operational (short-term) forest planning. It is the objective of this thesis to develop a computer model for this important planning phase.

The long range models have commonly been used for resource allocation and information retrieval. The earliest example of the latter is Amidon's (1966) graphical information system. There have been numerous allocation models developed. Among these are Maxmillion (Clutter, 1969), Timber RAM (Resource Allocation Method) (Navon, 1971) and CARP (Computer Assisted Resource Planning) by the British Columbia Forest Service (Williams, et al., 1975).

The lack of development work on short-term planning models was due partly to the limited computer technology available. Until recently most computers were large and required tightly controlled environments. Also, highly trained computer
specialists were needed to provide an interface between the user and the computer. These two factors precluded the high degree of computer interaction that a planner needed to efficiently devise a forest resource development plan. The small desktop computer developed over the last several years has brought the necessary computing power, with the required interactive capability, to the often remote geographical locations where the short-term planning is frequently done. This has made possible the development of practical short-term planning models.

The United States Forest Service (USFS), through the Pacific Northwest Forest and Range Experiment Station in Seattle, has pioneered development in this area. Separate programs have been produced to analyze skyline design and location (Carson, 1975), road design and location (Burke, 1974), and harvesting costs (Burke, 1976). These models are suitable for the actual on-site implementation of harvest planning. Their use in the analysis of the various alternatives in short-term planning, although quicker than the previous manual approaches, is still cumbersome and slow, as they do not use the full potential of the desktop computer. Map data describing the ground conditions must be entered into the computer for each new alternative to be tested, even if differing only slightly from the previous trial. This lack of a single data base, available for repeated use, is in the author's opinion, the biggest drawback for development planning. An additional problem is that each model is a separate package, thereby reducing their overall compatibility and efficiency.
The idea of using a single data base, defined prior to analysis, for operational planning is not new. Civil engineers have been utilizing this concept since the invention of the electronic computer (Meyer, 1969). This form of planning tool is commonly referred to as a digital terrain model (DTM). Burke (1974a) suggested their use for generating skyline profiles. The USFS, at the Pacific Southwest Forest and Range Experiment Station, has developed a computer package (for a large computing system) called VIEWIT (Travis et al., 1974) that permits assessment of viewable areas and delineation of slope and aspect classes. The program can produce overlay maps showing the various topographic features.

Individually the above planning systems address the major planning considerations. Unfortunately, because of their different data requirements and computer hardware, they do not represent an integrated approach.

The computer model described in this thesis is a TERRAIN SIMULATOR (another name for DTM) which was developed at the Faculty of Forestry of the University of British Columbia (UBC) (Young and Lemkow, 1976). The objective of the simulator was to provide a complete short-range planning package for a desktop computer. The model uses one data base to permit efficient analysis of the various facets of the planning process. The functions of the various planning modules in the system include:
a) Determination of the topographic features such as slope, aspect and elevation;

b) Design of settings and placement of yarding systems;

c) Location of roads;

d) Delineation of viewable area;

e) Estimation of harvesting costs and wood volume production.

The terrain simulator is described in two parts. The first section deals with the general use of the simulator; i.e. the various functions it can perform. No detailed attempt is made to discuss how the simulator would actually be used to develop a viable, comprehensive, short-term development plan. The second section describes the theoretical background for each of the modules.
Part I - An Overview of the Terrain Simulator And Its Applications

The terrain simulator has been developed on a Hewlett-Packard 9830A desktop computer. System specifications and the physical configuration are given in Table I and Plate 1, respectively. The digitizing unit (Plate 2) is the most important of the various peripheral devices. It permits the rapid entry and conversion of graphical information (e.g., roads and contour locations) into \((x, y)\) coordinates\(^1\) suitable for mathematical manipulation by the computer.

\(^1\) The units (Imperial or International System (S.I.)) for the coordinates are dependent on the digitizer. Consequently, when entering data from the digitizer into the terrain simulator program the scale must be specified in terms of digitizing units; e.g., feet per digitizing unit or meters per digitizing unit where the digitizing units could be in centimeters or inches. The computations, however, are independent of the units. For convenience all examples employ the Imperial System.
### TABLE I

**SYSTEM SPECIFICATIONS FOR THE HEWLETT-PACKARD 9830A COMPUTING SYSTEM**

**Calculator - Model 9830A**
- 16K bytes of read/write memory
- Matrix ROM (Read Only Memory)
- Extended I/O ROM
- Plotter Control ROM
- String Variable ROM
- Advanced Programing I ROM
- Advanced Programing II ROM

**Printer - Model 9866A**
- 80 characters per line
- 250 lines per minute
- thermal printing

**Plotter - Model 9862A**
- 10" by 15" plotting surface

**External Cassette Memory - Model 9865A**
- 130 feet/minute search speed
- maximum of 64K bytes storage capacity

**Digitizer - Model 9864A**
- table size of 36" by 48"
- table manufactured by Bendix
Plate 1. The Hewlett-Packard 9830A Computing System.

Plate 2. Digitizing a Transect Line
1.0 Initial Preparation

The simulator depends on one data base from which all analyses can proceed. For this, an elevation contour map is required. A forest covertype map, with species volumes, is also needed for the Costing and Wood Volume Production Module. The study area is delineated on the map by a rectangle (Figure 1). At the outset, the user must ensure that the size of the contained area is adequate for the proper development of his plan, since no additions can be made to the data base at a later stage.

The map precision will dictate the level of analysis that can be achieved. Generally, maps of scale 1 inch to 400 feet, with 25-foot contours, are adequate for all the planning modules. Maps with the same scale but 100-foot contours should be used with caution in skyline and road locations as they possess insufficient topographic detail. Maps with scales offering less detail (e.g., 20 chains to the inch) are useful only for topographic features and visual assessment, unless the terrain is very uniform.
Figure 1. Map of the Study Area with the Defining Rectangle and the 67 Transect Lines.
2.0 Input of the Elevation Data Base

There are at present two methods for obtaining the elevation data base:

(a) The Transect Method;
(b) The Contour Method.

The input of the forest covertype information is controlled by the Harvesting Costs and Wood Volume Production Module (Section 7).

2.1 Transect Method

For the transect method, the map is aligned on the digitizing table with the rectangle describing the area to be analyzed parallel to the table. With the digitizer, the user sets the origin of the \((x, y)\) cartesian coordinate system at the lower left corner of the rectangle. The upper right corner is then entered\(^2\) thereby defining the rectangle.

The user drafts 67 equally-spaced transect lines onto the rectangular map area (Figure 1). The user then specifies the contour interval and the initial contour elevation of the

\(^2\) All map points are entered from the digitizer. The cursor is placed over the point and the appropriate button pressed (Plate 2).
current transect line. The point of intersection is entered for each contour that crosses the line. The user must specify whether the next contour to be crossed is uphill, downhill or level with respect to the last one entered. This is accomplished by digitizing a point at least one inch to the left of the line for "up" (Figure 2) and conversely, digitizing a point to the right for "down". The level condition is indicated by an "up" and a "down". The entire process is controlled by a 'MENU' (Figure 3).

On completion of a line, the computer interpolates between the digitized contour crossings to determine the elevation of 67 equally-spaced points along the line. The data base will, therefore, always be a regular grid of 67x67 (4489) points. The data base is of this regular form to allow for rapid processing by the planning modules.

This transect method should be utilized if the map is large and/or complex as less map detail will be lost than if using the contour method. The work of digitizing is tedious and subject to human errors. However, an important advantage of this method is that upon completion of the last transect line the data base is ready for use. The map shown in Figure 1 required approximately three hours to digitize.
Figure 2. Entering the Contour Locations for a Transect Line.
Figure 3. Menu for the Transect Method of Entering Contours.
2.2 Contour Method

As in the transect method, the map must be properly aligned with the table. The rectangle is similarly defined. Data entry is simpler than with the transect method: for each contour line the elevation is entered and then the line traced. It is also possible to enter single points (e.g., the elevation of a mountain peak) and transect lines. The ground direction, i.e., uphill, downhill or level, along the transect line is indicated using a menu (Figure 4).

The method is extremely quick and subject to far less human error than the transect approach. The sometimes difficult decision of whether subsequent contours are uphill or downhill is eliminated. Also, the process of following a line is far more efficient than entering specific points. Unfortunately, due to limited machine capacity, this method can only be used for either small maps or those with limited complexity. An additional drawback is that upon completion of the entry, which is frequently less than 30 minutes, the data base is not ready for use. The computer must then compile the irregularly spaced points describing the contour shapes into the same 67x67 regular grid points as used in the transect method; a process which can take up to 12 hours.
Figure 4. Menu for the Contour Method of Entering Contours.
3.0 Analysis of Topographic Features

This module acts as a graphical information retrieval system for ground slope, aspect and elevation. Overlay maps showing these three topographic components, either separately or in combination, can be obtained. This represents a significant improvement over the manual approach currently used. The potential uses are varied, and include:

a) Delineation of areas of steep slope that could indicate slope instability or presence of rock.

B) Analysis of regeneration problems and species selection for planting stock.

C) Determination of logging season: winter versus summer operations.

D) Prediction of general equipment allocation; e.g., skylines in steep terrain and ground skidding in flat areas.

In addition to the overlay map, a tabular distribution of acreage by classes is produced. All class intervals are defined by the user. Slope classes are truncated to even multiples of five percent and elevations to 100 feet. Aspect is categorized into nine classes: N, NE, E, SE, S, SW, W, NW and flat (less
than five percent slope). An example of a slope map with 40 percent classes is given in Figure 5. For combination maps the range of each component must be specified; for example:

ASPECT: southeast, southwest
SLOPE: 40 - 79% (could be specified as 75-79%)
ELEVATION: 500 - 999 feet

The results are shown in Figure 6.
Figure 5. Map Overlay for the 40 Percent Slope Classes.
Figure 6. Combination Overlay for Areas Between 40 and 79% Slope, 500 and 999 Feet Elevation and Southeast and Southwest Aspects.
4.0 Yarding Location and Setting Design

This module allows the planner to test the physical feasibility and potential of cable yarding systems in various locations on the terrain base. Economic evaluations (costs and machine production) are done using the Harvesting Costs and Wood Volume Production Module.

The Yarding Location and Setting Design Module provides the following functions:

- identification of useful landing locations
- evaluation of road spacing
- forecasting of machine allocation
- prediction of potential areas of difficult yarding

The computer requires the maximum possible external yarding distance and the heights of the landing and back spars. Also required are the weight of the load (logs plus carriage) and the weights and operating configuration of the cables (Figure 7). To simplify calculations, these latter components (load, cable weights, etc.) are accounted for by using the percent deflection
Figure 7. A Cable Yarding System
rule. The user must input the minimum attainable midspan deflection. The loadpath (the path the carriage takes over the terrain) is then determined by interpolation (Section 4.2).

There are three types of analysis available:

(1) The generation of ground profiles between any two map points located with the digitizer (Figure 8).

(2) The delineation of area accessible by a yarding system from a certain location; e.g., a landing or tailhold. The system specifications (as stated above) are needed by the computer along with the landing location (which is entered from the digitizer). The computer then projects up to 12 yarding roads, arbitrarily spaced 30 degrees apart. The user can, however, choose a subset of these 12. It is not the purpose of this function to project the actual yarding roads to be used.

For each road, the maximum physically possible external yarding distance is determined. This is achieved by placing the tailhold at the user-specified maximum yarding distance from the landing.

---

3 The percent deflection is defined as the ratio between the vertical distance from the chord to the carriage at midspan and the length of the chord (Figure 7). The 'rule' is the minimum attainable deflection for a specific yarding system.
Figure 8. Plot of Ground Profile.
Load clearance* with the ground is then checked for the entire span. If clearance is lacking then the tailhold is moved towards the landing and the process repeated. A profile plot is produced of the yarding road (Figure 9) when the backspar has been properly located. A plan view plot (Figure 10) is done on completion of all the yarding roads. This plot can be used as an overlay to help define the setting boundary on the map.

(3) User-specified yarding roads can be examined. Again, the system specifications are needed along with the landing and tailhold locations. The computer, at the user's discretion, will either move the tailhold to obtain clearance (as discussed above) or leave it as initially located. A profile of the loadpath and the groundline is shown in Figure 11.

*The minimum acceptable clearance is defined as being zero feet between the ground and the carriage. The length of the chokers can be accounted for by subtracting their length from both spar heights.
Figure 9. Ground Profile and Loadpath Plots for Projected Yarding Roads.
Figure 10. Plan View of Projected Yarding Roads

SPARS: 90 ft / 60 ft
AREA = 140 ACRES
% DEF = 8

yarding road # 3
Figure 11. Plot of Ground Profile and Loadpath for a Single Yarding Road.

Elevation (feet)

900

Midspan deflection = 8%
Height of landing spar = 90 feet
Height of back spar = 60 feet

700

500

300

landing spar

back spar locations

Horizontal Distance (feet)

200 400 600 800 1000 1200 1400 1600
5.0 Road Location

This module is typically used in conjunction with the yarding module. A common use is to check whether various landing locations can be connected by a road system. Plan view (Figure 12) and profile (Figure 13) plots along with side slopes, elevations and grades (Figure 14) can be easily generated for any road location.

Road locations are entered into the computer in two ways. In both methods the maximum allowable adverse and favourable road grades are required by the program:

(1) Two control points are selected (e.g. landings, switchbacks, junctions). The computer then locates the average groundline, subject to the grade constraints. The method employed is similar to manually projecting roads from a map with dividers.

(2) The user determines the location interactively. Points are entered along the proposed route with the digitizer (Figure 12). As each successive point is entered the grade is computed. If the grade exceeds allowable limits an audible warning is given and a new location is expected. Grade specifications can be easily over-ridden in difficult terrain.
Figure 12. Plan View Plot of Road Network.
Figure 13. Road Profiles
<table>
<thead>
<tr>
<th>ROAD 1 STATION</th>
<th>ELEVATION (FEET)</th>
<th>SIDESLOPE (PERCENT)</th>
<th>GRADE (PERCENT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>163</td>
<td>29</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>364</td>
<td>66</td>
<td>24</td>
<td>18</td>
</tr>
<tr>
<td>536</td>
<td>98</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>674</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>839</td>
<td>101</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1047</td>
<td>101</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>1172</td>
<td>123</td>
<td>31</td>
<td>22</td>
</tr>
<tr>
<td>1361</td>
<td>165</td>
<td>33</td>
<td>20</td>
</tr>
<tr>
<td>1466</td>
<td>186</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>1528</td>
<td>197</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1665</td>
<td>199</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1750</td>
<td>200</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1830</td>
<td>200</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>1947</td>
<td>200</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROAD 2 STATION</th>
<th>ELEVATION (FEET)</th>
<th>SIDESLOPE (PERCENT)</th>
<th>GRADE (PERCENT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>168</td>
<td>44</td>
<td>14</td>
</tr>
<tr>
<td>201</td>
<td>198</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>446</td>
<td>200</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>626</td>
<td>202</td>
<td>22</td>
<td>-5</td>
</tr>
<tr>
<td>672</td>
<td>200</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>752</td>
<td>207</td>
<td>15</td>
<td>-2</td>
</tr>
<tr>
<td>965</td>
<td>204</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>1102</td>
<td>225</td>
<td>48</td>
<td>5</td>
</tr>
<tr>
<td>1348</td>
<td>238</td>
<td>53</td>
<td>6</td>
</tr>
<tr>
<td>1509</td>
<td>248</td>
<td>49</td>
<td>-8</td>
</tr>
<tr>
<td>1685</td>
<td>235</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>1789</td>
<td>236</td>
<td>45</td>
<td>18</td>
</tr>
<tr>
<td>1975</td>
<td>270</td>
<td>57</td>
<td>13</td>
</tr>
<tr>
<td>2165</td>
<td>296</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>2357</td>
<td>300</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>2514</td>
<td>319</td>
<td>33</td>
<td>18</td>
</tr>
<tr>
<td>2664</td>
<td>348</td>
<td>76</td>
<td>20</td>
</tr>
<tr>
<td>2893</td>
<td>395</td>
<td>46</td>
<td>1</td>
</tr>
<tr>
<td>3092</td>
<td>397</td>
<td>48</td>
<td>5</td>
</tr>
<tr>
<td>3259</td>
<td>406</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

Figure 14. Road Analysis Output.
(Note: computer output has been re-typed for clarity)
No earthwork volume calculations are done as the terrain data are not sufficiently accurate to warrant such detailed analysis.
6.0 Aesthetic Analysis

Two forms of aesthetic analysis are available: viewable area (similar to VIEWIT) and projection of three dimensional terrain representations onto two dimensional surfaces.

6.1 Viewable Area

The location and elevation of the viewing point, which can be outside the data base, are entered. The computer then produces an overlay map showing the viewable area (Figure 15), and a tabular summary of the associated acreage.

Several different viewpoints can be combined to produce one overlay. This is useful when analyzing the viewable area from a road. Planametric details (roads, settings, etc.) can be plotted on the overlay as in Figure 15. An initial viewing angle can be specified in cases where the viewpoint is partially screened by a timber edge or cutbank (Figure 16); this is commonly used when the viewpoint is outside the map.

---

Throughout the text the phrase 'three dimensional representation' is used to define a three dimensional representation that is produced on a two dimensional surface.
Figure 15. Map Overlay for Viewable Areas
Figure 16. Determining the Initial Viewing Angle
6.2 Three Dimensional Representations

This feature produces a three dimensional (3D) representation (pseudo-photograph) of the data base. The viewing directions along with the area of the data base being viewed are input. No precise viewpoint is used as the projection is orthographic and not perspective. This limitation is due to the slow execution time of the computer coupled with the slightly more complicated algorithm required to produce perspective. Perspective plotting could be implemented if deemed necessary and is discussed in the theoretical section. The differences between the two methods are not noticeable when viewing from a long distance as shown in Figure 17 and Plate 3. The area is Cypress Bowl Provincial Park, north of Vancouver.

A planometric map can also be produced showing the locations of the hidden areas in the 3D plot (Figure 18). Planometric details can be plotted onto either the hidden area map or the 3D plot, aiding in the assessment of the visual impact of disturbances.
Figure 17. Orthogonal Plot of Cypress Bowl Provincial Park (looking north).
Plate 3. Cypress Bowl Provincial Park (looking north).
Figure 18. Planometric Map Showing the Hidden Areas for the Orthogonal Projection of Cypress Bowl Provincial Park.
7.0 Harvesting Costs and Wood Volume Production

This module performs an economic assessment of a logging plan. Using the locations derived from the yarding and road design sections, costs and productivity can be estimated for the various layouts. The results can be used for comparisons of layout alternatives.

An inventory data base must be set up before analysis can proceed. The various covertype locations along with their associated species volumes must be entered into the computer. The data base is structured on a grid system, the map being divided into rectangular cells. One point is digitized in each cell whose centroid lies within the covertype currently being entered into the data base (Figure 19). A simplified covertype map (Figure 20) is produced to verify that the entry is correct; errors can easily be corrected. The resulting data base is stored on a data tape. Soil/landform information, not presently used in the terrain simulator, could be similarly handled.

Analysis can now proceed on any setting. At present, production data are available for only three cable systems (the Madill 052 tension skidder, 90-foot spar highlead system and the Madill grapple yarder). Others could easily be added so that virtually any logging system could be simulated. First, the landing location is digitized. Then, the setting area is entered using the same grid system employed for entering the
Figure 19. Map of Study Area Showing Coverttype Boundaries and Associated Grid Overlay
<table>
<thead>
<tr>
<th>cell</th>
<th>1</th>
<th>6</th>
<th>11</th>
<th>16</th>
<th>21</th>
<th>26</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subscripts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>AAAABBBBBBBBBBBBDBHNNHHHHHHHHHHH COVERTYPE LISTING</td>
<td>SPECIES</td>
<td>% VOLUME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE A</td>
<td>TOTAL VOLUME = 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH (100's cu.ft.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>AAAABBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE B</td>
<td>TOTAL VOLUME = 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE C</td>
<td>TOTAL VOLUME = 120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE D</td>
<td>TOTAL VOLUME = 150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE E</td>
<td>TOTAL VOLUME = 120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE F</td>
<td>TOTAL VOLUME = 70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH COVERTYPE G</td>
<td>TOTAL VOLUME = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BBBBBBBBBBBBBBBBDHHHHHHHHHHHHHHH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 20. Covertype Map and Associated Species Listing. (Note: computer output has been re-typed)
covertypes. The computer then simulates the yarding of the wood in each setting cell to the landing. The average slope and yarding distance are computed and used to predict the productivity rate for that cell. Soil/landform information would prove useful at this stage as it will indicate general ground conditions. On completion of the setting, the acreage, volume by species, average yarding distance, average slope, total yarding cost, number of shifts and the percentage loss of the normal production rate (due to difficult yarding) are all computed and printed (Figure 21).

Another function of the module is the estimation of road construction costs. The projected road is traced using the digitizer. The simulator computes the length and applies an average cost per mile giving the road costs (Figure 21). Once again soil information would enhance the estimates.
ACRES = 127.4

<table>
<thead>
<tr>
<th>SPECIES</th>
<th>VOLUME (CUNITS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR</td>
<td>2911.0</td>
</tr>
<tr>
<td>CEDAR</td>
<td>5633.0</td>
</tr>
<tr>
<td>CYPRS</td>
<td>267.0</td>
</tr>
<tr>
<td>HEMLK</td>
<td>5371.0</td>
</tr>
<tr>
<td>BASLM</td>
<td>259.0</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td><strong>14441.0</strong></td>
</tr>
</tbody>
</table>

AVE VOL/ACRE = 113.3 CUNITS

MACHINE: SKIDDER

# SHIFTS = 205.3

YARDING COST = $265065.5

AVE % PRODUCTION LOSS = 8.1

AVE YARDING DISTANCE = 928.2 FEET

AVE SLOPE = 22.9 PERCENT

ROAD LENGTH = 0.4 COST @ $55,000/MILE = $19924.5

Figure 21. Costing Analyses for Roads and Settings.

(Note: computer output has been re-typed for clarity)
Part II of the thesis outlines the important theory of each module in the terrain simulator. Supporting examples used are, of necessity, dependent on the HP9830A computer. Therefore, at the end of each section is a brief discussion of the implementation of the theory on this system. All analyses, irrespective of the module, are in part dependent on elevations, making the structure of the elevation data base critical to the development of the modules. Consequently it is necessary to describe the nature of the elevation data base prior to proceeding to the rest of the simulator.
1.0 Elevation Data Base

The elevation data base must provide for quick, accurate retrieval of elevations for any map location. In generating the elevations a tradeoff exists between speed and accuracy, as generally more computation is required to achieve greater accuracy. There are two levels of accuracy required by the various modules:

a) High level accuracy: Elevations for specific (x,y) locations on the data base are required to the precision of the map and consequently speed will be sacrificed. Examples include skyline ground profiles, road profiles, costing and productivity estimation and viewable area assessment.

b) Low level accuracy: Approximate elevations, that are representative of a certain vicinity, are sufficient to classify the topographic features and produce three dimensional plots. Although the accuracy is less, retrieval time is more rapid.

In both cases the accuracy is measured not only by how well the data base describes the map but also the ability of the map to describe the ground. Even though map-making organizations are required to contour maps to certain standards (Meyer, 1969) the accuracy of the map can be extremely variable and difficult to predict. Although crucial, no comprehensive analysis of this
problem is provided here. The model therefore considers the map to be correct.

Two basic data structures can be employed to store the elevation data for subsequent use:

1) Retention of contour locations;
2) Conversion of contours into a grid of elevations.

1.1 Retention of Contour Locations.

The \((x,y)\) coordinates describing the shape of each map contour are retained in the database. Consequently, the database should provide results that are as accurate as the map itself as there is virtually no information loss. Unfortunately a large amount of fast-access storage is required. The retrieval of elevations is also slow and difficult. These two drawbacks make implementation on a desktop system infeasible. Therefore, this approach is only outlined and no detailed discussion will be given.

Finding the elevation of an \((x,y)\) point on the map is difficult because the two contour lines that bracket \((x,y)\) must be located. The interval defined by the two contour lines can be considered a closed polygon. To decide if a point is contained by the polygon it is necessary to check it against each line segment that makes up the polygon boundary. This
requires a point-in-polygon routine. An algorithm is given in Appendix A.

The method is slow because of the exhaustive test made of the boundary. Once the proper interval is located the elevation is computed by interpolating between the contour limits. A desktop computer cannot realistically handle this task as the slow execution will seriously detract from the interactive nature of the model.

An alternative approach is to disregard the contour shapes and simply interpolate between the closest \((x,y)\) contour points eliminating the need to locate the containing contour interval. Since the contour shapes are disregarded the closest contour points employed in the interpolation may not lie in the proper contour interval, resulting in decreased accuracy. For example, if an \((x,y)\) point is between the 2900 and 3000-foot contours, some of the closest contour points used, may have elevations greater or less than 3000 or 2900 feet, respectively. This type of error will commonly occur in steep terrain where the contour lines are close together.

The major time component in the two approaches is locating either the bracketing contours or the closest contour points. This search time can be drastically reduced by initially subdividing the map into a grid of small cells. It is then necessary to examine only the section of the data base that lies within the appropriate cell (and its eight adjacent neighbours)
that contains the \((x,y)\) point for which the elevation is being sought. Even this improvement is not sufficient for repeated use as the process is still too slow. An additional problem is the increased memory required because of the more elaborate data structure.

1.2 Conversion of Contours into a Grid of Elevations

In this method, the contour information is condensed to form a regular grid of elevations. The advantages lie in fast retrieval (for both the 'high' and 'low' levels of accuracy) and relatively small memory requirements making it suitable for desktop computers.

The basis of the method lies in dividing the map area into equal rectangular cells. Each cell surface is described mathematically by either an interpolation routine that uses the cell corner elevations, or, by a surface equation which is typically some form of geometric plane. The interpolation approach requires more computational work to generate an elevation but storage requirements are less as only one elevation per cell need be retained. The equivalent surface equation, (computed only once at the input stage using the same grid of elevations as the interpolation method) is simple to evaluate, but each of the parameters must be retained thereby increasing storage requirements.
The definition of the coordinate system is the same for both methods. The rectangle, defining the map area, has its origin (0,0) at its lower left corner (Figure 22). The diagonally opposite corner (upper right) is defined by the coordinate pair (u,v). The coordinates are expressed in digitizer units (usually inches). The rectangle contains \( n^2 \) cells, each \((u/n)(v/n)\) in size.

The \( n^2 \) cells are described by a grid of \((n+1)^2\) elevations with one elevation at each of the four cell corners. These elevations are stored in a matrix \( E(j,k) \) where \( j \) is the \( x \)-coordinate and \( k \) the \( y \)-coordinate. For an \((x,y)\) point in the data base the subscripts for the containing cell are:

\[
j=[xn/u]+1 \quad k=[yn/v]+1 \quad \ldots \ldots \quad (1.1)
\]

where \([\ ]\) represents the nearest lower integer value

The subscripts \((j,k)\) also define the cell's lower left elevation, \( E(j,k) \). The corresponding location of \((j,k)\) in digitizer units is \((s,t)\) with:

\[
s=(j-1)u/n \quad t=(k-1)v/n \quad \ldots \ldots \quad (1.2)
\]

The coordinates and elevations for each of the four cell corners are given in Table II.
Figure 22. Definition of Coordinate System.
1.2.1 Interpolation Method for Finding Elevations

An elevation for an \((x, y)\) location can be determined by interpolating between the elevations for the closest cell corners in the data base. The subscripts for the cell that contains \((x, y)\) are found using Equation (1.1). The number of grid points used will depend on the technique employed. To prevent bias a symmetrical set of grid points should be used; e.g., the four cell corner elevations. Using more than four points requires unacceptable amounts of computation time with little gain in accuracy (Section 1.2.2).

The two most appropriate methods developed in terms of speed and accuracy, when using the four cell corner elevations, were:

---

### TABLE II

COORDINATES AND ELEVATIONS OF THE FOUR CELL CORNERS

<table>
<thead>
<tr>
<th>corner</th>
<th>coordinates</th>
<th>elevations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 lower left</td>
<td>((s, t))</td>
<td>(E(j, k))</td>
</tr>
<tr>
<td>2 upper left</td>
<td>((s, t+v/n))</td>
<td>(E(j, k+1))</td>
</tr>
<tr>
<td>3 upper right</td>
<td>((s+u/n, t+v/n))</td>
<td>(E(j+1, k+1))</td>
</tr>
<tr>
<td>4 lower right</td>
<td>((s+u/n, t))</td>
<td>(E(j+1, k))</td>
</tr>
</tbody>
</table>
1) Three-step linear interpolation to form a weighted average of the four data points

2) Weighted average using 'inverse distance-squared law'

Three-Step Linear Interpolation for Computing Elevations

Three steps, each using linear interpolation, are required to generate an elevation for a point \((x,y)\) (Figure 23).

1) An estimate, \(e_1\), is made of the elevation at \((s,y)\) using \(E(j,k)\) and \(E(j,k+1)\):

\[
e_1 = E(j,k) + (E(j,k+1) - E(j,k)) \frac{(y-t)}{(v/n)}
\]

2) Similarly an estimate, \(e_2\), is made of the elevation at \((s+u/n,y)\) using \(E(j+1,k+1)\) and \(E(j+1,k)\):

\[
e_2 = E(j+1,k) + (E(j+1,k+1) - E(j+1,k)) \frac{(y-t)}{(v/n)}
\]

3) Finally, the elevation is estimated at \((x,y)\) by using the two elevations, \(e_1\) and \(e_2\), that lie on the line \(y=y\) (defined by \((s,y)\) and \((s+u/n,y)\)):

\[
e_{x,y} = e_1 + (e_2 - e_1) \frac{(x-s)}{(u/n)}
\]
Figure 23. Computing the Elevation for the Point \((x, y)\) Using 3-step Linear Interpolation and Inverse Distance-squared Interpolation.
Inverse Distance-Squared Interpolation for Computing Elevations

The elevation for a point \((x,y)\) is found by averaging the four cell corner elevations by the inverse of their distance-squared to \((x,y)\) thereby giving less weight to those cell corners furthest away from \((x,y)\) (Figure 23).

Computing the distance-squared to the corners:

\[
D_1^2 = (x-s)^2 + (y-t)^2 \\
D_2^2 = (x-s)^2 + (y-t-v/n)^2 \\
D_3^2 = (x-s-u/n)^2 + (y-t-v/n)^2 \\
D_4^2 = (x-s-u/n)^2 + (y-t)^2
\]

The elevation is then:

\[
e_{x,y} = \frac{E(j,k)}{D_1^2} + E(j,k+1)/D_2^2 + E(j+1,k+1)/D_3^2 + E(j+1,k)/D_4^2 / D
\]

Where \(D\) is the sum of the reciprocals:

\[
D = \frac{1}{D_1^2} + \frac{1}{D_2^2} + \frac{1}{D_3^2} + \frac{1}{D_4^2}
\]

Comparison of the Two Interpolation Methods

The accuracy was tested by tracing a contour line with the digitizer and, for each point entered, computing the elevation using both methods. The error in elevation at each point was determined by subtracting the actual contour elevation from the calculated value.

A sample size of 500 points was taken from five maps of
differing sizes but of identical scales. Figure 24 illustrates a sub-sample of 20 points obtained from digitizing one contour. The elevation errors for each of the 500 points were plotted against map sizes (Figure 25). These results are also given in Table III. These show the obvious, with a fixed number of elevation grid points the error and its variance will increase with the map size. The magnitude of the errors will depend on map scale, complexity of the contours, quality of the map drafting and the accuracy of the operator's digitizing. The distance-squared and three-step approaches yielded almost identical results. This was confirmed statistically by testing the null hypothesis that the difference in the mean error for the two methods, using paired observations, was zero. It was assumed that any differences between the two methods was not dependent on map size. The z-value was tested against a five percent critical limit:

\[ z = \frac{\sum_{i=1}^{n}(x_i - y_i)}{\sqrt{n}} / (\sigma / \sqrt{n}) \]

with \( x_i \): error for the \( i \)th observation using the three-step method
\( y_i \): error for the \( i \)th observation using the distance-squared method
\( \sigma \): standard deviation of the differences
\( n \): sample size

The value of \( z \) was 1.19 which is well below the critical limit of 1.96 hence the difference in the means is insignificant.

A more precise comparison can be made of the computational speed of the two methods. An operation count, i.e., the number
DISTANCE-SQUARED

<table>
<thead>
<tr>
<th>ERROR (FEET)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>5.6</td>
</tr>
<tr>
<td>12.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2.9</td>
<td>7.2</td>
</tr>
<tr>
<td>-11.2</td>
<td>-9.1</td>
</tr>
<tr>
<td>-17.7</td>
<td>-12.8</td>
</tr>
<tr>
<td>6.4</td>
<td>5.1</td>
</tr>
<tr>
<td>-10.6</td>
<td>-6.9</td>
</tr>
<tr>
<td>9.5</td>
<td>11.5</td>
</tr>
<tr>
<td>22.3</td>
<td>23.0</td>
</tr>
<tr>
<td>12.3</td>
<td>12.5</td>
</tr>
<tr>
<td>5.2</td>
<td>8.9</td>
</tr>
<tr>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>4.0</td>
</tr>
<tr>
<td>-7.7</td>
<td>-2.1</td>
</tr>
<tr>
<td>-6.4</td>
<td>-7.8</td>
</tr>
<tr>
<td>-2.6</td>
<td>-5.7</td>
</tr>
<tr>
<td>-2.9</td>
<td>-2.9</td>
</tr>
<tr>
<td>6.5</td>
<td>7.2</td>
</tr>
</tbody>
</table>

THREE-STEP LINEAR

<table>
<thead>
<tr>
<th>ERROR (FEET)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>5.6</td>
</tr>
<tr>
<td>12.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2.9</td>
<td>7.2</td>
</tr>
<tr>
<td>-11.2</td>
<td>-9.1</td>
</tr>
<tr>
<td>-17.7</td>
<td>-12.8</td>
</tr>
<tr>
<td>6.4</td>
<td>5.1</td>
</tr>
<tr>
<td>-10.6</td>
<td>-6.9</td>
</tr>
<tr>
<td>9.5</td>
<td>11.5</td>
</tr>
<tr>
<td>22.3</td>
<td>23.0</td>
</tr>
<tr>
<td>12.3</td>
<td>12.5</td>
</tr>
<tr>
<td>5.2</td>
<td>8.9</td>
</tr>
<tr>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>4.0</td>
</tr>
<tr>
<td>-7.7</td>
<td>-2.1</td>
</tr>
<tr>
<td>-6.4</td>
<td>-7.8</td>
</tr>
<tr>
<td>-2.6</td>
<td>-5.7</td>
</tr>
<tr>
<td>-2.9</td>
<td>-2.9</td>
</tr>
<tr>
<td>6.5</td>
<td>7.2</td>
</tr>
</tbody>
</table>

CONTOUR ELEVATION: 2900.0
NUMBER OF POINTS: 19.0
THE AVERAGE (MEAN) ERROR: 8.0
THE LARGEST ERROR: 22.3
VARIANCE: 92.4
STANDARD DEVIATION: 9.6

CONTOUR ELEVATION: 2900.0
NUMBER OF POINTS: 19.0
THE AVERAGE (MEAN) ERROR: 7.0
THE LARGEST ERROR: 23.0
VARIANCE: 77.7
STANDARD DEVIATION: 8.8

Figure 24. Accuracy Test Comparing the Three-step Linear and the Inverse Distance-squared Techniques for Computing Elevations.

(Note: computer output has been re-typed for clarity)
Figure 25. Graph of the Elevation Error as a Function of Map Area
### TABLE III

**COMPARISON OF ELEVATION ERRORS FROM MAPS OF DIFFERENT SIZES**

<table>
<thead>
<tr>
<th>Map Area (sq.in.)</th>
<th>Map Perimeter (in.)</th>
<th>Distance-squared Method Variance (ft²)</th>
<th>Method Mean* (ft)</th>
<th>Three-step Method Variance (ft²)</th>
<th>Method Mean (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>315.7</td>
<td>74.2</td>
<td>14.7</td>
<td>14.1</td>
<td>12.2</td>
<td>13.8</td>
</tr>
<tr>
<td>261.2</td>
<td>64.8</td>
<td>11.5</td>
<td>9.3</td>
<td>11.1</td>
<td>9.2</td>
</tr>
<tr>
<td>150.0</td>
<td>50.0</td>
<td>7.9</td>
<td>6.1</td>
<td>7.3</td>
<td>6.2</td>
</tr>
<tr>
<td>80.1</td>
<td>36.0</td>
<td>3.9</td>
<td>2.7</td>
<td>3.8</td>
<td>2.6</td>
</tr>
<tr>
<td>30.2</td>
<td>22.4</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
</tbody>
</table>

*All errors are computed as absolute values.
of additions, subtractions, etc., was done for each method (Table IV). These values can be compared by converting them to

<table>
<thead>
<tr>
<th>TABLE IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPARISON OF OPERATION COUNTS</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>Three-Step Method</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>3 additions</td>
</tr>
<tr>
<td>6 subtractions</td>
</tr>
<tr>
<td>6 multiplications</td>
</tr>
<tr>
<td>0 divisions</td>
</tr>
</tbody>
</table>

a measure of time. Using one addition as the basic time unit and the relative execution times for the Hewlett-Packard 9830A computer (Table V) the three-step method requires only 24 time units as compared to the 70 used by the inverse distance-squared approach. The desired method for implementation is therefore

<table>
<thead>
<tr>
<th>TABLE V</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATIVE EXECUTION TIMES</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>1 subtraction = 1 addition</td>
</tr>
<tr>
<td>1 multiplication = 2.5 additions</td>
</tr>
<tr>
<td>1 division = 7 additions</td>
</tr>
</tbody>
</table>

the three-step method as it is three times faster while still being as accurate as the distance-squared approach. However, this is only true when the data points are spaced in a regular
grid as is the case for the elevation data base. If the points are irregularly spaced then the distance-squared method is the effective approach as it is independent of the orientation of the points.

1.2.2 Surface Equation Method for Finding Elevations

In the surface equation method, interpolating polynomials are computed for each cell. They are subject to the constraint that elevations generated along the cell boundaries are the same no matter which of the adjacent cell surface equations are used, thereby producing continuous elevations with no jump-discontinuities between cells. This property is crucial when generating road and skyline profiles, thus, statistically fitted surfaces can not be used. A statistical approach, however, is useful when determining the topographic features where continuous elevations are not required (Section 3).

There are many feasible interpolating polynomials available. The number of terms in the equation will depend on the number of data points (nodes) used. There must be the same number of nodes as parameters being calculated in order for the polynomial to be unique (Shampine and Allen, 1973).

The simplest form is a zero degree system. However, this approach yields a flat surface, with no slope, creating a discontinuity at the boundaries. Similarly, a first degree
system, that permits slope in either the x or y direction, has discontinuities. It is therefore necessary to have at least a first degree, three node system that allows for slope in both the x and y directions.

A first degree, three node system is described by

\[ m_1 x + m_2 y + m_3 z = 1 \]

with \( z \) being the elevation and the parameters, \( m_i \), the rates of change (slope) with respect to \( x, y \) and \( z \). This equation requires three nodes. Every cell would therefore have to be divided into two equal triangles each with its own equation. The triangles introduce the extra computational task of deciding which triangle the \((x, y)\) point lies in and the requirement of \( 6n^2 \) elements of storage. Besides this large amount of storage and increased computational effort the errors are greater than the system to be described below.

The next possible system uses four nodes and is of the form

\[ m_1 x + m_2 y + m_3 xy + m_4 z = 1 \]

which is equivalent to the three-step interpolation method described earlier. This system best fits the grid-type data structure as the four required nodes coincide nicely with the four cell corners. The parameters \( m \) can be solved for by constructing a system of four linearly independent equations, each using one of the four cell corner elevations. The solution, which is straightforward, is as follows:

\[ A \mathbf{m} = \mathbf{b} \quad \text{therefore} \quad \mathbf{m} = A^{-1} \mathbf{b} \]
where \( A \) is the matrix of coefficients

\[
A = \begin{bmatrix}
  s & t & \text{st} & E(j,k) \\
  s & t+v/n & s(t+v/n) & E(j,k+1) \\
  s+u/n & t+v/n & (s+u/n)(t+v/n) & E(j+1,k+1) \\
  s+u/n & t & (s+u/n)t & E(j+1,k)
\end{bmatrix}
\]

\( b \) is the vector of constants

\[
b = \begin{bmatrix}
  1 \\
  1 \\
  1 \\
  1
\end{bmatrix}
\]

and \( m \) is the vector of parameters

\[
m = \begin{bmatrix}
  m_1 \\
  m_2 \\
  m_3 \\
  m_4
\end{bmatrix}
\]

Therefore

\[
z = (1-m_1 x-m_2 y-m_3 xy)/m_4 \]

and

\[
z = Ax + By + Cxy + D
\]

with

\[
A = -m_1 / m_4
\]

\[
B = -m_2 / m_4
\]

\[
C = -m_3 / m_4
\]

\[
D = 1 / m_4
\]

This system of equations is solved for each grid cell with parameters \( A, B, C, D \) being stored for the cell, requiring \( 4n^2 \)
elements of storage. Equation (1.3) takes only 10.5 time units to evaluate.

Other surfaces (with more nodes) can be utilized with the major problem being how many and which nodes to use. Since the basic data structure is a regular grid the next logical form should require 9 points which yields a complete quadratic (second degree) surface:

\[ m_1 x^2 y^2 + m_2 x^2 + m_3 x + m_4 x y + m_5 y^2 + m_6 y + m_7 y^2 x + m_8 x^2 y + m_9 z = 1 \]

The solution is similar to the four node case but with nine equations that are all linear in \( m_1 \). Evaluation of the equation is slow requiring 50 time units to compute one elevation. Another serious drawback is the increased flexibility of the polynomial gained through more terms. Consequently little can be guaranteed about the accuracy when evaluating between the nodes. This was borne out in tests similar to the one described in Section 1.2.1 (Figure 26). In many cases the results were excellent but there were some alarming errors (upwards of 50 feet) in places.

1.3 Implementation

The crucial factor in deciding the type of data structure was the limited amount of computer memory and the lack of a fast-access storage device (e.g., disc drive). Consequently, a grid-type data base was used employing a 67x67 array of elevations; i.e., \( n=66 \).
### THREE-STEP LINEAR

<table>
<thead>
<tr>
<th>Error (Feet)</th>
<th>Error (Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td>-3.0</td>
<td>-8.0</td>
</tr>
<tr>
<td>-4.2</td>
<td>-0.0</td>
</tr>
<tr>
<td>1.4</td>
<td>1.0</td>
</tr>
<tr>
<td>8.2</td>
<td>2.0</td>
</tr>
<tr>
<td>-0.4</td>
<td>-2.0</td>
</tr>
<tr>
<td>-6.7</td>
<td>-1.0</td>
</tr>
<tr>
<td>-8.4</td>
<td>-1.1</td>
</tr>
<tr>
<td>-4.5</td>
<td>-2.0</td>
</tr>
</tbody>
</table>

Contour Elevation: 2900.0

Number of Points: 10.0

The Average (Mean) Error: 4.0

The Largest Error: -8.4

Variance: 23.9

Standard Deviation: 4.9

### QUADRATIC

<table>
<thead>
<tr>
<th>Error (Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.7</td>
</tr>
<tr>
<td>8.9</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>4.8</td>
</tr>
<tr>
<td>7.9</td>
</tr>
<tr>
<td>-0.0</td>
</tr>
<tr>
<td>-0.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>1.9</td>
</tr>
</tbody>
</table>

Contour Elevation: 2900.0

Number of Points: 10.0

The Average (Mean) Error: 2.0

The Largest Error: -8.0

Variance: 8.3

Standard Deviation: 2.9

### THREE-STEP LINEAR

<table>
<thead>
<tr>
<th>Error (Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.2</td>
</tr>
<tr>
<td>5.5</td>
</tr>
<tr>
<td>-3.6</td>
</tr>
<tr>
<td>-3.8</td>
</tr>
<tr>
<td>6.0</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>2.0</td>
</tr>
<tr>
<td>9.1</td>
</tr>
<tr>
<td>6.0</td>
</tr>
<tr>
<td>1.1</td>
</tr>
</tbody>
</table>

Contour Elevation: 2500.0

Number of Points: 10.0

The Average (Mean) Error: 7.0

The Largest Error: 29.2

Variance: 89.1

Standard Deviation: 9.4

### QUADRATIC

<table>
<thead>
<tr>
<th>Error (Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.7</td>
</tr>
<tr>
<td>8.9</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>4.8</td>
</tr>
<tr>
<td>7.9</td>
</tr>
<tr>
<td>-0.0</td>
</tr>
<tr>
<td>-0.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>1.9</td>
</tr>
</tbody>
</table>

Contour Elevation: 2500.0

Number of Points: 10.0

The Average (Mean) Error: 8.0

The Largest Error: 54.7

Variance: 282.0

Standard Deviation: 16.8

Figure 26. Accuracy Test for the Quadratic Interpolation Technique.

(Note: computer output has been re-typed for clarity)
Using integer storage, one elevation requires one word\(^6\) of memory thereby using 4489 of the 8000 available words. In addition, the coordinates \((u,v)\), defining the map area rectangle, and the map scale were retained as part of the data base. The retrieval of the elevations was accomplished by the three-step linear interpolation method.

\[\text{One word} = 2 \text{ bytes} = 16 \text{ bits}\]
2.0 Input of the Elevation Data Base

The purpose of the Elevation Data Base Input Module is to allow the user to enter the contour locations, that describe the terrain, into the computer. The contour information is then condensed to form a regular grid of elevations. The input and subsequent grid creation are performed, as discussed in Part I, by either the transect or contour method. Both methods produce the same data base structure of an \((n+1)^2\) grid of elevations.

2.1 Transect Method

In the transect method the input is controlled by the menu (Figure 3) whose operation is described in Appendix B. The \(n+1\) transect lines, used to describe the map, correspond to the columns of the elevation matrix \(E\) (using the same definitions given in Section 1.2). Each transect line has its origin at the bottom of the map (i.e., the \(x\)-axis \((Y=0)\)) and the endpoint at the top of the map \((Y=v)\).

Upon completion of the entry of transect line \(j\), the irregularly spaced contour intersections are converted to \(n+1\) elevation points spaced at intervals of \(v/n\) along the line. These points correspond to the intersections of the rows of the \(E\) matrix with the column (transect line \(j\)) being compiled. Each
of these elevations \((E(j,k), k=1,2,\ldots,n+1)\) is computed by linearly interpolating between the two closest, bracketing, contour intersections on the transect line. Therefore, the elevation for grid point \((j,k)\), whose map location is \((s,t)\) (Equation 1.2), will be:

\[
E(j,k) = T(i,2) + (T(i+1,2) - T(i,2)) \frac{(t-T(i,1))}{(T(i+1,1) - T(i,1))}
\]

with \(T(i,r)\) the matrix of the \((y,z)\) points (contour intersections) for transect line \(j\)

- \(r=1\): y-coordinate \((y)\)
- \(r=2\): elevation \((z)\)

\(i\) = subscript of the contour intersection in \(T\) that is closest to and still less than the \(t\)-coordinate of the grid point \((s,t)\).

After compiling the line, the same matrix, \(T\), is used for the next line since contour intersections are retained only for the line being compiled.

The major disadvantage of the transect method is that the elevations for each line are compiled independently. Large errors, up to one half the contour interval, can occur in regions where the transect lines run parallel to the contour lines. These errors could be reduced by running a second set of lines \((T^*)\) perpendicular to the series just described (Figure 27). These will correspond to the rows of the matrix \(E\).

In this double transect method the elevation \(E(j,k)\) for
Figure 27. Double Transect Method for Generating Elevations.
grid point \((j,k)\) would be computed by locating the two closest bracketing points on \(T\) and similarly the two closest on \(T^\perp\) and then weighting these four elevations by the inverse distance-squared method (Section 1.2.1) to form the average. The three-step linear interpolation method can not be used efficiently as the elevation points would be irregularly spaced. Although accuracy would be improved, it is now necessary to retain all the intersections for the \(2(n+1)\) lines:

\[
T_i : i=1,2,3,\ldots,n+1 \quad \text{and} \quad T^\perp_i : i=1,2,3,\ldots,n+1
\]

The error reduction has not been quantified as memory limitations did not permit implementation on the HP computer. Also, the amount of work required to digitize the contour locations would be doubled.

2.2 Contour Method

The technique used in the contour method is similar to that described in Section (1.1). To form an elevation \(E(j,k)\), whose digitizer coordinates are \((s,t)\), it is necessary to search through the contour points digitized by the user, for the closest points. The inverse distance-squared technique is then applied to these points to obtain the elevation. Searching the entire input data is both time consuming and unnecessary. Thus, to employ a systematic search, the map area is subdivided into cells and only the relevant cells are then searched.

The process of computing the elevation grid is done, therefore, in three phases:
1) entry of contour locations from the digitizer;
2) sorting of \((x,y)\) contour points into smaller map units to allow systematic searching (data structuring);
3) generating the \((n+1)^2\) elevations for the matrix \(E\).

2.2.1 Entry of Contour Lines

The entry of the contour lines is controlled by a menu (Figure 4). The \((x,y)\) points are entered in continuous mode\(^7\) from the digitizer as each contour is traced. Not all the points are used. Those close together are thinned out using a distance test from the last recorded point \((x_i,y_i)\):

\[
\text{reject } (x,y) \text{ if } (x-x_i)^2 + (y-y_i)^2 < \text{tolerance}^2
\]

The thinning is also done to ensure that the contours are described by an evenly distributed set of points.

2.2.2 Sorting of Data Points to Facilitate Searching

The contour points can be classified into either large grids or bands to facilitate more efficient manipulation.

---

\(^7\) Two entry modes can be used when digitizing. A single point entry allows the user to select the point to be digitized. The continuous mode permits the computer to take a continuous stream of points and is therefore used when tracing lines.
The Band Data Structure for Searching

Bands can be created in either the x or y direction. For purposes of discussion (and implementation) the bands are assumed parallel to the x-axis (Figure 28).

The contour points are placed into bands, \( b_i \), each having a width of \( w \). A total of \( u/w \) bands will be required to cover the map area. The point \((x, y, z)\) is in \( b_i \) if:

\[
(i-1)w \leq y < iw
\]

The points are stored in band \( b_i \) by constructing a four dimensional matrix \( B(i, x, y, z) \). If the computer memory is limiting, an alternate method could be to store the points associated with each band onto separate data tape files. This would allow individual bands to be brought into memory as required.

The search time for locating the closest contour points to grid point \((s, t)\) can be reduced by sorting the \((x, y, z)\) contour points within a band by their \( x \)-coordinates. The need for this will become apparent in Section (2.2.3).

An important consideration when using bands is the presence of an "edge effect". When determining the elevation \( E(j, k) \) for the point \((s, t)\) that is close to the band boundary, it is highly probable that the closest contour points will not all lie in the same band. This
Figure 28. Relationship of Bands, Contour Points and Grid Points.
makes it necessary therefore, to also check the contour points in the closest, adjacent band.

The selection of the band width is important as it affects the number of contour points that lie within a band. If narrow bands with few points are used then it will not be sufficient to merely check the two bands as discussed above. Conversely, a wide band with many points will decrease the likelihood of missing a relevant data point. This does however increase the search time within a band thereby creating an tradeoff in efficiency.

The optimal band width will occur when \((s,t)\) is bracketed by two contour lines. Since two bands are always being used, i.e., the band containing \((s,t)\) and the closest band to \((s,t)\), only one contour per band is needed. Correspondingly, the band width would be equal to the average distance (in digitizing units) between contour lines. A factor of safety should, however, be introduced as the average distance between contours could vary substantially within the study area, especially in extremely varied terrain. It is difficult to estimate how many contour lines should be included per band to remove this uncertainty. Although not thoroughly investigated due to limited computer capacity, two contour lines are probably sufficient making the band width twice the average distance between contours.
The other important parameter used to control the number of points in a band is the thinning tolerance. Using the above rationale of having two contours per band, a thinning tolerance equal to one half the band width should be used. For a contour running perpendicular to the orientation of the bands, this thinning tolerance will ensure that no more than two contour points on the contour line will be contained within any one band.

The band method is most useful when memory limits the complexity of the data structure or there is no fast-access storage device, like a disc, that permits quick swapping of search units in and out of memory.

The Grid Data Structure for Searching

The grid data structure is analogous to banding simultaneously in two directions. The grids used are not to be confused with the elevation grid E(j,k) and can be considered as "macro-grids". Using a cell size of w, \((u/w)(v/w)\) cells will be needed. A contour point \((x,y,z)\) is in cell, \(G(h,i)\), if:

\[(h-1)w \leq x < hw \quad \text{and} \quad (i-1)w \leq y < iw\]

The points can be stored in their appropriate cells, \(G(h,i)\), by extending \(G\) to five dimensions: \(G(h,i,x,y,z)\).

The selection of the cell width, \(w\), and the thinning tolerance is made on the same basis as used for the bands.
The "edge effect" is more pronounced in the grid approach as the cells have twice the number of edges. Consequently, it is necessary to search the three closest, adjacent cells to the point \((s,t)\) in addition to the cell that contains \((s,t)\).

The grid data structure requires a fast-access storage device as there are more data structure units to handle than in the band approach.

2.2.3 Generating the Grid of Elevations

The elevation \(E(j,k)\) for the point \((s,t)\) is computed by dividing the area around \((s,t)\) into \(p\) equal regions (e.g., fourths \((p=4)\), sixths \((p=6)\)) and finding the closest point in each region (Figure 29). It is not sufficient to merely locate the \(p\) closest points as this would create a bias if the points are concentrated to one side of \((s,t)\). Increasing the number of regions will require more computational work but the level of accuracy will also increase.

The decision as to which points to check will depend on the type of data format used to retain the contour points; i.e., bands or grids.

If bands are used then it is necessary to determine which band contains \((s,t)\). This band, \(b_i\), is indexed by: \(i=[t/w]+1\).

The position of \((s,t)\) within \(b_i\), with respect to the x-
Figure 29. Using Four Regions to find an Elevation.
coordinate, is easily found using a standard binary search (Conte and deBoor, 1973) as the points in the bands have been previously sorted (Section 2.2.2). The point within band \( b_i \), having the closest \( x \)-coordinate to \( s \), will only lie in one of the \( p \) regions around \( (s,t) \). Letting the subscript of this point within \( b_i \) be \( c \), it is necessary to examine several of the ordered contour points within \( b_i \) that lie to either side of point \( c \) to ensure that all the regions around \( (s,t) \) are represented. In order to maintain an unbiased search, the selection of the number of points to be examined should correspond to the band width and thinning tolerance used. Because the criterion used in defining these two parameters was to have one band width containing either two contour lines or two contour points (depending on whether the contour lines are parallel or perpendicular to the bands) it is necessary to check four contour points on both sides of point \( c \).

To eliminate the "edge effect" it is also necessary to perform the same search as discussed above on the closest, adjacent band. This band is identified by comparing the \( y \)-coordinate of \( (s,t) \) to the midpoint (with respect to \( y \)) of the band \( b_i \). If \( t < w(i-1/2) \) then the closest band is \( b_{i-1} \); otherwise it is \( b_{i+1} \). In total then, 18 contour points must be checked.

If a grid system is used to retain the \( (x,y,z) \) contour points then searching for the closest points becomes significantly easier. The subscripts of the cell, that contains \( (s,t) \), are found from:
\[ h = \lceil \frac{s}{w} \rceil + 1 \quad \text{and} \quad i = \lceil \frac{t}{w} \rceil + 1 \]

All the points in cell \( G(h,i) \) are searched along with the three closest, adjacent cells to \((s,t)\). These three cells are found by comparing \((s,t)\) to the centroid \((w(h-1/2), w(i-1/2))\) of the cell \((h,i)\). For example, if \( s < w(h-1/2) \) and \( t < (w(i-1/2) \) then the three closest cells would be \( G(h-1,i-1), G(h-1,i) \) and \( G(h,i-1) \).

The checking of the \((x,y,z)\) contour points for the closest point in its respective region around \((s,t)\) is the same regardless of whether the grid or band search structure is used. Each contour point examined (18 in total if the band method is used) must first be placed in its appropriate region about \((s,t)\). The region is determined from the angle, \( \theta \), between \((s,t)\) and \((x,y,z)\) (with \(z\) ignored) where:

\[ \theta = \arctangent \left( \frac{y-t}{x-s} \right) \]

The angle \( \theta \) must be converted to a full 360 degree range as most computers use arctangents between plus and minus 90 degrees. The region number, \( r \) \((1 \leq r \leq p)\), is found from:

\[ r = \lceil \frac{\theta}{360/p} \rceil + 1 \]

The distance is computed between \((x,y,z)\) and \((s,t)\) from

\[ d^2 = (x-s)^2 + (y-t)^2 \]

This distance, \( d^2 \), is compared to \( d_{r}^2 \) (the distance between the

---

\( ^8 \) Converting an angle obtained from an arctangent function to a full 360 degree range is accomplished by \( \theta = 180^\circ \) and

- if \( x > s, y > t \) then \( \theta = \theta \)
- if \( x < s, y > t \) then \( \theta = 180 - \theta \)
- if \( x < s, y < t \) then \( \theta = 180 + \theta \)
- if \( x > s, y < t \) then \( \theta = 360 - \theta \)
closest point yet found and \((s,t)\) in region \(r\). If \(d_r^2 > d_r^2\) then \((x,y,z)\) is rejected. Conversely if \(d_r^2 < d_r^2\) then \(d_r^2 = d_r^2\). The elevation of the region is also updated: \(e_r = z\).

When all the appropriate points have been checked the elevation is computed using the distance-squared technique and the closest point in each of the \(p\) regions.

\[
E(j,k) = \frac{e_1/d_1^2 + e_2/d_2^2 + \ldots + e_p/d_p^2}{1/d_1^2 + \ldots + 1/d_p^2}
\]

2.3 Comparison of the Transect and Contour Methods

The transect and contour methods were compared in a manner similar to that described in Section 1.2.1. Contours were traced and the average difference in elevation between the actual and the computed contour elevation was determined for each method (Figure 30). For the contour method only the band technique of computing the \(E(j,k)\) was used as the grid structure exceeded the capability of the computer. In the example shown in Figure 30 the data base was created by digitizing the 100-foot contours only. The error in the transect method was generally quite uniform irrespective of whether the contour being tested had been used to form the data base. The opposite was true for the contour method. It was superior to the transect method only when generating elevations along contour lines that were used to create the data base. As seen from Figure 30, the error in the 2825-foot contour is significantly higher than for the 2800-foot contour. This error could be
<table>
<thead>
<tr>
<th>CONTOUR METHOD</th>
<th>TRANSECT METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEVATION ERROR (FEET)</td>
<td>ELEVATION ERROR (FEET)</td>
</tr>
<tr>
<td>3.00</td>
<td>4.68</td>
</tr>
<tr>
<td>6.91</td>
<td>10.29</td>
</tr>
<tr>
<td>20.34</td>
<td>7.47</td>
</tr>
<tr>
<td>10.50</td>
<td>-10.68</td>
</tr>
<tr>
<td>5.89</td>
<td>-4.01</td>
</tr>
<tr>
<td>6.50</td>
<td>0.12</td>
</tr>
<tr>
<td>-0.96</td>
<td>4.79</td>
</tr>
<tr>
<td>-1.00</td>
<td>-10.78</td>
</tr>
<tr>
<td>5.46</td>
<td>12.50</td>
</tr>
<tr>
<td>0.98</td>
<td>3.50</td>
</tr>
</tbody>
</table>

CONTOUR ELEVATION: 2800.00
NUMBER OF POINTS: 10.00
THE AVERAGE (MEAN) ERROR: 6.15
THE LARGEST ERROR: 20.34
VARIANCE: 39.82
STANDARD DEVIATION: 6.31

CONTOUR ELEVATION: 2800.00
NUMBER OF POINTS: 10.00
THE AVERAGE (MEAN) ERROR: 6.88
THE LARGEST ERROR: 12.50
VARIANCE: 65.49
STANDARD DEVIATION: 8.09

CONTOUR METHOD

ELEVATION ERROR (FEET)
-12.31
-13.32
10.32
12.26
21.39
-17.70
17.44
5.20
-13.58
-5.83

CONTOUR ELEVATION: 2825.00
NUMBER OF POINTS: 10.00
THE AVERAGE (MEAN) ERROR: 12.93
THE LARGEST ERROR: 21.39
VARIANCE: 211.62
STANDARD DEVIATION: 14.55

TRANSECT METHOD

ELEVATION ERROR (FEET)
-1.25
-1.23
0.54
-3.89
-7.50
-6.59
10.29
2.75
-4.33
-8.02

CONTOUR ELEVATION: 2825.00
NUMBER OF POINTS: 10.00
THE AVERAGE (MEAN) ERROR: 4.64
THE LARGEST ERROR: 10.29
VARIANCE: 30.86
STANDARD DEVIATION: 5.56

Figure 30. Comparison of the Accuracy of the Contour and Transect Methods.

(Note: computer output has be re-typed for clarity)
reduced if the number of search regions was increased and/or by retaining a higher percentage of the contour coordinates obtained when digitizing the data base; i.e., thinning fewer points. The other obvious possibility is to digitize more contours. The potential improvement was not tested as machine capacity was limiting.

2.4 Implementation

Both the transect and contour methods have been implemented on the HP 9830A computer. Although the contour method is superior theoretically, it can only be used for small maps or those with limited detail due to limitations of the HP9830A computer; i.e., slow execution speed, lack of a fast access storage device and a small memory. Using a thinning tolerance of 0.20 inches and a complexity similar to that shown in Figure 1, the largest map size that can be effectively accommodated by the contour method is approximately 12 inches square.

The transect method is restricted to one set of 67 lines (i.e., n=66) each with a capacity of 200 contour intersections. There was insufficient memory to allow for implementation of the double transect method.

In the contour method only 2300 \((x,y,z)\) contour points can be retained. To improve search time for locating the closest points when generating the elevations, horizontally oriented
bands are used. To make the implementation possible a fixed number of bands (20) is always used making the band width, \( w \), variable. The bands are stored individually on data tape files and have a capacity of 200 points each. The elevations were determined using four regions.

2.5 A New Method for Data Entry

The major drawback to the entire simulator is entering the data. It is a time consuming, tedious process which will tend to discourage many potential users. One relatively new, promising approach is in the use of orthophotographs. Orthophotographs are created by removing the distortion in ordinary aerial photographs due to the elevation differences of the landscape (Meyer, 1969); the process can be either manual or automated. The distortion is removed by systematically assembling a photographic mosaic of the original aerial photograph. Each small unit making up the mosaic is optically adjusted to a single, common elevation. Inherent in the process therefore is the determination of the landscape elevations. This is done on a regular pattern, typically using a grid or transect approach, with the elevations and their respective \((x, y)\) locations being recorded on a data tape. This data can then be relatively easily converted into the regular grid of elevations used by the terrain simulator by either the transect or contour methods thereby eliminating the digitizing phase.
3.0 Topographic Features

The Topographic Features Module provides a facility for obtaining map overlays and acreage distributions for ground slope, aspect and elevation. The slope, aspect and elevation are determined for each grid cell in the elevation database. This time consuming process is done only once for any data set with the results being stored on data tape.

3.1 Determining the Topographic Features

The slope, aspect and elevation for each grid are computed using the surface equation for a linear plane: \( m_1 x + m_2 y + m_3 z = 1 \). Only three nodes are required for this equation (Section 1.2.2). Inclusion of a fourth node, thereby using all the four cell corner elevations, requires a least-squares statistical fit.

With four nodes, four linearly independent equations can be developed. These can be expressed in matrix notation as \( A m = b \) with \( A, m, b \) defined in Section 1.2.2. The three parameters, \( m \), are solved using least-squares by

\[
m = (A^T A)^{-1} A^T b \tag{3.1}
\]

The solution of the system gives:

\[
z = c x + by \tag{3.2}
\]

with \( c = 1/m_1 \).
$$a = -\frac{m_1}{m_3}$$
$$b = -\frac{m_2}{m_3}$$
which is the equation for determination of all topographic features.

3.1.1 Solving for Average Elevation

The average elevation, $\bar{z}$, is evaluated at the centroid of the cell $(j,k)$. The coordinates of the centroid are:
$$x = (s+u/n/2) \text{ and } y = (t+v/n/2)$$
Substituting these values for $x$ and $y$ into Equation (3.2) yields:
$$\bar{z} = c \cdot a (s+u/n/2) + b (t+v/n/2)$$
which is the average elevation for the cell $(j,k)$.

3.1.2 Solving for the Maximum Surface Slope

The plane, defined by Equation (3.2) (Figure 31), has three intercepts:
- $z$-intercept is $c$ $(x,y=0)$ (point C)
- $x$-intercept is $-c/a$ $(y,z=0)$ (point A)
- $y$-intercept is $-c/b$ $(x,z=0)$ (point B)

The tangent of the angle, $\theta_x$, which is subtended by the plane and the $x$-axis, represents the rate of change (surface slope) of $z$ with respect to $x$. Similarly, the tangent of the angle, $\theta_y$, is the surface slope of $z$ with respect to $y$. The maximum surface slope of $z$ with respect to both $x$ and $y$ is the tangent of $\theta_p$. The angle $\theta_p$ ($\angle CPQ$) is subtended by the plane and the
Equation of the surface plane:
\[ z = c + ax + by \]

Figure 31. The Surface Plane and its Coordinate System.
line segment PO with P lying on AB. If the length of PO is \( i \) then the tangent of \( \angle CPO \) will be \( \frac{c}{i} \). This tangent will be at a maximum when \( i \) is a minimum. This occurs only when PC is the altitude of triangle ABC and PO is the altitude of triangle OAB.

Noting that OP is the altitude of \( \triangle OAB \), the value of \( i \) can be determined from the cosine of \( \angle BOP \) multiplied by the length of \( OB \) which is the y-intercept \( (-c/b) \).

Letting \( \angle BOP = \alpha \) then:

\[
\cos \alpha = \frac{i}{(-c/b)}
\]

Rearranging:

\[
\frac{1}{i} = \frac{-b/c}{\cos \alpha}
\]  \hspace{1cm} (3.3)

Using the relationship \( 1/\cos \alpha = \sqrt{1 + \tan^2 \alpha} \) in Equation (3.3):

\[
\frac{1}{i} = \frac{-b/c}{\sqrt{1 + \tan^2 \alpha}}
\]

The tangent \( \alpha \) can be replaced by \( \frac{a}{b} \)

since \( \angle BOP = \angle BAO \) and \( \tan \alpha = \frac{-c/b}{(-c/a)} = a/b \)

Therefore

\[
\frac{1}{i} = \frac{-b/c}{\sqrt{1 + \frac{a^2}{b^2}}}
\]

and

\[
i = -\frac{c}{\sqrt{a^2 + b^2}}
\]

Hence, \( \tan \theta = \frac{c}{i} = \frac{\sqrt{a^2 + b^2}}{\sqrt{a^2 + b^2}} \) (sign can be ignored)

The maximum slope, in percent, is therefore

\[
\left( \frac{\sqrt{a^2 + b^2}}{i} \right) 100/\text{map scale} \hspace{1cm} (3.4)
\]

3.1.3 Solving for Aspect

The aspect of the surface plane is determined by selecting any point on the ray OP (Figure 31) and then determining the compass quadrant in which it lies. A convenient point is
(a, b, 0) which defines the angle, \( \beta \), of the ray \( \mathbf{OP} \) to be the arctangent of \( \frac{b}{a} \). The point \( (a, b, 0) \) can be easily shown to lie on \( \mathbf{OP} \) by considering the triangle \( OAP \). \( \alpha + \beta = 90 \) and since \( \alpha = \text{arctangent of } \frac{a}{b} \), \( \beta \) must be the arctangent of \( \frac{b}{a} \).

Using Equation (3.2), if \( a > 0 \) then \( \frac{dz}{dx} > 0 \); i.e., when proceeding east (\( x \) increasing) across the surface plane the elevation will also increase. Consequently, by definition, the aspect will be westerly. Extending this logic to the other three quadrants yields a set of relationships opposite to those employed in Section 2.2.3 that can be used to convert \( \beta \) to the full 360 degree aspect range (Figure 32). Namely:

\[
\beta = |\beta| \\
\text{if } a < 0, b < 0 \text{ then } \beta = \beta \\
\text{if } a > 0, b < 0 \text{ then } \beta = 180 - \beta \\
\text{if } a > 0, b > 0 \text{ then } \beta = 180 + \beta \\
\text{if } a < 0, b > 0 \text{ then } \beta = 360 - \beta
\]

The angle \( \beta \) can be converted into one of the eight main aspects (N, NE, E, SE, S, SW, W, NW). For convenience the angle \( \beta \) is rotated by 22.5 degrees thereby shifting the boundaries of the aspect classes onto even 45-degree multiples (Figure 32).
Figure 32. Definition of the Aspect Classes.
The map orientation\(^9\) that was used when digitizing the contours must be accounted for. Assuming the angle of the orientation is positive for counter-clockwise rotations of north then the equation for \(\beta\) is:

\[
\beta = \beta + 22.5 - \text{rotation} \times 360
\]

(360 is added to eliminate negative angles)

To ensure that \(\beta\) is not greater than 360:

\[
\beta = \beta - \lfloor \frac{\beta}{360} \rfloor 360
\]

The angle \(\beta\) can now be converted to the appropriate aspect index (Table VI) by

\[
I = \lfloor \frac{\beta}{45} \rfloor + 1
\]

The value of \(I\) is stored as the aspect index for cell \((j,k)\).

<table>
<thead>
<tr>
<th>TABLE VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINITION OF ASPECT INDICES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>east</td>
</tr>
<tr>
<td>2</td>
<td>northeast</td>
</tr>
<tr>
<td>3</td>
<td>north</td>
</tr>
<tr>
<td>4</td>
<td>northwest</td>
</tr>
<tr>
<td>5</td>
<td>west</td>
</tr>
<tr>
<td>6</td>
<td>southwest</td>
</tr>
<tr>
<td>7</td>
<td>south</td>
</tr>
<tr>
<td>8</td>
<td>southeast</td>
</tr>
</tbody>
</table>

\(^9\) When entering the elevation data base with the transect method it is more accurate to have the map aligned so that the transect lines are perpendicular to the majority of the contours. Consequently the aspects computed for the "table north" will not be the same as for the true north. The map orientation between the true north and the table north must therefore be added to the degree of aspect computed for the surface plane.
3.2 Storage of the Slope, Aspect and Elevation

The topographic features indices are calculated only once and then stored in a three-dimensional matrix \( F(i,j,k) \), with

- \( i=1 \) slope
- \( i=2 \) aspect
- \( i=3 \) elevation

and \((j,k)\) the cell subscripts.

3.3 Output of the Topographic Features

For topographic class maps the class interval, \( c \), defining the various class ranges (Figure 5), must be declared by the user. In the case of an aspect map, \( c \) is automatically set to one in order to produce each of the eight aspects. The class index is determined for each cell and stored in a \((n+1)^2\) matrix \( P \) (for subsequent plotting of the map overlay) by

\[
P(j,k) = \lfloor F(i,j,k)/c \rfloor + 1
\]

For example, if the map overlay was desired in 30-percent slope classes then \( c=30 \). If for cell \((j,k)\) the slope is 48 percent, then the cell is in the 30 to 59 percent slope range, which is the second class. Hence \( P(j,k)=2 \).

A tally can be kept of the frequency of the various classes in a vector \( T \) with the class number, \( P(j,k) \), being the subscript. Continuing with the above example, it is necessary to increment the frequency counter for class 2; i.e., increment
To produce a combination map the topographic features for each cell are checked against the allowable ranges set by the user (Figure 6). The corresponding elements of $P$ (the same plotting matrix used for the class maps) will be either true (1) or false (0) depending whether the cell has the specified slope, aspect and/or elevation. The combination overlay mapping could easily be extended to include other forms of data; for example, soils, land-use, wildlife and wood inventory.

The plotting of the overlay map is done in two phases. First, the boundaries of the various classes are plotted; then, area labels are added. The overlay is most efficiently produced by using a grid map (Figure 6) with the smallest plotting unit corresponding to one grid cell of the elevation data base. The class boundary lines will therefore be either vertical ("north-south") or horizontal ("east-west"). The boundary lines will only be needed to separate cells of different classes. The plotting algorithm is given in the flowchart shown in Figure 33.

3.4 Implementation

Memory restrictions made it impossible to retain both the elevation data base and the complete set of topographic features in computer memory at the same time. The topographic features matrix was consequently reduced in size to a grid of $34 \times 34$. 

$T(P(j,k))$ by one where $P(j,k)=2$. 

note: plot(x, y, d) moves the plotting pen to (x, y) with pen down (d).

START

plot boundary

Do east/west lines first; do lines in pairs; 1st east

- plot east line first; do west line on return trip

penup + true

1

x=d

x<66

x+x+1

no

cell above the same?

penup?

yes

no

no action required; cells are the same.

plot (x-1, y, d) penup true

penup? no

P(x, y) = P(x, y+1)?

no

plot(x, y, up) penup = false

yes

complete rest of line; plot(66, y, d)

penup?

yes

finished east line.

finished pair.

y+y+1

2

do return trip for the west line; method same as for east direction.

x=66; y+y+1

y=65?

no

do north/south lines.

yes

end

Figure 33. Flowchart for Plotting Map Overlays.
(1156) cells. Each of these topographic cells was therefore four times the size of the elevation data base cell and contained nine elevations. The least-squares fit, Equation (3.1), used to generate the three parameters of the surface plane, was done using nine equations instead of the four as described in Section 3.1.1. This resulted in generally less precision as the variance increased because the three-parameter, flat plane was being fitted to five more elevation points.

To conserve storage, the topographic features for each cell were combined into one 16-bit word. This was achieved by limiting the elevation to 100-foot classes with a maximum elevation of 12,700 feet. This resulted in 128 classes using seven of the 16 bits. Similarly, slope was restricted to five-percent intervals to a maximum of 165 percent, requiring five bits to represent the 32 classes. Aspect requires nine classes and four bits with the ninth class for flat areas (defined as those under five percent slope). The overlay maps were done using the algorithm in Figure 33.
4.0 Yarding Location and Setting Design

The prime function of the Yarding Location and Setting Design Module is to assess the physical feasibility of various cable systems in different locations. This is achieved by comparing the loadpath (the path the logs follow when yarded to the landing) to the ground profile. If the two intersect such that the front end of the logs can not be clear of the ground then the backspar location is probably infeasible. The major problem, then, is to determine the loadpath.

Theory of Cable Mechanics

Cable theory is well documented in the literature. The USFS has done extensive research in this area (Carson 1970, 1971, 1975). Recently cable theory has been examined at UBC (Appendix C).

A different mathematical formulation is required for each cable system depending on its geometry (number and configuration of the lines) and method of operation. For a particular system, several different models can be developed depending on the governing physical assumptions. The most theoretically correct model is called a catenary model. This approach assumes that the cable weight is distributed uniformly along its length. The
solution is complex and slow to evaluate and is therefore not suited for frequent use on a relatively slow computer. It does however serve as an effective benchmark for other loadpath models.

There are two models that are useful for the terrain simulator level of analysis.

(1) Parabolic Approximation.

(2) Percent Deflection Rule.

4.1 Parabolic Model for Approximating the Loadpath

The basic theory for the parabolic model is developed in Appendix C. The parabolic model differs from the catenary approach because the cable weight is assumed to be distributed uniformly on the chord and not on the cable itself as for the catenary approach. The cable system discussed here is a running skyline (Figure 34). The loadpath is defined by Equation 6 from Appendix C:

\[
y = x (L-x) \left(2R+x\left(w_1 + w_3\right) + (L-x) 2w_2\right) / (4HL) - Ex/L \quad \ldots \ldots \ldots (4.1)
\]

with

- \(y\) = \(y\)-coordinate of the load position
- \(x\) = \(x\)-coordinate of the load position
- \(L\) = span length (horizontal)
- \(R\) = weight of load plus carriage
- \(H\) = horizontal tension in the haulback cable segment
Figure 34. Schematic Drawing of a Running Skyline System.
between the carriage and the backspar

\( E = \) elevation difference of the spars

\( w_1 = \) effective line weight on the subchord of the haulback segment between the landing and the carriage

\( w_2 = \) effective line weight on the subchord of the haulback segment between the carriage and the backspar

\( w_3 = \) effective line weight on the subchord of the mainline plus slackpuller

The effective line weights, \( w_1, w_2, w_3 \) are found using 

\[ w_i = \frac{w_i}{\cos \theta_i} \]

where \( w_i \) is the actual cable weight and \( \theta_i \) is the angle of the subchord of the respective cable segment. The horizontal tension, \( H \), is the governing parameter of the cable system and is defined by Equation 7 of Appendix C:

\[ H = T \cos \left( \arctan \left( \frac{E+y}{L-x} + \frac{w_2(L-x)}{2H} \right) \right) \]  

(4.2)

The solution for \( y \) is non-trivial as \( y \) is contained in both sides of Equation (4.1). Because a closed form solution is not possible an iterative approach is needed. Newton's method for solving non-linear equations is employed because it is easy to implement and converges quickly and dependably.

Following is a description of the iterative solution of \( y \) in Equation (4.1):

The basic Newton iterative relationship is:

\[ y_{i+1} = y_i - \frac{f(y_i)}{f'(y_i)} \]
To simplify development, some common values are established.

Let \( A = x(L-x)(2R+x(w_1 + w_2) + 2w_2(L-x))/(4L) \)

\( B = E_0/L \)

\( U = (E+y)/(L-x) + w_2(L-x)/(2H) \)

\( M = T \cos(\arctan(U)) \)

Therefore Equation (4.1) can be rewritten as:

\( y = A/M - B \)

This can be rearranged to the form \( f(y) = 0 \):

\[ f(y) = y + B - A/M = 0 \]

The problem is now to find \( f'(y) \), the derivative of \( f(y) \). This is difficult since the \( w_i \) are functions of the sag \( y \) through the relationship \( w_i = w_i'/\cos \theta_i \) with \( \theta_i \) a function of \( y \). The horizontal tension, \( H \), also is a function of sag, \( y \).

Intuitively, changes in \( w_i \) and \( H \) with respect to \( y \) are relatively small compared to the remaining components of the derivative which concern the geometry of the system. The approach taken, therefore, was to simplify the derivative of \( f(y) \) by assuming \( w_i \) and \( H \) are independent of \( y \). This results in an approximation to the true \( f'(y) \) which at worst will retard slightly the rate of convergence of Newton's iterative relationship. The correction factor \( f(y)/f'(y) \) will still converge to zero as \( y_i \) converges to \( y \), providing \( f(y) \) is correctly formulated. It is important to note that although \( w_i \) and \( H \) are assumed independent of \( y \) they must be recalculated at the beginning of each iteration in order for \( f(y) \) to be correct.
The value of \( H \) can be obtained from Equation (4.2). Using this simplification the derivative of \( f(y) \) becomes \( f'(y) = (y + B - A/M) / dy \)

Thus:
\[
f'(y) = 1 - A \frac{dm - 1}{dy} = 1 - A/m^2 \frac{dm}{dy}
\]

Now \( \frac{dm}{dy} = -T \sin \left( \arctan(U) \right) \frac{d(\arctan(U))}{dy} \)

and \( \frac{d(\arctan(U))}{dy} = \frac{1}{(1 + U^2)(L - x)} \)

Therefore \( M' = -T \sin(\arctan(U)) / \{(1 + U^2)(L - x)\} \)

and \( f'(y) = 1 - AM'/M^2 \)

The iteration equation becomes,
\[
y_{i+1} = y_i + \frac{(y_i + B - A/M)}{(1 + AM'/M^2)}
\]

which is terminated when
\[
|y_{i+1} - y_i| \leq \text{tolerance}
\]

Initial values are required for both \( y \) and \( H \). Suitable values are

\( H = T \cos(\arctan(E/L)) \) and \( y = 0 \)

The slope \( E/L \) acts as an estimate of the cable slope of the haulback at the tailhold. The initial guess of \( y = 0 \) is sufficient as Newton's method is extremely insensitive to starting values for this application. The convergence properties are shown in Figure 35 which are typical of most situations. Normally only two to three iterations are required to converge within a tolerance of 0.1 foot.
Figure 35. Loadpath Solution for a Running Skyline System Using the Parabolic Model and Newton Iteration.

(Note: computer output has been re-typed for clarity)
4.1.1 Comparison to the Catenary Method

The parabolic model provides results that are extremely close to those obtained from a catenary solution. An example illustrating this is given in Table VII which shows the errors in tensions and load positions for varying values of E and L at midspan. This example is for a running skyline system (Madill 052 tension skidder) with a 35000 pound load and a maximum allowable haulback tension of 56500 pounds (the approximate line pull capacity of the yarder interlock). The ability of the parabolic to closely approximate the catenary has also been verified in field studies (Guimier, 1977).

The differences between the two models will always be small providing the subchords approximate the cables between the landing and the carriage and the carriage and the tailhold. This condition will minimize the effect of the underlying assumption in the parabolic model that the cable weight is distributed on the subchords and not on the cable itself. In most cable yarding setups the subchords do approximate the cables unless the system is grossly underloaded.

4.1.2 Improvement of the Iterative Method

Although no other method converges more quickly than Newton's method, the solution is still fairly slow especially if the iteration must be invoked at several points along the span. One possible improvement, when computing at several points in
TABLE VII
ERRORS FROM THE PARABOLIC MODEL WITH THE CARRIAGE AT MIDSPAN FOR A MADILL 052 TENSION SKIDDER

LOAD INCLUDING CARRIAGE = 35000 LB.
TENSION (HAULBACK) AT INTERLOCK = 56500 LB.
WEIGHT OF HAULBACK/FOOT = 2.34 LB.
WEIGHT OF MAINLINE+SLACKPULLER = 4.13 LB.

SPAN = 1400 FT.

ERRORS IN TENSIONS AT THE CARRIAGE

<table>
<thead>
<tr>
<th>CHORD SLOPE%</th>
<th>ERROR IN LOAD POSITION (FT.)</th>
<th>HAULBACK LNDG-CARR (LB.)</th>
<th>HAULBACK CARR-TAIL (LB.)</th>
<th>MAINLINE + SLACKPULLER (LB.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.0686</td>
<td>0.34</td>
<td>-5.63</td>
<td>-11.60</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL COMPONENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>0.79</td>
<td>4.23</td>
<td>-8.79</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>-0.17</td>
<td>-0.25</td>
<td>-3.84</td>
</tr>
<tr>
<td>50</td>
<td>0.0053</td>
<td>-0.20</td>
<td>-1.82</td>
<td>-3.44</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>-0.74</td>
<td>2.55</td>
<td>-4.18</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>-0.05</td>
<td>0.04</td>
<td>-1.81</td>
</tr>
<tr>
<td>0</td>
<td>-0.0132</td>
<td>0.08</td>
<td>0.02</td>
<td>-0.03</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>-0.33</td>
<td>-0.30</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>0.04</td>
<td>0.20</td>
<td>0.41</td>
</tr>
<tr>
<td>-50</td>
<td>-0.0249</td>
<td>-0.63</td>
<td>-0.90</td>
<td>-1.13</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>1.23</td>
<td>-2.82</td>
<td>4.42</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>0.09</td>
<td>-0.53</td>
<td>0.63</td>
</tr>
<tr>
<td>-100</td>
<td>-0.0542</td>
<td>-2.41</td>
<td>-3.46</td>
<td>-4.54</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>2.20</td>
<td>-4.59</td>
<td>7.28</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>0.13</td>
<td>-0.05</td>
<td>2.56</td>
</tr>
</tbody>
</table>
the span, is to use superior initial guesses. This can be achieved by use of a difference table.

A difference table is a tabular expression of an interpolating polynomial (Conte and deBoor, 1973). Using the loadpath results shown in Figure 35 a difference table can be constructed and is shown in Table VIII. The first column of

| TABLE VIII |
| DIFFERENCE TABLE FOR SAG (Y) |
|---|---|---|---|
| x  | y  | D^1 | D^2 |
| 100 | 20.18 | 13.89 | |
| 200 | 34.07 | 7.53 | -6.36 |
| 300 | 41.60 | 1.05 | -6.48 |
| 400 | 42.65 | -5.61 | -6.66 |
| 500 | 37.04 | | |

differences approximate the first derivative; second column of differences approximate the second derivative, etc. As seen from Table VIII, the second difference is almost constant indicating that the function can be approximated by a parabola.

Values for the function in the difference table can be extrapolated by using the approximate derivative values:

\[
y_{i+1} = y_i + D_i^1 + D_i^2 \quad (D: \text{"delta"})
\]

where

\[
D_i^1 = y_i - y_{i-1}
\]

\[
D_i^2 = D_i^1 - D_{i-1}^1
\]

\[
= (y_i - y_{i-1}) - (y_{i-1} - y_{i-2})
\]

The above equation for \( y_{i+1} \) can be simplified to yield:
\[ y_{i+1} = 3y_i - 3y_{i-1} + y_{i-2} \]

For example, an estimate for \( y \) can be made at \( x = 500 \) by using the second to last diagonal of the table:

\[ y = 42.65 + 1.05 - 6.48 = 37.22 \]

This represents an error of only 0.18 foot. Employing the difference table to compute initial values of \( y \) will virtually guarantee convergence in one iteration thereby almost halving the computational time required.

4.2 Percent Deflection Rule for Approximating the Loadpath

The percent rule method is applicable to any cable system. The percent deflection (for a system) is defined here as the minimum attainable midspan load displacement from the chord divided by the length of the chord (Figure 36) and is usually known through local experience. The application of the percent rule, as currently used in the forest industry for predicting loadpaths, has two serious drawbacks:

(a) One universal deflection value is used for one system regardless of the yarding conditions.

(b) The percent deflection defines the load position only at midspan providing little information about the rest of the span.

The effect of applying one deflection value for all yarding conditions is clearly illustrated (for a running skyline system) in Table IX. The deflections given are computed using a
loadpath defined by the interpolating polynomial \( y = x \left( \frac{(4Y_m - E)}{L} + 2x \frac{(E - 2Y_m)}{L^2} \right) \)

node 1: \((0,0)\) (landing spar)

node 2: \((L/2, Y_m)\) (midspan)

node 3: \((L, E)\) (backspar)

**Figure 36.** Using an Interpolating Polynomial to Predict the Loadpath.
### Table IX

**Deflection at Midspan for Madill 052 Tension Skidder**  
*(Chord Definition)*

Load including carriage = 35000 lb.  
Tension (haulback) at interlock = 56500 lb.  
Height of haulback/foot = 2.34 lb.  
Height of mainline+slackpuller = 4.13 lb.

<table>
<thead>
<tr>
<th>Chord Slope%</th>
<th>800</th>
<th>1000</th>
<th>1200</th>
<th>1400</th>
<th>1600</th>
<th>1800</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>9.4</td>
<td>9.6</td>
<td>9.7</td>
<td>9.9</td>
<td>10.0</td>
<td>10.2</td>
<td>10.4</td>
</tr>
<tr>
<td>90</td>
<td>9.3</td>
<td>9.5</td>
<td>9.6</td>
<td>9.8</td>
<td>9.9</td>
<td>10.1</td>
<td>10.3</td>
</tr>
<tr>
<td>80</td>
<td>9.2</td>
<td>9.4</td>
<td>9.5</td>
<td>9.7</td>
<td>9.8</td>
<td>10.0</td>
<td>10.1</td>
</tr>
<tr>
<td>70</td>
<td>9.2</td>
<td>9.3</td>
<td>9.4</td>
<td>9.6</td>
<td>9.7</td>
<td>9.9</td>
<td>10.0</td>
</tr>
<tr>
<td>60</td>
<td>9.1</td>
<td>9.2</td>
<td>9.3</td>
<td>9.5</td>
<td>9.6</td>
<td>9.8</td>
<td>9.9</td>
</tr>
<tr>
<td>50</td>
<td>9.0</td>
<td>9.1</td>
<td>9.2</td>
<td>9.4</td>
<td>9.5</td>
<td>9.7</td>
<td>9.8</td>
</tr>
<tr>
<td>40</td>
<td>8.9</td>
<td>9.0</td>
<td>9.1</td>
<td>9.3</td>
<td>9.4</td>
<td>9.5</td>
<td>9.7</td>
</tr>
<tr>
<td>30</td>
<td>8.7</td>
<td>8.9</td>
<td>9.0</td>
<td>9.1</td>
<td>9.3</td>
<td>9.4</td>
<td>9.5</td>
</tr>
<tr>
<td>20</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
<td>9.3</td>
<td>9.4</td>
</tr>
<tr>
<td>10</td>
<td>8.5</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
<td>9.3</td>
</tr>
<tr>
<td>0</td>
<td>8.4</td>
<td>8.5</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
</tr>
<tr>
<td>-10</td>
<td>8.3</td>
<td>8.4</td>
<td>8.5</td>
<td>8.7</td>
<td>8.8</td>
<td>8.9</td>
<td>9.1</td>
</tr>
<tr>
<td>-20</td>
<td>8.2</td>
<td>8.3</td>
<td>8.4</td>
<td>8.6</td>
<td>8.7</td>
<td>8.9</td>
<td>9.0</td>
</tr>
<tr>
<td>-30</td>
<td>8.1</td>
<td>8.2</td>
<td>8.3</td>
<td>8.5</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
</tr>
<tr>
<td>-40</td>
<td>8.0</td>
<td>8.1</td>
<td>8.3</td>
<td>8.4</td>
<td>8.6</td>
<td>8.7</td>
<td>8.9</td>
</tr>
<tr>
<td>-50</td>
<td>7.9</td>
<td>8.1</td>
<td>8.2</td>
<td>8.4</td>
<td>8.5</td>
<td>8.7</td>
<td>8.9</td>
</tr>
<tr>
<td>-60</td>
<td>7.9</td>
<td>8.0</td>
<td>8.2</td>
<td>8.4</td>
<td>8.5</td>
<td>8.7</td>
<td>8.9</td>
</tr>
<tr>
<td>-70</td>
<td>7.8</td>
<td>8.0</td>
<td>8.2</td>
<td>8.3</td>
<td>8.5</td>
<td>8.7</td>
<td>8.9</td>
</tr>
<tr>
<td>-80</td>
<td>7.8</td>
<td>8.0</td>
<td>8.2</td>
<td>8.4</td>
<td>8.5</td>
<td>8.7</td>
<td>8.9</td>
</tr>
<tr>
<td>-90</td>
<td>7.8</td>
<td>8.0</td>
<td>8.2</td>
<td>8.4</td>
<td>8.6</td>
<td>8.8</td>
<td>9.0</td>
</tr>
<tr>
<td>-100</td>
<td>7.8</td>
<td>8.0</td>
<td>8.2</td>
<td>8.4</td>
<td>8.6</td>
<td>8.8</td>
<td>9.0</td>
</tr>
<tr>
<td>-90</td>
<td>9.2</td>
<td>9.5</td>
<td>9.7</td>
<td>10.0</td>
<td>10.2</td>
<td>10.5</td>
<td>10.8</td>
</tr>
<tr>
<td>-100</td>
<td>9.7</td>
<td>10.0</td>
<td>10.3</td>
<td>10.5</td>
<td>10.8</td>
<td>11.1</td>
<td>11.4</td>
</tr>
</tbody>
</table>
catenary model. The average midspan deflection in the Table IX is approximately nine percent. The result of using a single deflection value for all conditions is clearly indicated since actual deflections range from 7.8 to 10.4 percent causing a maximum error of 33.6 feet. These errors are of sufficient magnitude to cause concern. This problem can be alleviated by using deflection tables similar to the one in Table IX (Lysons and Mann, 1967).

Often the percent deflection is defined not by the chord length but by the span length. The application of a universal deflection value defined in this manner yields even greater errors as the effect of the elevation difference of the spars is totally ignored. The large variations of the deflections calculated in this fashion are shown in Table X.

The other drawback of the percent rule is its inability to predict the loadpath for the entire span. This can be overcome by using an interpolating polynomial. There are three known points on the loadpath: landing spar, midspan load position (Ym) and tailhold (Figure 36). These can be used to generate a second degree polynomial.

Starting with the general expression of a second degree polynomial:

\[ p(x) = a_0 + a_1 (x-x_0) + a_2 (x-x_0)(x-x_1) \]

The coefficients \( a_0, a_1, a_2 \) can be found by recursively using this expression and the three nodes (Shampine and Allen, 1973).
TABLE X

DEFLECTION AT MIDSPAN FOR MADILL 052 TENSION SKIDDER
(SPAN DEFLECTION)

LOAD INCLUDING CARRIAGE = 35000 LB.
TENSION (HAULBACK) AT INTERLOCK = 56500 LB.
WEIGHT OF HAULBACK/FOOT = 2.34 LB.
WEIGHT OF MAINLINE+SLACKPULLER = 4.13 LB.

<table>
<thead>
<tr>
<th>CHORD SLOPE%</th>
<th>SPAN - FEET</th>
<th>800</th>
<th>1000</th>
<th>1200</th>
<th>1400</th>
<th>1600</th>
<th>1800</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td>13.3</td>
<td>13.5</td>
<td>13.8</td>
<td>14.0</td>
<td>14.2</td>
<td>14.4</td>
<td>14.7</td>
</tr>
<tr>
<td>90</td>
<td></td>
<td>12.5</td>
<td>12.8</td>
<td>13.0</td>
<td>13.2</td>
<td>13.4</td>
<td>13.6</td>
<td>13.8</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>11.8</td>
<td>12.0</td>
<td>12.2</td>
<td>12.4</td>
<td>12.6</td>
<td>12.8</td>
<td>13.0</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>11.2</td>
<td>11.4</td>
<td>11.5</td>
<td>11.7</td>
<td>11.9</td>
<td>12.1</td>
<td>12.2</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>10.6</td>
<td>10.7</td>
<td>10.9</td>
<td>11.1</td>
<td>11.2</td>
<td>11.4</td>
<td>11.6</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>10.0</td>
<td>10.2</td>
<td>10.3</td>
<td>10.5</td>
<td>10.6</td>
<td>10.8</td>
<td>11.0</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>9.5</td>
<td>9.7</td>
<td>9.8</td>
<td>10.0</td>
<td>10.1</td>
<td>10.3</td>
<td>10.4</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>9.1</td>
<td>9.3</td>
<td>9.4</td>
<td>9.5</td>
<td>9.7</td>
<td>9.8</td>
<td>10.0</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>8.8</td>
<td>8.9</td>
<td>9.1</td>
<td>9.2</td>
<td>9.3</td>
<td>9.5</td>
<td>9.6</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>8.5</td>
<td>8.7</td>
<td>8.8</td>
<td>8.9</td>
<td>9.1</td>
<td>9.2</td>
<td>9.3</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>8.4</td>
<td>8.5</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
</tr>
<tr>
<td>-10</td>
<td></td>
<td>8.3</td>
<td>8.4</td>
<td>8.6</td>
<td>8.7</td>
<td>8.8</td>
<td>9.0</td>
<td>9.1</td>
</tr>
<tr>
<td>-20</td>
<td></td>
<td>8.3</td>
<td>8.5</td>
<td>8.6</td>
<td>8.7</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
</tr>
<tr>
<td>-30</td>
<td></td>
<td>8.4</td>
<td>8.6</td>
<td>8.7</td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
<td>9.3</td>
</tr>
<tr>
<td>-40</td>
<td></td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>9.1</td>
<td>9.2</td>
<td>9.4</td>
<td>9.6</td>
</tr>
<tr>
<td>-50</td>
<td></td>
<td>8.9</td>
<td>9.0</td>
<td>9.2</td>
<td>9.4</td>
<td>9.6</td>
<td>9.7</td>
<td>9.9</td>
</tr>
<tr>
<td>-60</td>
<td></td>
<td>9.2</td>
<td>9.4</td>
<td>9.6</td>
<td>9.7</td>
<td>9.9</td>
<td>10.1</td>
<td>10.3</td>
</tr>
<tr>
<td>-70</td>
<td></td>
<td>9.6</td>
<td>9.8</td>
<td>10.0</td>
<td>10.2</td>
<td>10.4</td>
<td>10.6</td>
<td>10.8</td>
</tr>
<tr>
<td>-80</td>
<td></td>
<td>10.0</td>
<td>10.2</td>
<td>10.5</td>
<td>10.7</td>
<td>10.9</td>
<td>11.2</td>
<td>11.4</td>
</tr>
<tr>
<td>-90</td>
<td></td>
<td>10.5</td>
<td>10.7</td>
<td>11.0</td>
<td>11.3</td>
<td>11.5</td>
<td>11.8</td>
<td>12.0</td>
</tr>
<tr>
<td>-100</td>
<td></td>
<td>11.0</td>
<td>11.3</td>
<td>11.6</td>
<td>11.9</td>
<td>12.2</td>
<td>12.5</td>
<td>12.8</td>
</tr>
</tbody>
</table>
Using the first node \((p_0 = 0, x_0 = 0)\) (landing)

\[ p(x_0) = a_0 \]

Therefore

\[ a_0 = p_0 = 0 \]

Adding the second node \((p_1 = Y_m, x_1 = L/2)\) (midspan)

\[ p(x_1) = a_0 + a_1 (x_1 - x_0) \]

Therefore

\[ a_1 = \frac{(p_1 - a_0)}{(x_1 - x_0)} \]

\[ a_1 = \frac{2Y_m}{L} \]

Finally, using the third node \((p_2 = E, x_2 = L)\) (backspar)

\[ p(x_2) = a_0 + a_1 (x_2 - x_0) + a_2 (x_2 - x_0) (x_2 - x_1) \]

Therefore

\[ a_2 = \frac{(p_2 - a_0 - a_1 (x_2 - x_0))}{(x_2 - x_0) (x_2 - x_1)} \]

\[ a_2 = \frac{2(E-2Y_m)}{L^2} \]

Substituting the values of \(a_0, a_1, a_2\) into the general polynomial and simplifying yields

\[ y = x(\frac{4Y_m-E}{L} + 2x(\frac{E-2Y_m}{L^2})) \] (4.3)

Evaluation of this equation is easy but can be made simpler by converting it to a difference table form. Assuming the load positions are to be determined at regular intervals of \(I\), the recursive relationship is:

\[ D^1_i = D^1_{i-1} + D^2 \]

\[ y^1_i = y^1_{i-1} + D^1 \]

The initial \(D^1_1\) is set to \(y^1_1\) which is calculated from Equation (4.3). The \(D^2\) is the second derivative which is constant. It is found by differentiating Equation (4.3) twice to yield:

\[ D^2 = I^2 \frac{4(E-2Y_m)}{L^2} \]
The results from this interpolating polynomial compare favourably to those obtained from catenary model (Table XI) providing the midspan deflection is known. These errors are relatively insignificant considering the accuracy of the 25-foot contour maps that are generally used. The errors shown in the table were determined at quarter span by first computing the midspan deflection using the catenary model. Then, applying Equation (4.3), the load position at quarter span was determined and compared to the corresponding load position derived from the catenary model. It should be noted that the maximum error will not occur exactly at quarter or three-quarter span (the order of magnitude of the error at quarter and three-quarter span is virtually the same) but will vary slightly depending on the operating conditions.

6.3 Implementation

The frequent use of the loadpath calculations dictated the use of the percent rule because the slow execution speed of the computer made computation times for catenary and parabolic methods excessive. The ground profile is determined at 31 points along the initially defined span. The load positions are then computed at these points using Equation (4.3) and a difference table. The loadpaths generated are reasonably accurate (Table XI) as long as the user employs deflections from a table like that given in Table IX.


**TABLE X1**

PERCENT RULE ERROR AT QUARTER SPAN FOR A MADILL 052 TENSION SKIDDER

LOAD INCLUDING CARRIAGE = 35000 LB.
TENSION (HAULBACK) AT INTERLOCK = 56500 LB.
WEIGHT OF HAULBACK/FOOT = 2.34 LB.
WEIGHT OF MAINLINE+SLACKPULLER = 4.13 LB.

<table>
<thead>
<tr>
<th>CHORD SLOPE%</th>
<th>SPAN - FEET</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 1000 1200 1400 1600 1800 2000</td>
<td></td>
</tr>
<tr>
<td>(ERRORS IN FEET)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>-5.6</td>
</tr>
<tr>
<td>90</td>
<td>-5.0</td>
</tr>
<tr>
<td>80</td>
<td>-4.5</td>
</tr>
<tr>
<td>70</td>
<td>-4.0</td>
</tr>
<tr>
<td>60</td>
<td>-3.5</td>
</tr>
<tr>
<td>50</td>
<td>-3.0</td>
</tr>
<tr>
<td>40</td>
<td>-2.6</td>
</tr>
<tr>
<td>30</td>
<td>-2.1</td>
</tr>
<tr>
<td>20</td>
<td>-1.7</td>
</tr>
<tr>
<td>10</td>
<td>-1.2</td>
</tr>
<tr>
<td>0</td>
<td>-0.8</td>
</tr>
<tr>
<td>-10</td>
<td>-0.4</td>
</tr>
<tr>
<td>-20</td>
<td>-0.0</td>
</tr>
<tr>
<td>-30</td>
<td>-0.4</td>
</tr>
<tr>
<td>-40</td>
<td>0.7</td>
</tr>
<tr>
<td>-50</td>
<td>1.1</td>
</tr>
<tr>
<td>-60</td>
<td>1.4</td>
</tr>
<tr>
<td>-70</td>
<td>1.7</td>
</tr>
<tr>
<td>-80</td>
<td>2.1</td>
</tr>
<tr>
<td>-90</td>
<td>2.4</td>
</tr>
<tr>
<td>-100</td>
<td>2.7</td>
</tr>
</tbody>
</table>
Actual yarding roads to be used in harvesting cannot be projected as there is too much uncertainty about the ground shape due to the generally low accuracy of the available contour maps. The effectiveness of this module is therefore limited to general development planning.
5.0 Road Location

The basic component of the Road Location Module is the projection of roads locations. This is accomplished in either 'manual' or 'automatic' mode.

5.1 Manual Road Projection

The road location is entered via the digitizer. The only computation required is to test whether the grade between successive trial points is within allowable adverse and favourable limits.

Let \((x_i, y_i, z_i)\) be the coordinates and \(z_i\) the elevation of the last recorded point of the location. Then a new point \((x, y, z)\) is acceptable if:

\[-|a| \leq 100(z-z_i)/(\text{mapscale} \sqrt{(x-x_i)^2+(y-y_i)^2}) \leq |f|\]

where \(|a|\) and \(|f|\) are the maximum allowable adverse and favourable road grades.

If the above is true then the newest point on the road location is:

\[x_{i+1} = x \quad \text{and} \quad y_{i+1} = y\]

If the grade limits have been violated then a suitable
warning is given and a replacement point \((x_r, y_r)\) is expected. An override (i.e. accept \((x, y)\) irrespective of the slope) can be incorporated by testing if the two points are within a certain distance of each other:

\[
\text{use } (x, y) \text{ if } (x_r - x)^2 + (y_r - y)^2 \leq \text{tolerance}^2
\]

The coordinates and associated elevations can be stored in three vectors \(X, Y\) and \(Z\). To permit several roads in memory at one time a fourth vector, \(N\), can be added which contains the road numbers for the corresponding elements in \(X, Y\) and \(Z\).

5.2 Automatic Road Projection

The technique locates, in the horizontal plane, the average grade between the start and end points by searching in successive intervals towards the finish. The method provides a heuristic solution and is not guaranteed to be optimal.

Let the starting and finishing points be \((x_i, y_i, z_i)\) with \((i=0)\) and \((x_f, y_f, z_f)\), respectively. The average grade \(g\) between the two is given by:

\[
g = 100 \frac{(z_f - z_i)}{\text{mapscale} \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}}
\]

An immediate check on feasibility can be made. If \(g\) exceeds the grade limits \((a, f)\) then the process is terminated.

The searching for the grade \(g\) will be done at a distance of
L from \((x_i, y_i)\), along the line \(\overline{AB}\) (Figure 37). The line \(\overline{AB}\) is constructed perpendicular to the line between \((x_i, y_i)\) and \((x_f, y_f)\). The intersection of the two lines is at \((x_s, y_s)\):

\[
x_s = x_i + L \cos \theta \quad \text{and} \quad y_s = y_i + L \sin \theta
\]

with \(\theta = \arctan \left( \frac{y_i - y_f}{x_i - x_f} \right)\)

Searching proceeds along \(\overline{AB}\) for the point \((x, y)\), such that the grade between \((x, y)\) and \((x_i, y_i)\) is equal to the average grade \(g\).

Points along \(\overline{AB}\) can be expressed as a distance \((d)\) from \((x_s, y_s)\):

\[
x = x_s + d \cos (\theta + 90) \quad \text{and} \quad y = y_s + d \sin (\theta + 90) \quad \cdots \cdots \cdots \cdots \cdots (7.1)
\]

The grades between these points and \((x_i, y_i)\) can therefore be expressed as a function of \(d\). When the correct value of \(d\) is found the resultant grade \(s\) will equal \(g\). Hence:

\[
f(d) = s - g = 0 \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (7.2)
\]

This is a non-linear equation in \(d\) and can be solved by using secant iteration.

Let \(d_0\) and \(d_1\) be the first two, arbitrarily chosen, guesses and \(f_0\) and \(f_1\) the respective function values of Equation (7.2). Subsequent estimates of \(d\) can be made by

\[
d_{j+1} = d_j - f_j (d_{j} - d_{j-1}) / (f_{j} - f_{j-1})
\]

The iteration can be stopped when: \( \left| d_{j+1} - d_j \right| < \text{tolerance} \). The value of \(f_{j+1}\) should be near zero. The method works well when the ground profile along \(\overline{AB}\) is relatively uniform; in irregular
Figure 37. Planometric View of 'Automatic' Road Location.
terrain the results are highly unstable. An iteration counter provides a safeguard against non-convergence.

The coordinate location of \((x,y)\) for the final iteration value of \(d\) is found using Equation (7.1) and becomes the new starting point of the road:

\[
x_{i+1} = x \quad \text{and} \quad y_{i+1} = y
\]

The process is repeated until the endpoint of the road is reached.

There are three parameters that control the location process (Figure 38):

1. The distance, \(L\), that \(\overline{AB}\) is from \((x_i,y_i)\). This value controls the number of steps required to locate the road and should be chosen to ensure that no major breaks in the topography are missed. A suitable value is the average horizontal distance between contours. This can be difficult to estimate as there is often substantial variation within a map.

2. The distance, \(d\), along \(\overline{AB}\). This controls the flexibility of the route; a large value will allow the road to wander. Through experimentation it was concluded that \(d\) should not be greater than twice the value of \(L\).

3. The stopping tolerance. Tolerance should be fine enough to assure that the proper \(d\) \((f(d)=s-g=0)\) has been located but too small a value will require too many iterations. A
Figure 38. The Effect of the Controlling Parameters on 'Automatic' Road Location
suitable tolerance value was found to be five percent of the maximum allowable $d$, as specified in (2) above.

5.3 Road Location Output

Three forms of output are available: profile and plan-view plots and, ground information along the road location. The ground information (elevation, sideslopes, grade and distance from the start of the road) is computed for each coordinate pair describing the road location. The sideslope is determined by computing an elevation off to the side of the road and using it to calculate the slope to the road.

5.4 Earthwork Calculations and Gradeline Location

The procedures for allowing gradeline location and subsequent calculations of volumes are well documented in the literature (Burke, 1974). This feature has not been incorporated in the Road Location Module because the contour maps available in British Columbia are not sufficiently accurate to provide the necessary detail required in earthwork volume calculations.

5.5 Implementation

The data structure (200x3 array) allows for 200 road coordinates. Column one is the road number; columns two and three hold the $(x,y)$ coordinates.
The parameters controlling the automatic road projection were chosen through trial and error such that the road generated did not wander excessively or disregard important terrain features. Suitable parameters are:

\[ L = 0.125 \text{ inches} \times \text{map scale} \]
\[ |AB| = 0.5 \text{ inches} \times \text{map scale} \]
\[ \text{tolerance for convergence} = 0.01 \text{ inches}. \]

To conserve memory, the elevations were not stored for each pair of \((x,y)\) road coordinates. These could be generated whenever needed by using the \((x,y)\) coordinates to compute the elevation from the elevation data base.
6.0 Aesthetic Analysis

The Aesthetic Analysis Module consists of two main parts: the viewable area assessment and the three dimensional representations.

6.1 Viewable Area Assessment

The viewable area assessment function allows the user to determine the area of the data base that is viewable from a certain location. A person's line of sight is simulated in a full 360 degree spectrum from the viewpoint location \( (x^*, y^*, z^*) \) which is entered from the digitizer.

The map area is subdivided into a grid of \( n^2 \) cells that correspond to the elevation matrix, \( E \). The cells are checked by projecting sight lines from the viewpoint to each of the \( 4(n-1) \) border cells. Each cell, upon completion of the analysis, will be either viewable (true) or hidden (false).

6.1.1 Determining the Viewable Cells on a Sight Line

The digitizing coordinate system is scaled to correspond to the subscripts of the elevation matrix.

\[
x^* = nx*/u+1 \quad \text{and} \quad y^* = ny*/v+1
\]
The sight lines are intersected with each cell encountered between the viewpoint and the border cell. To simplify computations all cells are examined at the same relative position as the viewpoint has within its cell. The border subscripts \((x_b, y_b)\) are shifted to the same relative position as the viewpoint (Figure 39) by
\[
x_b = x^* - [x^*] x_b \quad \text{and} \quad y_b = y^* - [y^*] y_b
\]
The length of the sight line is found using
\[
L = \sqrt{(y_b - y^*)^2 + (x_b - x^*^2)}
\]
The planametric angle, \(\theta\), of the sight line is computed from
\[
\theta = \arctan\frac{(y^*-y_b)}{(x^*-x_b)}
\]
Commencing at the viewpoint with \((x_i, y_i)\) \((i=0)\) and \(x_i = x^*\) and \(y_i = y^*\) the next point on the line to be examined is:
\[
x_{i+1} = x_i + D\cos\theta \quad \text{and} \quad y_{i+1} = y_i + D\sin\theta
\]
The value \(D\) is the increment required to advance between two cells along the sight line. By setting \(D\) to be the smaller of \(L/(y^*-y_b)\) and \(L/(x^*-x_b)\) every cell in the grid will have been examined at least once upon completion of all the sight lines (Figure 45).

The cell containing \((x_{i+1}, y_{i+1})\) is viewable if the slope, \(s_{i+1}\), from \((x_{i+1}, y_{i+1})\) to the viewpoint is greater than the largest slope yet encountered for the sight line (Figure 40). The slope is calculated using \(s_{i+1} = (z_{i+1} - z^*)/D\). A tally for the cell is kept indicating the number of times the cell has been checked and the number of occasions it was viewable. Upon completion of all the sight lines, each cell will be defined as
Figure 39. Locating Grid Cells Along the Sight Line.
Figure 40. Determining the Viewable Area Along a Sight Line.
viewable (true) if at least 50 percent of the checks made were true. The overlay map showing the viewable areas is produced using the algorithm in Figure 33.

An allowance must be made for viewpoints located outside of the map area. The only adjustment required to the procedure already outlined is to skip over fictitious cells; i.e., those that lie outside the map area.

An allowance must also be made for cases when obstructions, not represented by the elevation data base, screen the viewpoint location (Figure 16). This is accomplished by giving the user the opportunity to override the initial viewing angle set by the computer for each sight line.

6.1.2 Implementation

Memory limitations dictated the use of a grid size of only 29x29 (741) cells. In addition, cells were deemed viewable if at least one sight line test showed the cell to be viewable. This eliminated the need to keep two tallies for each cell but introduced a bias in favour of viewable areas.
6.2 Three Dimensional Representations of the Elevation Data Base

There are two basic types of three dimensional projections: orthographic and perspective. Producing three dimensional (3D) representations of the data base is a three step process regardless of the projection:

(a) transforming the elevation data base to the correct orientation for plotting;
(b) plotting the 3D representation of the elevation data base and concurrently removing the hidden areas from the plot;
(c) plotting the locations of the planometric details, like roads and setting boundaries, on the 3D representation of the elevation data base.

Several parameters must be defined:

(a) area of the data base to be viewed;
(b) the cardinal viewing direction (N,S,E,W);
(c) the viewing angle within the cardinal direction;
(d) the viewing angle above the horizon;
(e) the viewpoint location (perspective);
(f) the viewing aperture (perspective).

The area to be viewed is defined by a rectangle that is parallel with the rectangle outlining the map area of the data base. This rectangle is located by digitizing its lower left $(x_L, y_L)$ and upper right $(x_R, y_R)$ corners. These coordinates must be converted to subscripts using Equation (1.1):
\[ j_L = [x_L \text{n}/\text{u}] + 1 \quad \text{and} \quad k_L = [y_L \text{n}/\text{v}] + 1 \]
\[ j_R = [x_R \text{n}/\text{u}] + 1 \quad \text{and} \quad k_R = [y_R \text{n}/\text{v}] + 1 \]

6.2.1 Transforming the Elevation Data Base

The elements within the elevation matrix must be rotated to coincide with the cardinal viewing direction. This is done because, although the transformation to be described can handle any rotation between 0-360 degrees, the hidden area algorithm must work from the foreground to the background of the elevation matrix. Consequently, if the foreground elevations in the matrix have their coordinates rotated to be in the background (rotation=180) the elevations would still be plotted first but should in fact be done last. Therefore the matrix of elevations must be "physically" rotated to the appropriate cardinal direction. The following rotations are required:

- if looking NORTH then no rotation is needed
- if looking SOUTH then rotate 180 degrees clockwise
- if looking EAST then rotate 270 degrees clockwise
- if looking WEST then rotate 90 degrees clockwise

To conserve memory the matrix is rotated within itself. The elevation matrix can be considered a series of \((n+1)/2\) concentric boxes. The boxes are individually rotated by systematically rotating each symmetric group of four elements; i.e., one element per side (Figure 41). The rotation algorithm is presented in Figure 42.
Figure 41. Rotation of One Box of the Elevation Matrix
Figure 42. Flowchart for the Rotation of the Elevation Matrix

* The symbol pair ( ) refers to the nearest lowest integer.
The subscripts defining the viewing area of the matrix must also be rotated:

looking west:
\[ x = j_R \]
\[ j_R = k_R \]
\[ k_R = n+2-j_L \]
\[ j_L = k_L \]
\[ k_L = n+2-x \]

looking south:
\[ y = n+2-k_R \]
\[ x = n+2-j_R \]
\[ k_R = n+2-k_L \]
\[ j_R = n+2-j_L \]
\[ j_L = x \]
\[ k_L = y \]

looking east:
\[ y = j_L \]
\[ j = n+2-k \]
\[ k = j_R \]
\[ j_R = n+2-k_L \]
\[ k_L = y \]

The actual viewing transformation can now be applied. The angle, \( \theta \), is the viewing rotation within the cardinal direction and must be applied first. The angle, \( \beta \), is the rotation above
the horizon and is applied second (Figure 43). The angle $\theta$ is a rotation about the $z$-axis (elevation) and the angle $\beta$ is a rotation about the $x$-axis. A rotation about the $y$-axis (depth) would tilt the database which was deemed undesirable and was therefore not included. The rotation for a pair of axes is simple and is found in most general mathematic books:

$$x' = x \cdot \cos \theta + y \cdot \sin \theta$$

$$y' = y \cdot \cos \theta - x \cdot \sin \theta$$

This rotation can be applied to the three dimensional case (Newman and Sproull, 1973):

Rotation about the $z$-axis ($\theta$) is

$$(x', y', z') = (x, y, z) \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation about the $x$-axis ($\beta$) is

$$(x', y', z') = (x, y, z) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix}$$

These two transformations can be combined:

$$(x', y', z') = (x, y, z) \begin{bmatrix} \cos \theta & -\sin \theta \cos \beta & \sin \theta \sin \beta \\ \sin \theta & \cos \theta \cos \beta & -\cos \theta \sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix}$$
Figure 43. Viewing Controls for Orthographic Projection.
Therefore,

\[ x' = x \cdot \cos \theta + y \cdot \sin \theta \]
\[ y' = -x \cdot \sin \theta \cos \beta + y \cdot \cos \theta \cos \beta + z \cdot \sin \beta \]
\[ z' = x \cdot \sin \theta \sin \beta - y \cdot \cos \theta \sin \beta + z \cdot \cos \beta \]

At this stage in the development the orthographic projection is complete. Perspective requires an additional two transformations which are described in Appendix D. It is assumed that the additional transformations creating the perspective projection have now been applied making the remainder of the discussion applicable to both projections.

The transformation (perspective or orthographic) must be applied to each \((x, y, z)\) point of the elevation data base. The \(y'\) is not required in the orthographic projection (but is in the perspective as is shown in Appendix D) as there is no adjustment made for depth. The 3D effect is created by projecting the transformed data onto a two-dimensional screen placed in front of the data base. This is accomplished by using the \((x', z')\) coordinates from each point.

The \(z'\) values are retained in the original elevation matrix. The \(x'\) values need not be stored as they can be generated as required. To simplify calculations the original \((x, y)\) planarmetric locations for the elevation grid, \(E\), are stored in two vectors \(X\) and \(Y\). The location for \(E(j,k)\) is therefore \((X(j), Y(k))\). The transformation to derive \(x'\) for point \((j, k)\) can be obtained from
\[ x' = x(j) \times y(k) \]

with \( X = X\cos\theta \) and \( Y = Y\sin\theta \)

To scale the 3D plot for output it is necessary to determine the maximum and minimum values for \( z' \). The corresponding range in \( x' \) is found from using the subscripts defining the viewing area:

\[ x'_{\text{min}} = x(j_L) + y(k_L) \quad \text{and} \quad x'_{\text{max}} = x(j_R) + y(k_R) \]

The \( x' \) values should be scaled such that their range is equal to the number of subscripts in the horizon vector, \( H \), to be used in the removal of hidden areas. This will allow "direct" addressing between the \( x' \) values and the horizon vector; i.e., the elevation at point \( x' \) in the horizon vector will be \( H(x') \).

Therefore:

\[ X_{\text{range}} = x'_{\text{max}} - x'_{\text{min}} \quad \text{and} \quad X = (X - X(j_L)) \cdot \frac{\text{(number of subscripts in horizon vector)}}{X_{\text{range}}} \]

\[ Y = (Y - Y(j_L)) \cdot \frac{\text{(number of subscripts in horizon vector)}}{X_{\text{range}}} \]

The plotting is now scaled:

- in the \( x \)-direction: 0 to the number of subscripts in horizon vector
- in the \( z \)-direction (where \( z \) is now the \( y \)-direction of the two dimensional screen): \( z'_{\text{max}} \) to \( z'_{\text{min}} \)
6.2.2 Plotting the Three Dimensional Representation

The most complicated aspect of plotting is the removal of those sections of the data base that are hidden from view. The 3D representation can be produced by plotting two sets of ground profiles; one set corresponding to the rows and the other to the columns of the data base. Figure 44 shows an example of using only row profiles. Figures 45a,b,c produced using the UBC computing center library program UBC:PERSP on the IBM 370/168 computer system, shows the inclusion of both sets of profiles for three different viewing directions. It is apparent that with the rotation being zero the column lines become almost vertical and contribute little to the perception of the area. Conversely when the rotation approaches 90 degrees the situation is reversed. Therefore the row profiles are used when the rotation is less than 45 degrees; column profiles are used when the rotation exceeds 45 degrees.

The plotting method is the same for the perspective and orthographic projections. The lines are plotted sequentially commencing at the foreground of the rotated data base. Each point along the profile line is checked against the viewing horizon. The point is plotted only if it is above the horizon; i.e., viewable.

Initially the horizon vector, \( H \), is set to zero. The \( k \) profile line (corresponding to one row of the elevation matrix) is plotted from right to left proceeding from grid point \( z_{j,k}^* \).
Figure 44. Orthographic Projection using One Set of Profile Lines and an Angle of Rotation of 0 degrees.
Figure 45a. Orthographic Projection using Two Sets of Profile Lines and an Angle of Rotation of 0 degrees.

*The plot is produced using a UBC Computing Centre library program called UBC:PERSP.
Figure 45b. Orthographic Projection using Two Sets of Profile Lines and an Angle of Rotation of 45 degrees.*

*The plot is produced using a UBC Computing Centre library program called UBC:PERSP.
Figure 45c. Orthographic Projection using Two Sets of Profile Lines and an Angle of Rotation of 80 degrees.*

*The plot is produced using a UBC Computing Centre library program called UBC:PERSP.
to grid point \( z_{j-1,k} \) where \( j=n+1, n, n-1, \ldots, 2 \). To create better resolution, points along the line projected between the two grid points are also checked. These points, \( P_i \), are determined from using the equation of the line between \( z_{j,k} \) and \( z_{j-1,k} \). Each point is checked against its corresponding position in the horizon array. The points are chosen at intervals of one unit in the x-direction thereby occurring at the same frequency as the elements in the horizon array. This is made possible by the scaling described earlier. The more points along the line the better the plot quality (Figure 46).

The points \( P_i(x_i, y_i, z_i) \) are generated by

\[
\begin{align*}
    z_{Pi} &= z_{Pi-1} + m_j, & \text{where } z_{P0} = z_{j,k}^i \\
    x_{Pi} &= x_{Pi-1} + 1, & \text{where } x_{P0} = x_{j,k}^i \\
    m_j &= (z_{j,k}^i - z_{j-1,k}^i) / (x_{j,k}^i - x_{j-1,k}^i) \quad \text{(slope of the line)} \\
    \text{where } x_{j,k}^i &= x(j) + y(k) \\
    \text{and } k &= \text{current row (profile)} \\
    \text{and } j &= \text{current grid point on row } k.
\end{align*}
\]

The elevation of each point, \( P_i \), is checked with its corresponding horizon elevation:

If \( z_{Pi} > H(x_{Pi}) \) then \( P_i \) is viewable and the plotting pen is moved to \((x_{Pi}, z_{Pi})\). Upon reaching \((x_{Pi}, z_{Pi})\) the pen is lowered. Consequently if the pen was previously down then \( P_i \) (the previous point on the profile) was viewable and a line is drawn. Otherwise, if the pen was previously up then \( P_i \) was hidden and no line is drawn. The horizon vector is updated with the higher elevation: \( H(x_{Pi}) = z_{Pi} \).
grid point $E(j,k)$ on profile $k$

intermediate elevation generated between $E(j+1,k)$ and $E(j,k)$.

gap created going from viewable to hidden; gap size dependent on number of intermediate elevations used.

Intermediate elevations are computed and checked against each element in the horizon array.

Figure 46. Removing Hidden Areas.
If $z_{p_i} < H(x_{p_i})$ then $P_i$ is hidden. The plotting pen is lifted but remains at its current position (the last viewable point).

6.2.3 Plotting the Locations of Planametric Details on the Three Dimensional Representation

The locations of the planametric details, like roads and settings, are plotted onto the three dimensional representation by applying the same transformation used to generate the 3D plot. The only difficulty is deciding whether a planametric detail is hidden from view. A possible solution would be at each $(x,y)$ point, describing the location, to search back on a line parallel to the viewing direction until the foreground is reached\textsuperscript{10}. If this sight line does not intersect with the ground then the $(x,y)$ point for the location of the planametric detail is viewable. This would be a very slow process. A good approximation is attainable by "flagging" the elevations in the data base array that are hidden, by, setting the signs of the elevations to negative as the 3D plot is being produced. Points describing the location could then be checked by merely testing

\textsuperscript{10} In the case of perspective projection, it is necessary to search back on a line that ends at the viewpoint as the sight lines are not parallel.
whether the elevation grid cell that it lies in, is viewable (plus) or not (minus). The error in this approach occurs because the entire cell area will be either hidden or viewable.

The plotting of the locations is quite simple. The \((x,y)\) coordinates for the location of the planometric details are digitized and then scaled to correspond to the subscripts of the elevation matrix:

\[
x_s = nx/u \quad \text{and} \quad y_s = ny/v
\]

The points \((x_s,y_s)\) must be rotated to the cardinal direction of view:

- if North no rotation is required
- if West - \(t = y_s\)
  - \(y_s = n - x_s\)
  - \(x_s = n\)
- if East - \(t = x_s\)
  - \(x_s = n - y_s\)
  - \(y_s = t\)
- if South - \(x_s = n - x_s\)
  - \(y_s = n - y_s\)

The coordinates \((j,k)\) of the grid cell containing \((x_s,y_s)\) are found using

\[
j = [x_s] \quad \text{and} \quad k = [y_s]
\]

The \(x'\) coordinate of the 3D representation can be found precisely by using the original transformation \(x' = x\cos\theta + y\sin\theta\). This is cumbersome however as the scaling used must also be applied. A simpler method that is as accurate uses
interpolation:

\[ x' = x(j) + (x(j+1) - x(j)) \times (x_s - j) + y(k) + (y(k+1) - y(k)) \times (y_s - k) \]

Similarly the elevation, \( z' \), can be found using three-step interpolation (Section 1.2.1):

\[
\begin{align*}
    z_1 &= z_{j,k}' + (z_{j,k}' - z_{j+1,k}') (y_s - k) \\
    z_2 &= z_{j+1,k}' + (z_{j+1,k}' - z_{j+1,k+1}') (y_s - k) \\
    z' &= z_1 + (z_2 - z_1) (x_s - j)
\end{align*}
\]

- where all \( z' \) values are taken as absolute values to avoid the +/- flags.

The plotting pen can now be moved to \((x', z')\). The decision as to whether the point is hidden or not depends on the sign of \( z' \) in the vicinity of \((x_s, y_s)\). A simple but effective criterion is to test the sign of \( z'(x_s, y_s) \) where the values \((x_s, y_s)\) are rounded to the nearest integer value and not truncated as done above, i.e., \( j = [x_s + 0.5] \) and \( k = [y_s + 0.5] \).

A planometric map showing the hidden areas of the 3D plot can be generated by using the plus/minus designation of the elevation values as true and false indicators. The map can then be produced using the plotting algorithm in Figure 33.
6.2.4 Implementation

The slow execution speed of the HP9830A computer made the use of the perspective projection infeasible. The orthographic projection required approximately one hour depending on the viewing controls. The horizon vector was limited to only 200 elements.
7.0 Harvesting Costs and Wood Volume Production

The Harvesting Costs and Wood Volume Production Module is potentially the most important of all analyses in assessing whether a harvesting plan derived from the other simulator modules is feasible. The elevation data base must be augmented by wood volume and soil data. The wood volume data are readily available, usually in the form of forest inventory maps. However, the soil/landform data tend to be lacking, limiting the effectiveness of the productivity estimates.

The module consists of two parts: creation of the wood volume and soils data base; and, the analysis of the various setting alternatives (volumes and costs).

9.1 Creation of the Wood Volume and Soils Data Base

There are two possible data base structures: retaining the shape of the covertype boundaries; or, using a grid system with each cell being designated as a certain covertype and/or soil type. The former is difficult to implement as attempting to overlay one polygon (setting boundary) onto several polygons (covertypes) is a problem that far exceeds the capability of the desktop computer. The grid system, however, is simple, efficient and directly compatible to the elevation data base.
The method of entry of the inventory (wood volumes and soils data) into the data base can also be accomplished in two ways. The easiest from a user's standpoint is to trace the polygon boundary of each covertype and then have the computer determine which cells are located within the polygon. The range and domain of the boundary points are used to indicate which of the \( n^2 \) cells in the grid can feasibly be contained by the polygon. The centroid of each of these cells is used in a point-in-polygon algorithm (Appendix A) to determine if the cell is actually within the polygon boundary.

A simpler method to implement is having the user digitize one point in each cell contained by the covertype polygon (Figure 19). In this way the user performs the point-in-polygon task. The cell subscripts for an \((x,y)\) point from the digitizer are determined using Equation (1.1).

\[
J = \left\lfloor \frac{x}{u} \right\rfloor + 1 \quad \text{and} \quad k = \left\lfloor \frac{y}{v} \right\rfloor + 1
\]

An edit facility to correct errors (often occurring at the boundary regions of the polygons) must be provided. A simple grid map (Figure 20) will aid in locating these errors. Cells can then be properly allocated using the entry method described above.
9.2 Setting Analysis

If the data structure is a grid then the logging setting area must be similarly defined. The cells that are contained by the setting boundary are entered as before; i.e., either by tracing the boundary and then using a point-in-polygon routine or manually digitizing points in each cell. In addition to the setting boundary, the landing location \((x_L, y_L)\) and the type of yarding system to be used are required.

The productivity and wood volume estimates are made by deterministically simulating the yarding of the wood that lies on each cell. Productivity is calculated by using a regression equation or a loss-factor approach.

The loss-factor approach commences with a normal production rate for the cell. Then, certain deductions are made from this value according to various yarding conditions. The initial production rate and the deductions are machine dependent. An example of loss-factors (derived from local experience or production studies) for a 90-foot spar highlead system are:

- yarding distance; lose five percent production for each 100 feet beyond 600 feet;
- ground slope: lose five percent production for each 20 percent of slope beyond 60 percent;
- for downhill yarding lose five percent production.

In the above example the losses are additive which in some circumstances will not be correct. The yarding time for each
cell is computed from

\[
time = \frac{volume}{\text{adjusted production rate}}
\]

There are many important variables associated with estimating the productivity of any yarding system:

- type of yarding system
  - machine characteristics
  - mechanical availability and utilization
- yarding distance and direction
- ground slope
- crew experience
- piece volume
- turn volume
- species
- weather
- type of terrain
- quality of landing
- amount of deflection.

All of these components could be incorporated into the module providing their effect on yarding is quantitatively known. Some of these, however, such as weather and crew experience, are probably too specific for the development planning level. Species and piece volume size can be estimated from the inventory data while the soil/landform data can provide information on the terrain type. The crew experience, weather and landing quality are all highly variable and if used would have to be provided by the user for each setting examined.
The yarding distance is computed by:

\[ d = \sqrt{(x_L - x)^2 + (y_L - y)^2} \]

with \((x, y)\) the coordinates of the cell being "yarded".

The ground slope in the cell will be equal to the slope computed from the Topographic Features Module. If the slope data cannot be kept in memory then the slope can be estimated by computing the elevation for a point near the centroid that lies on a line that extends from the centroid to the landing. This arbitrary point \((x', y')\) is found from:

\[ x' = x + d \cdot \cos \theta \quad \text{and} \quad y' = y + d \cdot \sin \theta \]

with \(d = \) the distance to \((x, y)\) from \((x', y')\)

\[ \theta = \arctangent \left( \frac{y_L - y}{x_L - x} \right) \]

A summary of the operating statistics for the setting can be easily produced (Figure 21).

9.3 Road Costing

Approximate road costing can be determined by tracing the route location with the digitizer and computing the length:

\[ L_{i+1} = L_i + \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \]

with \((x, y)\) being the road location coordinates.
soil/landform data are paramount for properly estimating road costs. The cells containing the soils data that are intersected by the road can be determined from Equation (1.1). Using the soil data for each cell a proper cost could then be made either through a regression or a loss-factor approach.

9.3 Implementation

The slow execution speed of the computer dictated the use of the manual entry method for constructing the wood volume database. The point-in-polygon approach was tried but proved to be too slow. A limit of 30 covertypes, each having up to six species, was imposed. The stand formula consisted of the combined species volume per acre along with the percent composition of each species. A loss-factor approach (derived from local experience) was employed with data available for three machines: a Madill 052 tension skidder, a 90-foot highlead spar and a Madill grapple yarder.
Discussion

1.0 Economic Justification of the Terrain Simulator

The general deficiencies of the modules in the terrain simulator, as currently implemented, have been previously outlined in the thesis. For the most part, the drawbacks were due to three main factors: the inadequacies of the Hewlett-Packard 9830A computer; the low precision of the elevation data; and, the lack of soils, hydrological, land-use, productivity and costing data. These limitations generally dictated the use of the least precise approaches.

Given these drawbacks the important question to be answered is whether or not the terrain simulator in its present form constitutes a useful tool for short-term forest planning. Does it in fact justify the expenditure of some $40,000 to $50,000 on a computer like the HP9830A? This question was partially answered during the summer of 1976 when the model was tested in an operational environment at MacMillan Bloedel's Franklin River Logging Division.

For the analysis, three study areas were selected, each being at a different stage of development. The areas were approximately 1000 acres in size with the only available data
being wood inventory and elevation contour maps. The evaluation of the terrain simulator was primarily based on its applicability for the planning of harvesting operations.

The first area to be examined had recently been harvested, primarily with a long-line, Madill 052 tension skidder cable system with a maximum yarding distance of 2000 feet. Originally the area was to be logged using only the conventional highlead system which has a maximum yarding distance of less than 800 feet. This approach required the building of approximately one mile of road. Upon completion of this access road, it was discovered that a tension skidder, strategically located at several points along existing roads could reach the majority of the wood in the area to be harvested, making the construction of the just-completed road unnecessary.

This same after-the-fact conclusion was reached by using the terrain simulator. With the model it was possible to predict, within an accuracy of 50 feet, the locations of the setting boundaries used in the tension skidder operations. This unneeded road, cost the company an amount equivalent to the price of the computing system capable of operating the terrain simulator. There is, of course, no guarantee that with the simulator this mistake would have been avoided as it is still the responsibility of the planner to devise the alternatives.

The other two study areas were in various preliminary stages of planning development. Although extensive testing was
not done on these two areas, the simulator did prove useful in helping to decide which yarding systems would be needed along with their required deployments. It also helped to assess what field work was necessary to establish the operational harvesting plan.

The general conclusion derived from the tests was that the simulator constitutes a powerful tool for preliminary harvest planning. Its main advantage is that it gives the planner the opportunity to examine a large number of alternatives not normally possible using the existing manual techniques. With the often complex cost tradeoffs between yarding systems and the high cost of road construction, it is imperative that all possible options are investigated. Realistically, this can only be achieved with the aid of a computer model like the one developed in this thesis.

For the other planning considerations, such as aesthetics, the improvements or savings gained from using the simulator are difficult to quantify. They are nonetheless important considerations for justifying the acquisition of a desktop computer. The desktop computer has other valuable uses in a logging division. These include the compilation of field notes for roads (Burke, 1974) and yarding profiles (Carson, 1975) and various types of accounting procedures.
2.0 Additions to the Current Model

The modules presented in this thesis, collectively, do not represent all the components needed in forest development planning. Two important components that could be incorporated, with little difficulty, would be a hydrology module and a graphical information retrieval module. The following descriptions of these two modules are meant only to outline what might be done.

2.1 A Hydrology Module

A hydrology module could be designed to analyze water flow through the study area. The basic input could consist of the locations and classifications of the streams and the water flow properties of the soil types in the study area. The stream locations could be entered in an analogous way to that used for the road network; i.e., by digitizing the locations. The water flow properties would best be incorporated as part of the classification system used to describe the soil types. Water flow could then be simulated by systematically analyzing each cell in the data base - assuming that a grid type data structure is being used.

One approach to simulating water flow could commence with the cell of the highest elevation in the study area. The water
it holds (e.g., from spring-runoff or from a major storm which could both be inputs) could be "passed" onto its adjacent neighbours according to its aspect and ground slope as calculated from the Topographic Features module. The other outputs, like transpiration and evaporation, should also be considered. The amount of water transferred would depend on the cell's ground slope and soil type. This process could then be repeated for each cell in the study area by examining them in descending order of elevation. Water accumulated in a cell intersected by a stream could be "moved" along the stream.

Using this type of water flow simulation, various types of analyses could be performed:
- estimating possible sedimentation problems and the resultant water quality
- estimating required culvert and bridge sizes
- predicting possible erosion control problems
- estimating total water quantity and supply.

2.2 A Graphical Information Retrieval Module

A graphical information retrieval module could be designed as an extension of the Topographic Features module already developed. In addition to slope, aspect and elevation the map overlays could include soils, land-use (like the Canada Land Inventory) and wood inventory data. This function would have widespread use in general resource allocation. For example, if
a general resource development plan was being devised for the study area, it would be feasible with this type of module to generate map overlays depicting the various conflict areas between resources and/or uses. As in the combination mapping in the Topographic Features module, an acreage summary would also be produced. For instance, it may be desirable to know which regions in the study area had both good harvesting potential and sensitive wildlife habitat. The module could search for all the cells in the data base that contained these two resource uses and then show the locations of these areas on a map overlay along with the associated acreage.

The effect on the resource base of land-removals could also be determined. By examining all the data base cells in the removal area it would be possible to determine how much of a certain resource, such as wood volume, had been removed. The cells in the region could be identified to the computer by digitizing the region boundary and then using a point-in-polygon routine (Appendix A).

To implement such a module it would be necessary to build a large data base which will require considerable amounts of computer memory and/or fast-access storage such as that obtained with a disc drive. Each multi-dimensional cell in the data base will have to hold information on each of the resources and land-uses being considered in the study area. The data base could be created in a manner similar to that used to enter the wood inventory into the Harvesting Costs and Wood Volume Production
2.3 Summary

Two modules have been suggested as possible additions to the terrain simulator model. These are a hydrology module for estimating water flow patterns through the study area; and, a graphical information retrieval module to locate and summarize the various resource and land-use tradeoffs. Other possibilities for extending the applications of the terrain simulator model undoubtedly exist.
Conclusion

The management of forest resources has become an extremely complex problem. The existing procedures used in the planning process have not, in many cases, been able to provide the necessary solutions. The computer model presented in this thesis was developed to help meet this difficult challenge. Part of the task in creating the model was the integration of several developments, such as VIEWIT (Travis et al., 1975), into one planning package. It was designed to deal with the following aspects of short-term resource planning:

- collection of the required terrain elevations and forest inventory data;
- determination and production of map overlays of the topographic features: slope, aspect and elevation;
- design of logging settings for cable systems and placement of yarding roads;
- location of forest-access roads;
- delineation of viewable areas and production of three dimensional representations of the terrain;
- estimation of harvesting costs and wood volume production.
The format used for the elevation data base was a regular grid of elevations. The most accurate format, however, would retain the locations of the original elevation contours. In this method there would be virtually no information loss. Unfortunately, the algorithm required to generate an elevation for a map location far exceeds the capability of the HP9830A computer.

For the regular grid approach, each cell surface was described mathematically by either an interpolation routine that used the cell corner elevations, or, by a surface equation method which was typically some form of geometric plane. The interpolation approach requires more computational work to generate an elevation but storage requirements are less as only one elevation per cell need be retained. The equivalent surface equation method, (computed only once at the input stage using the same grid of elevations as the interpolation method) is simple to evaluate, but each of the parameters must be retained thereby increasing required storage capacity.

Two methods were proposed for forming the elevation data base. The simplest method used equally spaced transect lines to cover the map of the study area. The contour intersections on each line were digitized and then used to compute the elevations at equally spaced points on the transect line. Therefore, the data base was always in the form of a regular grid. The problem
with the method was that each line was compiled independently which could introduce large errors when the contours run parallel to the transect lines. To avoid this problem, a second set of transect lines could be run perpendicular to the first group. The grid of elevations could then be computed by merging the two sets of lines. This approach would increase digitizing time as well as the amount of storage required and was therefore not considered.

The second approach developed for creating the elevation data base was the contour method. Each contour line was digitized and stored in memory. The elevation was computed for each grid point by using the elevations from the closest digitized point in each of the four quadrants surrounding the grid point. To lessen the search time required to locate the closest points, the digitized points were initially classified into smaller map units. This made it necessary to examine only those points that were contained in the map unit that held the grid point. The accuracy of the contour method was superior to that of the transect method; but, due to memory restrictions, the HP9830A system could only handle small maps or those of limited complexity when using the contour method.

The topographic features (slope, aspect and elevation) were computed for each cell of the elevation data base from a geometric plane fitted to the ground surface using a least-squares procedure. The parameters of the resultant plane equation were then used to determine the slope, aspect and
elevation. Map overlays showing these three topographic features, either singly or in varying combinations, were produced using the area of one data base cell as the smallest plotting unit.

The Yarding Location and Setting Design module concentrated on developing the cable mechanics theory needed to predict the loadpaths of skyline cable systems. The most theoretically correct approach (the catenary model) was not used because similar accuracy could be achieved using the simpler, parabolic model. (In the parabolic formulation, cable weight was assumed to be distributed on the chords of the cables, whereas in the catenary model, cable weight was distributed on the cable itself).

Newton's method for solving non-linear equations was used to solve the parabolic model. The iteration equation was easy to formulate and converged quickly and dependably to the solution for the load position. Even with fast convergence, however, the execution time using the HP computer was excessive. The percent deflection rule was therefore needed to predict the loadpath. It expressed the minimum attainable midspan load deflection as a percentage of the chord distance between the tops of the landing and back spars. To determine the loadpath for the rest of the span, an interpolating polynomial was fitted to the three nodes; i.e., the midspan load position and the tops of the landing and back spars.
Results from the percent deflection approach compared favourably to the catenary model, provided the midspan deflection was known. These deflections are readily available in tables that are produced using the catenary model. The inaccuracy of the available contour maps limits the module to development planning; it cannot be used for the actual operational planning phase.

In the Road Location module, two methods for the projection of roads are provided: "manual" and "automatic". In the "manual" method, points along the road location were digitized by the user. The only computation required was to test whether the grade between successive trial points was between the specified allowable adverse and favourable grade limits. The "automatic" road projection routine used an heuristic algorithm to locate in the horizontal plane, the average grade between the digitized start and finish points of the route. A secant search, employed at successive intervals along the route, located the average grade in each of these intervals. The determination of earthwork volumes for road construction was deemed inappropriate due to the inaccuracy of the maps.

To determine the viewable area, a viewpoint was specified from which sight lines were projected to each border cell in the elevation grid. The cells that intersected the sight line were judged viewable if the slope for the line of sight between it and the viewpoint was greater than any similarly calculated slope on the sight line. The overlay was produced using the
same plotting algorithm put forth in the Topographic Features module.

There were two methods developed for producing three dimensional representations of the study area: orthographic and perspective projection. The perspective projection was not implemented due to the added transformations required. The three dimensional effect was produced by plotting lines that were perpendicular to the line of sight onto a two dimensional surface. The hidden areas were removed by eliminating points on the profile lines that were at a lower elevation than the corresponding points in the previous profile. The locations of planometric details (roads and setting boundaries) were plotted on the three dimensional representations by applying the same transformation used to generate the representation. Planometric details that were in a position that was hidden from view were not plotted.

In the Harvesting Costs and Wood Volume Production module, two methods were developed to enter the wood inventory and soils data into the grid-type data base. These were: (1) digitizing a point in each cell contained by the covertype; or, (2) by digitizing the covertype boundary and then using a point-in-polygon routine to determine which cell were contained in the covertype. An algorithm deterministically simulated yarding the wood from each grid cell in the logging setting. Productivity estimates could use either a loss-factor or regression equation approach; only the loss-factor approach was used. Road
Construction costs were estimated by digitizing the road location, computing its length and then determining the cost using either a loss-factor or regression equation approach. In the implementation of the model, however, a constant cost per mile was assumed due to lack of data.

The principal limitations of the current version of the simulator, are due to three main reasons:

- the inadequacies of the Hewlett-Packard 9830A computer;
- the lack of accurate terrain data;
- the lack of data on soils, hydrology and land-use.

The limited computer technology, i.e., slow execution speed, small memory and lack of a fast-access storage device, was the most critical of these drawbacks. This meant that in the majority of the cases the least precise algorithm had to be used.

Future development of the terrain simulator should concentrate on several main areas:

1. As computer technology improves along with the availability and quality of data, the various restrictions due to these drawbacks should be removed from the model. The HP9830A computer is presently being replaced by a new generation of desktop computers which appear to provide the necessary execution speed and fast-access storage capability needed to remove many of the current limitations of the model. The use of orthophotographs for obtaining
the elevation data base should also be thoroughly investigated.

(2) New planning components should be developed such as a hydrology module to analyze water flow through the study area and a graphical information retrieval module that would be able to produce map overlays for various resource and land-use combinations.

Although full implementation has not been possible, this study has demonstrated that the terrain simulator, even in its present form, is a useful planning tool that can aid in the management of forest resources.
Literature Cited


APPENDIX A

Point-in-Polygon Algorithm

There are several possible approaches for deciding if an (x,y) point is subtended by a polygon. All the methods require an exhaustive search of the polygon boundary except in the case of a strictly convex shape (Nordbeck and Bengt, 1967). Polygons cannot, however, be restricted to only convex shapes.

A simple and effective method is to project a ray from the (x,y) point being examined. If it crosses the polygon boundary an odd number of times it must be subtended by the polygon (Figure 47).

The polygon boundary is represented by a series of p line segments that are defined by p+1 (x,y) points. The point to be checked has the coordinates (x*,y*). The computations are simplified by using a ray that has the defining equation x=x*. The kth line segment, defined by the points (x_k,y_k) and (x_{k+1},y_{k+1}), intersect the ray CA (Figure 48) at (x_i,y_i) with x_i=x*.

To find the point of intersection (x_i,y_i) it is necessary to compute the parameters b and m for the equation of the kth line segment (y=mx+b). The parameter b (the y-intercept) can be solved by substituting (x_k,y_k) into the equation of the line:

\[ y_k = mx_k + b \]

Therefore

\[ b = y_k - mx_k \]

The slope of the line, m_k, is found from

\[ m_k = (y_{k+1} - y_k)/(x_{k+1} - x_k) \]
Figure 47. Determining if a Point is Within a Polygon
Figure 48. Determining if an Intersection Takes Place.
The equation for the $k^{th}$ line segment is therefore
\[ y = m_k(x - x_k) + y_k \] \hspace{1cm} (1)

The y-coordinate of the point of intersection $(x_i, y_i)$ is computed by evaluating Equation (1) with $x = x^*$. Therefore
\[ y_i = m_k(x^* - x_k) + y_k \]

If $y_i < y^*$ then the intersection is fictitious as it takes below C on the ray $\vec{CA}$. If $y_i > y^*$ then the intersection occurs on $\vec{CA}$ but it is necessary to test if $(x_i, y_i)$ is on the line segment. This will be the case if
\[ |x_{k+1} - x_k| = |x^* - x_k| + |x^* - x_{k+1}| \]

If the line segment does not cross $\vec{CA}$ then $(x_i, y_i)$ is not on the line segment and
\[ |x_{k+1} - x_k| < |x^* - x_k| + |x^* - x_{k+1}| \]

Each of the $p$ line segments must be examined in this fashion and a record kept of the number of intersections. On completion, if the number of intersections ($n$) is odd then $(x^*, y^*)$ is inside the polygon. A suitable test is
\[ \text{Odd if } n - 2(n/2) = 1 \]

where $(n)$ is the nearest lower integer.
APPENDIX B

Menu Operation

The two menus used for entering either the transect or contour lines represent a graphical approach for allowing the user to control the program which is normally done from keyboard. This allows the operator to concentrate on the digitizing rather than continuously moving between the digitizer and the keyboard.

For the purposes of discussion, the transect menu will be presented. The menu is divided into regions which are all referenced to the top left corner of the menu. Each region has associated with it a certain program control.

As seen from Figure 49, a potential problem arises when the menu is not properly aligned. To alleviate this, the top \((x_t, y_t)\) and the bottom \((x_b, y_b)\) left corners of the menu are digitized. The menu coordinates can be rotated through an angle of \(\theta\) so that the line between \((x_t, y_t)\) and \((x_b, y_b)\) is parallel to the y-axis. The angle \(\theta\) is defined by

\[
\theta = \arctan \left( \frac{x_b - x_t}{y_t - y_b} \right)
\]

Rotating the menu coordinate system through the angle of \(\theta\) is given by

\[
x' = x_t \cos \theta + y_t \sin \theta \quad \text{..........................(1)}
\]
\[
y' = y_t \cos \theta - x_t \sin \theta \quad \text{..........................(2)}
\]

Using equations (1) and (2), each point entered in the menu must be rotated to the 'new' menu coordinates. This must be done to the reference point of the menu:

\[
x_t' = x_t \cos \theta + y_t \sin \theta
\]
\[
y_t' = y_t \cos \theta - x_t \sin \theta
\]

The menu in Figure 49 is based on quarter-inch squares; therefore, each digitized \((x,y)\) point must be converted to the subscripts of the squares. First it is rotated to \((x',y')\) using Equations (1) and (2). The subscripts are found from:
Figure 49. The Menu Coordinate System.
are found from:

\[
x' = \left\lfloor 4(x' - x_c) \right\rfloor + 1
\]

\[
y' = \left\lfloor 4(y' - y) \right\rfloor + 1
\]

where \( \lfloor \cdot \rfloor \) is the lowest nearest integer.

The command selected by the user is determined by examining \((x', y')\):

- If \( x' = 6 \) then the command chosen is 'COMPLETED'
- If \( x' = 7 \) then the command chosen is 'END OF ENTRY'
- If \( x' = 8 \) then the command chosen is 'MOVE MENU'
- If \( x' = 9 \) then the command chosen is 'CHANGE CONTOUR INTERVAL'
- If \( x' < 6 \) then a number is being entered. The base ten exponent will be equal to \( 5 - x' \). The number chosen will be \( y'(5 - x') \).

A full five-digit number can be found by accumulating each entry until a 'COMPLETED' command is chosen.
The cable mechanics theory developed in this appendix is partially derived from the participation of the author in a graduate-level university course entitled Cableways with fellow graduate students, Mr. D.G. Guimier and Mr. D.I. Anderson and chaired by Professor G.G. Young.
There are numerous cable logging systems in use today and it is possible to model any of these. Generally, each will require a different formulation principally depending on the number of lines and their operational configuration. Some of the most common cable systems are shown in Figures 50, 51 and 52.

For any cable system the model, at a distance $x$ from the landing spar, locates the load position at a vertical displacement of $y$ such that all forces in the yarding system will be in equilibrium. Only those forces encountered in the static condition are considered. The dynamic forces are extremely difficult to compute and are generally not predictable. This problem is clearly illustrated when attempting to model the highlead cable method of logging. The precise type of models to be discussed are not suitable as loads are usually dragged over the terrain to the landing spar. This practice results in large, unpredictable forces.

Various models can be devised for any one system. The differences are due to the assumptions used for the cable weight distribution. These assumptions yield different equations for describing the shape of a freely hanging cable. Consequently the balancing of the forces in the system, although similar in nature, will be mathematically different. There are three possible ways to distribute the cable weight:

1) The cable weight is uniformly distributed over the horizontal span of the system.
Figure 50. Running Skyline System. (from Studier and Binkley, 1974)
Figure 51. Gravity Skyline System. (from Studier and Binkley, 1974)
Figure 52. Highlead System. (from Studier and Binkley, 1974)
2) The cable weight is uniformly distributed over the chord of the system.

3) The cable weight is uniformly distributed over the length of the cable system itself.

These different approaches are shown in Figure 53.

The greatest accuracy is achieved using assumption 3. This model is called a catenary. Although the catenary is theoretically most correct, assumptions 1 and 2 provide an easier formulation and little sacrifice in accuracy (Guimier, 1977). The chord assumption (2) is the more accurate of the two and consequently development in this thesis is based on it. Providing the cables are tight, which they normally are in logging systems, the results are extremely good.

Derivation of the Equation for the Shape of a Freely Hanging Cable

Geometry

The basic geometry is illustrated in Figure 54 with the following definitions in use:

A : left hand support
B : right hand support
L : span; horizontal distances between supports
E : difference in elevation between supports
\( \overline{AB} \) : chord between supports
\( \theta \) : angle between the chord and the horizontal with \( \theta = \arctan(E/L) \)
P : any point on the cable
Case 1. The cable weight is uniformly distributed over the horizontal span of the system.

Case 2. The cable weight is uniformly distributed over the chord of the system.

Case 3. The cable weight is uniformly distributed over the length of the cable system itself.

Figure 53. The Three Different Loading Assumptions for the Cable Weight.
Figure 54. Geometry of a Freely Hanging Cable.
0 : point of maximum sag of the cable; acts as the origin for the 
(x',y') coordinate system

**Forces**

The tension at any point on the cable acts along the tangent to 
the cable at that point.

- \( T_P \) : tension in the cable at P
- \( H_P \) : horizontal component of the tension at P
- \( V_P \) : vertical component of the tension at P
- \( T_A \) : tension in the cable at support A
- \( T_B \) : tension in the cable at support B
- \( \omega' \) : weight of the cable per unit length

Employing the assumption that the cable weight acts on the chord:

\[ \omega = \omega'/\cos \theta \]

As shown in Figure 55 the forces acting on the system (the section of 
the cable between 0 and P) are:

- \( \tilde{H} \) horizontal tension at 0
- \( \tilde{H}_P + \tilde{V}_P = \tilde{T}_P \) tension at P
- \( \omega = x' \omega \) weight of cable between C and C

The equations of equilibrium are:

- Sum of the vertical forces: \( \tilde{V}_P - \omega x' = 0 \)
- Sum of the Horizontal forces: \( -\tilde{H} + \tilde{H}_P = 0 \)
- Sum of the moments about P=0: \( \tilde{H} y' - \omega x'(x'/2) = 0 \)

Rearranging the moment equation yields:

\[ y' = \frac{\omega x'^2}{2H} \]

Equation (1) represents the equation of the cable shape within 
the (x',y') coordinate system whose origin is at 0 (the point of maximum 
sag). By inspection the equation describes a parabola, hence the name
Figure 55. Forces Acting on the Cable Segment OP.
'Parabolic Approximation'. Equation (1) is not convenient in its present form since the origin of its coordinate system moves depending on the geometry and the tensions. The next step is to translate Equation (1) to a fixed point which is normally at one of the supports; support A is used.

Equation of the Cable Shape in the (x,y) Coordinate System, with Origin at Support A

The translation is defined by:

\[ x' = x - a \]
\[ y' = y - b; \quad \text{with } a, b \text{ shown in Figure 56} \]

Substituting for \( x' \) and \( y' \) in Equation (1) yields:

\[ y + b = \frac{\omega (x-a)^2}{2H} \]

The values \( a \) and \( b \) are found by employing the limit conditions (\( x=0, y=0 \)) and (\( x=L, y=E \)). The first gives:

\[ 0 + b = \frac{\omega (0-a)^2}{2H} \]

Therefore: \( b = \frac{\omega a^2}{2H} \)

Similarly the second yields: \( E + b = \frac{\omega (L-a)^2}{2H} \)

Simplifying: \( E = \frac{\omega (L^2 - 2La + a^2)}{2H} - b \)

Substituting for \( b \) gives: \( E = \frac{\omega (L^2 - 2La)}{2H} \)

Rearranging: \( a = \left( \frac{L}{2} - \frac{HE}{\omega L} \right) \)

Therefore: \( b = \frac{\omega \left( \frac{L}{2} - \frac{HE}{\omega L} \right)^2}{2H} \)
Figure 56. Translating the Coordinate System from $O(x', y')$ to $A(x, y)$. 
Substituting the values of a and b into Equation (2) gives:

\[ y + \frac{\omega}{2H} \left( \frac{L - \frac{HE}{2}}{\omega L} \right)^2 = \frac{\omega}{2H} \left( x - \frac{L - \frac{HE}{2}}{\omega L} \right)^2 \]

Simplifying: \( y = \frac{\omega x^2}{2H} - \frac{\omega L}{H} \left( \frac{L}{2} - \frac{HE}{\omega L} \right) x \)

And finally:

\[ y = \frac{\omega x^2}{2H} + \left( \frac{E}{L} - \frac{\omega L}{2H} \right) x \] ........................ (3)

This is the parabolic equation describing the shape of a freely hanging cable with the coordinate system origin at the left hand support (A).

Tangents to the Cable

The tension acts along the tangent of the cable. The tangent of the cable at any point P is given by the first derivative of Equation (3) with respect to \( x \).

\[ \frac{dy}{dx} = \frac{\omega x}{H} + \left( \frac{E}{L} - \frac{\omega L}{2H} \right) = \text{tangent} \alpha = \text{slope of the cable at } x \]

Therefore:

\[ \tan \alpha = \frac{\omega x}{H} + \left( \frac{E}{L} - \frac{\omega L}{2H} \right) \] ........................ (4)

Derivation of the Model for a Five-Line Cable System

The five-line cable system is common in logging and is often referred to as the running skyline (Figure 50). The five cable segments (Figure 57) are:
Figure 57. Geometry of a Five Line Cable System.

Definition of cable segments:
1,2,2': haulback
3: mainline
3': slackpuller
1: haulback segment between support A and carriage
2,2': haulback segment between carriage and support B
3: mainline segment
3': slackpuller segment.

Additional definitions are:

- $\omega_1$: weight per unit length of haulback
- $\omega_2$: weight per unit length of mainline
- $\omega_3$: weight per unit length of slackpuller
- $R$: weight of load (carriage and logs)
- $C$: carriage

$\theta_1, \theta_2$: angles of subchords with:

- $\theta_1 = \arctan \left( \frac{y}{x} \right)$
- $\theta_2 = \arctan \left( \frac{(y+E)}{(L-x)} \right)$

$\alpha_{C1}, \alpha_{C2}, \alpha_{C3}, \alpha_{C2'}, \alpha_{C3'}$: angles of cables at the carriage with the horizontal.

The general load position equation can be derived by using the equations that describe the equilibrium of the carriage.

Sum of horizontal forces: $H_1 + H_3 + H_3' = H_2 + H_2'$

Sum of vertical forces: $V_1 + V_2 + V_2' + V_3' + V_3 = R$

The vertical components, $V_i$, can be replaced by: $V_i = H_i \tan \alpha_{C_i}$.

Therefore:

$$H_1 \tan \alpha_{C1} + H_2 \tan \alpha_{C2} + H_2' \tan \alpha_{C2'} + H_3 \tan \alpha_{C3} + H_3' \tan \alpha_{C3'} = R \quad \text{......................... (5)}$$

$\tan \alpha_{C1}$ can be replaced by Equation (4) evaluated at $x=0$:

$$\tan \alpha_{C1} = \left( \frac{y}{x} - \frac{\omega_1 x}{2R_2} \right); \quad \text{L replaced by x as the span of segment 1 is x units.}$$
\[ \tan \alpha_{C2} = \left( \frac{E+y}{L-x} - \frac{\omega_2(L-x)}{2H_2} \right); \text{ span is } L-x; \text{ difference in elevation is } E+y \]
\[ \tan \alpha_{C2'} = \left( \frac{E+y}{L-x} - \frac{\omega_2'(L-x)}{2H_2'} \right) \]
\[ \tan \alpha_{C3} = \left( \frac{y}{x} - \frac{\omega_3 x}{2H_3} \right) \]
\[ \tan \alpha_{C3'} = \left( \frac{y}{x} - \frac{\omega_3' x}{2H_3'} \right) \]

With \( \omega_1 = \omega_1'/\cos \theta_1 \), \( \omega_3 = \omega_3'/\cos \theta_1 \), \( \omega_3' = \omega_3'/\cos \theta_2 \), \( \omega_2 = \omega_1'/\cos \theta_2 \)

Substituting these values of the tangents into Equation (5) yields:
\[ H_1 \left( \frac{y}{x} - \frac{\omega_1 x}{2H_1} \right) + H_2 \left( \frac{y+E}{L-x} - \frac{\omega_2(L-x)}{2H_2} \right) + H_2' \left( \frac{y+E}{L-x} - \frac{\omega_2'(L-x)}{2H_2'} \right) + H_3 \left( \frac{y}{x} - \frac{\omega_3 x}{2H_3} \right) + H_3' \left( \frac{y}{x} - \frac{\omega_3' x}{2H_3'} \right) = R \]

At this point several simplifications can be made. Assuming the haulback segments 2 and 2' are parallel and there is no friction at the tailhold sheave (support B) then:

\[ V_{C2} = V_{C2'} \]
\[ \omega_2 = \omega_2' \]
\[ H_2 = H_2' \]
\[ T_{C2} = T_{C2'} \]

Therefore the vertical force balance equation becomes:
\[ \frac{y(H_1+H_3+H_3')}{x} - \omega_1 x/2 - \omega_3 x/2 - \omega_3' x/2 + 2H_2(y+E)/(L-x) - \omega_2'(L-x) = R \]

Using the horizontal equilibrium of forces \( H_1+H_3+H_3' = 2H_2 \) and gathering terms yields:
Therefore:

\[
y = \frac{x(L-x)(2R + x(\omega_1 + \omega_3 + \omega_3') + 2\omega_2(L-x)) - \frac{Ex}{L}}{4H_2L} - \frac{Ex}{L} \quad \cdots \cdots (6)
\]

Two problems exist in solving for \( y \) using Equation (6). First, the values of \( \omega_1, \omega_2, \omega_3 \) depend on the angles of the subchords \( \overline{AC} \) and \( \overline{CB} \). These are computable only if the \( y \) value for the carriage location is known. A direct solution for \( y \) is therefore not possible and an iterative solution is required.

The second problem is in determining \( H_2 \). This value constrains the solution and is dependent on either the maximum allowable tension in the haulback (breaking strength adjusted by a factor of safety), the maximum line pull that the yarder can exert, or, in extreme conditions, the maximum allowable tension in the mainline. The latter occurs in uphill yarding when the chord (\( \overline{AB} \)) slope is excessive (e.g. 100 percent). This is illustrated in Table XII.
TABLE XII

LINE TENSIONS WITH THE CARRIAGE AT MIDSPAN FOR A MADILL 052 TENSION SKIDDER USING THE PARABOLIC MODEL

LOAD INCLUDING CARRIAGE = 35000 LB.
TENSION (HAULBACK) AT INTERLOCK = 56500 LB.
WEIGHT OF HAULBACK/FOOT = 2.34 LB.
WEIGHT OF MAINLINE+SLACKPULLER = 4.13 LB.

SPAN = 1400 FEET

<table>
<thead>
<tr>
<th>CHORD SLOPE%</th>
<th>PERCENT MIDSPAN DEFLECTION</th>
<th>HAULBACK LNDG-CARR (LB.)</th>
<th>HAULBACK CARR-TAIL (LB.)</th>
<th>MAINLINE + SLACKPULLER (LB.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>9.9</td>
<td>46319.18</td>
<td>36151.88</td>
<td>25984.55</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL COMPONENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>-34372.72</td>
<td>44944.28</td>
<td>-20515.96</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>57679.75</td>
<td>57679.75</td>
<td>33107.59</td>
</tr>
<tr>
<td>50</td>
<td>9.4</td>
<td>54482.28</td>
<td>46930.98</td>
<td>39379.65</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>-16670.04</td>
<td>32306.17</td>
<td>-12942.34</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>56975.50</td>
<td>56975.42</td>
<td>41451.78</td>
</tr>
<tr>
<td>0</td>
<td>8.8</td>
<td>55502.11</td>
<td>55502.10</td>
<td>55502.07</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>8908.62</td>
<td>8908.68</td>
<td>8274.02</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>56212.55</td>
<td>56212.90</td>
<td>56116.13</td>
</tr>
<tr>
<td>-50</td>
<td>8.4</td>
<td>46087.54</td>
<td>52603.00</td>
<td>59118.45</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>30695.34</td>
<td>-17296.91</td>
<td>38898.45</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>55373.86</td>
<td>55373.64</td>
<td>70767.44</td>
</tr>
<tr>
<td>-100</td>
<td>8.4</td>
<td>34869.15</td>
<td>42810.80</td>
<td>50752.39</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VERTICAL</td>
<td>41850.57</td>
<td>-33683.19</td>
<td>60515.73</td>
</tr>
<tr>
<td></td>
<td>AXIAL</td>
<td>54473.23</td>
<td>54473.03</td>
<td>78980.56</td>
</tr>
</tbody>
</table>
Examining the first two cases, the maximum will be expressed in terms of axial tension. If the maximum is due to machine capacity then $T_{\text{max}}$ will occur in the haulback at the yarder. Otherwise it will occur in the haulback at either support A (yarder) or B depending upon which is higher in elevation. In either case $T_{\text{max}}$ can always be expressed at the support B. If it occurs at B no adjustment is required. On the other hand if $T_{\text{max}}$ is at A then the corresponding tension at B can be found by employing the catenary relationship which states that the difference in tension between points on a cable is equal to the cable weight multiplied by the elevation difference between points (Carson, 1971). This is possible providing there is no loss of tension at the carriage in the haulback; i.e., $T_{C1}=T_{C2}$. This is a valid assumption since the carriage rides on sheaves.

Therefore:

- **if $T_{\text{max}}$ occurs at B then** $T=T_{\text{max}}$
- **if $T_{\text{max}}$ occurs at A then** $T=T_{\text{max}}-E_{\omega}^{1}$

The horizontal tension, $H_{2}$, can now be expressed by using the tangent of the cable at B:

$$H_{2} = T\cos(\arctan(\tan_{B2}))$$

Using Equation (4) for the cable slope:

$$H_{2} = T\cos(\arctan\left(\frac{y+E}{L-x} + \frac{(L-x)\omega_{2}}{2H_{2}}\right)) \quad (7)$$

Depending on the iterative technique used for the solution of Equation (6) this expression (7) may not be suitable as $H_{2}$ can not be solved directly. Equation (7) can be rearranged by employing the relation: $1/\cos\alpha = \sqrt{1 + \tan^{2}\alpha}$

The expression for $H_{2}$ becomes: $H_{2}\sqrt{1 + \tan^{2}\alpha_{B2}} = T$
Therefore: \( H_2 \sqrt{1 + \left( \frac{y+E}{L-x} + \frac{(L-x)\omega_2}{2H_2} \right)^2} = T \)

Squaring both sides: \( H_2^2 \left( 1 + \left( \frac{y+E}{L-x} + \frac{(L-x)\omega_2}{2H_2} \right)^2 \right) = T^2 \)

Expanding: \( H_2^2 \left( 1 + \left( \frac{y+E}{L-x} \right)^2 + 2 \left( \frac{y+E}{L-x} \right) \frac{(L-x)\omega_2}{2H_2} + \left( \frac{(L-x)\omega_2}{2H_2} \right)^2 \right) = T \)

Gathering terms and simplifying:
\[
H_2^2 \left( 1 + \left( \frac{y+E}{L-x} \right)^2 \right) + H_2 (y+E)\omega_2 + \left( \frac{(L-x)\omega_2}{2} \right)^2 - T^2 = 0 \quad \ldots (8)
\]

\( H_2 \) can be solved by using the formula for a quadratic equation:
\[
H_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}
\]
with \( a = 1 + \left( \frac{y+E}{L-x} \right)^2 \)
\( b = \omega_2 (y+E) \)
\( c = \left( \frac{(L-x)\omega_2}{2} \right)^2 - T^2 \)

If the critical tension occurs in the mainline then \( H_2 \) must be solved in a more complicated manner. \( H_2 \) must first be expressed in terms of the horizontal tension in the mainline and the slackpuller (\( H_3 \)) as the constraint is now in the mainline and the slackpuller combined. Using the force balance at the carriage:
\[
H_1 + H_3 = 2H_2
\]
\[
H_1 = 2H_2 - H_3
\]
And; \( H_2 = \frac{(H_1 + H_3)}{2} \)

Also, since there is continuity of tension at the carriage in the haulback then:
\[
T_{C1} = T_{C2}
\]
\[
H_1 \sqrt{1 + \tan^2 \alpha_{C1}} = T_{C1}
\]
And \( H_2 \sqrt{1 + \tan^2 \alpha} \frac{c_2}{c_2} = T \)

Using the same expansion in the derivation of Equation (8):

\[
\begin{align*}
    a_2 H_2^2 + b_2 H_2 + \left( \frac{(L-x) \omega_2}{2} \right)^2 &= T^2 \frac{c_2}{c_2} \quad \text{.......................... (9)} \\
    \text{And; } a_1 H_1^2 + b_1 H_1 + \left( \frac{x \omega_1}{2} \right)^2 &= T^2 \frac{c_1}{c_1} \quad \text{.......................... (10)} \\
    \text{with: } a_1 &= 1 + \left( \frac{y}{x} \right)^2 \quad b_1 = -y \omega_1 \\
    a_2 &= 1 + \left( \frac{y+E}{L-x} \right)^2 \quad b_2 = -(y+E) \omega_2
\end{align*}
\]

Substituting the alternate forms of \( H_1 \) and \( H_2 \):

\[ H_1 = 2H_2 - H_3 \]

\[ H_2 = \frac{(H_1 + H_3)}{2} \]

into Equations (9) and (10) gives:

\[
\begin{align*}
    a_1 H_1^2 + b_1 H_1 + \left( \frac{x \omega_1}{2} \right)^2 &= a_1 (2H_2 - H_3)^2 + b_1 (2H_2 - H_3) + \left( \frac{x \omega_1}{2} \right)^2 \\
    \text{And; } a_2 H_2^2 + b_2 H_2 + \left( \frac{(L-x) \omega_2}{2} \right)^2 &= a_2 (H_1 + H_3)^2 + b_2 (H_1 + H_3) + \left( \frac{(L-x) \omega_2}{2} \right)^2
\end{align*}
\]

Simplifying:

\[
\begin{align*}
    a_1 H_1^2 + b_1 H_1 &= a_1 (2H_2 - H_3)^2 + b_1 (2H_2 - H_3) \quad \text{............. (11)} \\
    a_2 H_2^2 + b_2 H_2 &= a_2 ((H_1 + H_3)/2)^2 + b_2 ((H_1 + H_3)/2) \quad \text{.... (12)}
\end{align*}
\]

Using Equation (11) \( H_1 \) can be solved by using:

\[
H_1 = \frac{-b_1 + b_2 b_1 - 4a_1 c_1}{2a_1} \\
c_1 = a_1 (2H_2 - H_3)^2 + b_1 (2H_2 - H_3)
\]

This expression for \( H_1 \) can now be used in Equation (12) which will yield an expression for \( H_2 \) in terms of \( H_3 \) only:

\[ H_3 = T \cos(\arctan \left( \frac{y}{x} - \frac{x \omega_3}{2H_3} \right)) \]

\( H_3 \) can be found by rearranging and using the same approach used to obtain Equation (8). Therefore:

\[
H_3 \left[ 1 + \left( \frac{y}{x} \right)^2 \right] - H_3 (y \omega_3) + \left( \frac{x \omega_3}{2} \right)^2 = T^2
\]
\[ H_3 = -b_3 + \frac{\sqrt{b_3^2 - 4a_3c_3}}{2a_3} \]

with \( a_3 = 1 + \left( \frac{y}{x} \right)^2 \)
\( b_3 = -y\omega_3 \)
\( c_3 = \left( \frac{x\omega_3}{2} \right)^2 - T^2 \)

and; \( T \) is the maximum mainline+slackpuller line tension at the carriage.

If the carriage is above the support \( A \) then \( T = T_{\text{max}} \). Conversely, if the carriage is below \( A \) then \( T = T_{\text{max}} - y\omega_3 \). This value of tension, \( T_{\text{max}} \), is for both the mainline and slackpuller combined. Separating the two requires knowledge of their working configuration in the carriage. As seen from Figure 58 the carriage drum is in equilibrium when the tension in the mainline (\( T_m \)) is equal to the tension in the slackpuller (\( T_s \)) plus weight of the load (not including the carriage).

Therefore:
\[ T_m = T_s + R' \quad (R' \text{ is the weight of the logs}) \]

The combined tension, \( T_{\text{max}} \), is:
\[ T_{\text{max}} = T_m + T_s = 2T_s + R' \]

So, if the mainline tension is the limiting value then rearranging the above in terms of \( T_m \) only yields:
\[ T_{\text{max}} = 2T_m - R' \quad \text{.......................... (13)} \]

Solution of \( H_2 \) using Equation (12) can now be achieved. Iteration is required as a closed form is not possible.
Figure 58. Configuration of the Lines at the Carriage Drum
Deciding Where Critical Tension Occurs in the Cable System

The critical tension can be due to one of the following three reasons:

1) line capacity of the haulback is exceeded ($T_h$)
2) line pull capacity of the yarder is exceeded ($T_y$)
3) line capacity of the mainline is exceeded ($T_m$)

The load position, $y$, is found first using either $T_h$ or $T_y$, whichever is less, for $T_{\text{max}}$. Then $H_3$ is computed by finding $H_2$ and $H_1$.

$H_1$ is evaluated using Equation (10) and $H_2$ is found when the value of $y$ is determined in the iterative technique. Therefore using the equilibrium of horizontal forces at the carriage $H_3 = 2H_2 - H_1$. The tension in the mainline+slackpuller at the carriage, $T_{C3}$, is found from:

$$T_{C3} = H_3 \sqrt{1 + \tan^2 \alpha_{C3}}$$

with $\tan^2 \alpha_{C3} = \left( \frac{y}{x - \frac{x \omega_3}{2H_3}} \right)^2$

This is converted to the maximum tension ($T_{\text{max}}$) in the mainline + slackpuller:

if the carriage is above support A then $T_{\text{max}} = T_{C3}$
if the carriage is below support A then $T_{\text{max}} = T_{C3} + y \omega_3$

The maximum allowable mainline tension ($T_m$) is exceeded if:

$$T_m < \frac{T_{\text{max}} + R'}{2}$$ from Equation (13)

If the critical mainline tension has not been exceeded then the solution for $y$ is acceptable. If not $y$ must be resolved using $H_2$ as defined by Equation (12).
Derivation of the Model for a Three Line Cable System

The three line cable configuration is commonly referred to as a shotgun or gravity system (Figure 51). The formulation is identical to that for a five line system. The resultant equation for the load displacement, $y$, is:

$$y = \frac{x(L-x)(2R + \omega_2(L-x) + x(\omega_1 + \omega_3)) - \frac{Ex}{L}}{2H_2 L}$$

The expression for tensions is identical to those for the five line system except the horizontal force balance equation at the carriage is $H_1 + H_3 = H_2$. The appropriate changes are needed in Equations (11) and (12). The solution when the critical tension is in the snubbing line (line segment 3) is simpler as there is only one line and no separation of tensions is required as was done for the slackpuller and mainline (Equation (13)).
APPENDIX D

PERSPECTIVE PLOTTING

Three dimensional representations using perspective are created in a similar manner to orthogonal projections (Newman and Sproull, 1973). In addition to specifying the viewing angles, about ($\theta$) and above ($\phi$), the location of the viewpoint ($x_v, y_v, z_v$) and the viewing aperture ($\alpha$) are required. These viewing controls are depicted in Figure 59.

The rotation angles $\theta$ and $\phi$ are applied to each $(x, y, z)$ data point in the same way as for the orthogonal projection.

$$(x', y', z', 1) = (x, y, z) \begin{bmatrix} \cos \theta & (-\sin \theta \cos \phi) & (\sin \theta \sin \phi) & 0 \\ \sin \theta & (\cos \theta \cos \phi) & (-\cos \theta \sin \phi) & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The coordinate system $(x', y', z')$ for the object is then translated such that the viewpoint $(x_v, y_v, z_v)$ becomes the origin. Therefore:

$$(x'', y'', z'', 1) = (x', y', z', 1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_v & -y_v & -z_v & 1 \end{bmatrix}$$

These two transformations (rotation and translation) can be combined:

$$(x', y', z', 1) = (x, y, z, 1) \begin{bmatrix} \cos \theta & (-\sin \theta \cos \phi) & (\sin \theta \sin \phi) & 0 \\ \sin \theta & (\cos \theta \cos \phi) & (-\cos \theta \sin \phi) & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ -x_v & -y_v & -z_v & 1 \end{bmatrix}$$
Figure 59. Viewing Controls for Perspective Projection.
The object can now be represented in three dimensions by projecting the \((x',y',z')\) points, that describe the object, back towards the viewpoint onto a screen placed between the object and the viewpoint as shown in Figure 60. Assume the screen is square in shape with width 2\(b\) and is placed at a distance "\(a\)" from the viewpoint. Using similar triangles (\(\triangle EBP = \triangle EAQ\)) the point \(P(x',y',z')\) is projected onto the screen as \(P(x_s,y_s,z_s)\) with:

\[
\frac{x_s}{a} = \frac{x'}{y'} \quad \text{and} \quad \frac{z_s}{a} = \frac{z'}{y'}
\]

Therefore:

\[
x_s = ax' \quad \text{and} \quad z_s = az'
\]

These points can be scaled to the screen size by dividing by \(b\):

\[
x_s = \left(\frac{a}{b}\right)x' \quad \text{and} \quad z_s = \left(\frac{a}{b}\right)z'
\]

The values of \(a\) and \(b\) are never required as their ratio \((a/b)\) represents the cotangent of the viewing aperture angle \((\alpha)\). Then,

\[
x_s = \frac{x'}{y'tan(\alpha/2)} \quad \text{and} \quad z_s = \frac{z'}{y'tan(\alpha/2)}
\]

The object is now plotted on the two-dimensional screen using the coordinates \((x_s,z_s)\). The original \(y'\) coordinate is not required as no depth is used. The point \((x_s,z_s)\) becomes \((x,y)\) of the screen coordinate system.

Depending upon the relative sizes of the aperture angle and the object certain regions of the projection could lie outside the viewing pyramid. Points \((x_s,z_s)\) will lie inside the viewing pyramid if:
Figure 60. Perspective Projection.
\[-y' < \frac{x'}{\tan(a/2)} < y'\]
\[-y' < \frac{z'}{\tan(a/2)} < y'\]

If a point lies outside, it is not plotted as part of the projection.

To obtain a clean and eye-pleasing plot, lines that cross the viewing pyramid boundary should be truncated at the boundary and not merely stopped at the closest viewable point. This clipping problem is discussed by Newman and Sproull (1973); a suitable algorithm is also provided.

The removal of hidden areas can be done in the same manner as for orthographic projections.

Perspective plotting, although more realistic, is far more difficult to implement than orthographic projection. The increased execution time becomes excessive especially when attempting to "clip" areas outside the viewing area (unless it can be done automatically by the plotting device being used). For this reason it was not used in the Terrain Simulator package developed for the Hewlett-Packard 9830A desktop computer system.