THE COMPUTER-AIDED DESIGN OF MOLDBOARD PLOW SURFACES USING THREE DIMENSIONAL GRAPHIC TECHNIQUES

By

Stephen Brian Richey

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Agricultural Engineering

1987

Approved by:

Ajit K. Srivastava

Major Professor

[illegible]

Department Chairman

Feb 20, 1987

Date

ABSTRACT

THE COMPUTER-AIDED DESIGN OF
MOLDBOARD PLOW SURFACES USING
THREE DIMENSIONAL GRAPHIC TECHNIQUES

By

Stephen Brian Richey

A computer model has been developed to show that three dimensional graphic techniques can be used to describe moldboard plow surfaces. A three dimensional soil-tool mechanics was developed to calculate the forces applied to the plow by the soil. The forces exerted on the plow by the soil are predicted without prior knowledge of the soil path over the surface.

The plow shape is defined by two Bezier surfaces. Each surface can be divided into a maximum of 100 elements. Force calculations are made for each element and then the forces and moments are summed over all the elements to determine the totals for each of the x, y and z directions. The soil path is determined by the surface shape and the forces exerted on the soil.

The force predictions were checked against measurements made by Emmet and Gunkel (1983) on a 16 inch Ford moldboard plow. Predicted draft forces were 15 percent larger than those measured.

This thesis is dedicated to Mary.

ACKNOWLEDGMENTS

The author gratefully acknowledges Dr. Ajit Srivastava, major professor, for his guidance of the research effort and his patience and understanding while this manuscript was completed in the two years that followed.

Many thanks to Dr. Larry Segerlind and Dr. Robert Wilkinson, committee members, for their support and advice while working towards the completion of this graduate degree.

There are many people that contributed to this research in small but important ways. Their assistance made the research easier and the daily activities enjoyable. The author specifically wishes to recognize Cliff Zehr, Tim Oliver, Wilbur Mahoney, Tom Esch and Luke Reese.

Deepest thanks are expressed to the author's parents, Fred and Arlene Richey, for their continual encouragement and support in reaching for my goals.

The person who made the most personal sacrific for this research was the author's wife, Mary. For her sacrific, encouragement and assistance the author extends his deepest heart-felt graditude.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	Adhesion of Soil on Metal
B	Acceleration Force of Soil
b	Tool Width
C	Soil Cohesion
D	Direction Cosine
d	Depth of Tool Cut
d_s	Distance the Soil Travels Along the Forward Failure Shear Plane
d_p	Distance the Soil Travels on the Tool Over the Described Path
F_x	Draft Force
F_y	Lateral Force
F_z	Vertical Force
F_0	Contact Area of Soil Block and Inclined Blade
F_1	Area of Forward Shear Failure Surface
G	Weight of Soil Segment
g	Acceleration Due to Gravity
h	Distance From One Shear Plane to Next
m	Mass of Accelerated Soil
N_0	Normal Load on Inclined Blade
N_1	Normal Load on Leading Forward Failure Surface
N_2	Normal Load on Trailing Forward Failure Surface
t	Time
V_A	Absolute Soil Velocity

V_e	Velocity of Soil Along Inclined Blade
V_s	Velocity of Soil Along Shear Plane
V_t	Forward Velocity of the Inclined Blade
V_0	Initial Soil Velocity
v	Velocity of Accelerated Soil
v_0	Velocity of the Tool
α	Angle Described in Figure (3.7)
β	Angle of Forward Failure Surface
γ	Bulk Density of Soil
δ	Lift Angle of Tool
η	Angle Described in Figure (3.9)
μ	Soil Internal Friction Coefficient
μ'	Soil-metal Friction Coefficient
σ	Normal Stress
τ	Shear Strength
ϕ	Angle of Internal Friction
ϕ'	Angle Described in Figure (3.8)

Chapter 1

INTRODUCTION

Ancient Egypt, Greece and Rome tilled their fields with a "Y" shaped stick pulled by draft animals or slaves. This plow design was practically unchanged until the 18th Century (Carlson, 1961) when formed metals were used for the moldboard (Gill and VandenBerg, 1967). Over many years moldboard plow shapes have evolved through the 'trial and error' design efforts of many individuals. The modern moldboards are very complex surfaces that have required elaborate and restrictive shape description techniques to manufacture.

1.1 Computer Graphics

The aeronautics industry has developed graphic techniques to describe complicated shapes of aerodynamically efficient components (Foley and VanDam, 1982). These techniques are directly applicable to the complex surfaces of moldboard plows. They are very general and can represent virtually any surface. The technique used in this study is a Bezier bicubic parametric surface.

Graphic techniques by their nature also offer a very effective communication media between the computer and the operator. The ancient Chinese proverb "one picture is worth a thousand words" is also very applicable when communicating with the computer. Graphic displays are more natural and efficient than the 'traditional' alphanumeric displays used on most computers (Foley and VanDam, 1982).

1.2 Soil-Tool Mechanics

Soil-tool mechanics offers a method of describing the application of forces to the soil and how the soil reacts. It was pioneered in the 1930's by M. L. Nichols and applied to a simple two dimensional inclined blade in 1956 by W. Soehne (Gill and VandenBerg, 1967). Other researchers have developed mechanics that are dependent on emperically derived soil paths (Emmet, 1983; O'Callaghan and McCoy, 1965).

This study modifies Soehne's two dimensional inclined blade mechanics and expands it to three dimensions for application to a moldboard plow. The soil path is calculated based on the plow surface shape and forces exerted on the soil.

1.3 Research Objectives

Research reported here sought to combine computer graphics and three dimensional soil-tool mechanics to predict the force required to move a moldboard plow through the soil. The specific objectives were:

1. To develop a computer model that:

   A. Uses three dimensional graphics techniques to describe a share and moldboard surface.

   B. Evaluates three dimensional force and moment reactions of a single plow without prior knowledge of the soil path over the plow surface.

C. Allows the user to specify a plow shape, size and set of operating conditions.

2. To verify the model results using existing force data for a moldboard plow.

Chapter 2

LITERATURE REVIEW

2.1 Plow Description

One of the first moldboard description techniques for construction purposes was developed by Thomas Jefferson (Gill and VandenBerg, 1967). The rigid framework of Figure (2.1) was its' basis. Using lines e-m and o-h as guides, a straight edge is moved from e-o to m-h so that the straight edge remains in a plane parallel to the y-z plane. The path swept out by this line describes the surface.

Several other moldboard shapes have been constructed with a modified Jefferson framework. The guides and straight edge were changed from straight lines to various catenaries, arcs of circles, etc. Intuition was always the basis for choosing one of these shapes over another (Gill and VandenBerg, 1967).

White (1918) was the first to mathematically describe the surface generated by this method. He achieved this by moving a reference cartesian coordinate system around on the surface, reducing the description to a series of hyperbolic paraboloids (Gill and VandenBerg, 1967).

Several graphical techniques have been used by researchers to describe the shape of the surface. The most common method is to project the shape onto a gridded plane

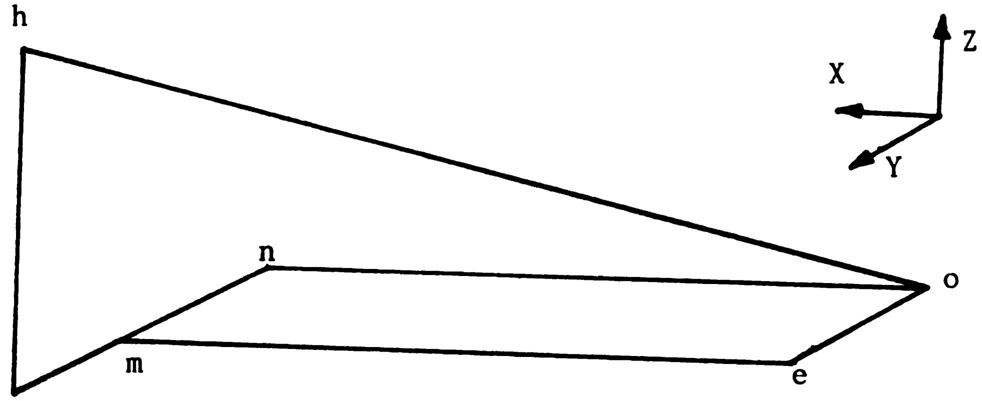

Figure 2.1. Thomas Jefferson's Framework for Designing a Moldboard Plow.

perpendicular to the direction of travel and then on to a plane parallel to the direction of travel. For example, if the relative position of a point in the perpendicular planes is known, a contour line for constant x value can be determined. Therefore, by plotting the contour lines in any one of the coordinate directions and connecting the contour line ends, a two dimensional representation of the surface is drawn (Gill and VandenBerg, 1967).

Soehne (1959) photographed a slit of light projected on the plow surface. By keeping the camera and light projector stationary and moving the surface incrementally, the same coordinate dimensions as described in the previous paragraph could be found.

Although moldboard shapes had to be defined for fabrication purposes, they were also defined in hopes of relating plow shape to forces applied by the soil or the resulting soil conditions. In this research, the actual path the soil travels over the moldboard was determined. Gill and VandenBerg (1967) reported Pfost, Corley and Soehne in separate efforts, covered the moldboard surface with a thin coat of lacquer. The scratches left in the lacquer after plowing a short distance were recorded using techniques previously described (Gill and VandenBerg, 1967).

Nichols and Kummer (1932) developed a method of mathematically describing the surface using polar coordinates and arcs independent of soil path. Their method allowed them to look at three different functional areas and evaluate the soil-tool mechanics for each area.

With the advent of the digital computer, several researchers (Carlson, 1961; O'Callaghan and McCoy, 1965; Emmett, 1983; Orlandea, et al., 1983) fit multi-order polynomials to these scratch traces and the contour lines. With the development of hardware that allows the computer to read the x, y and z coordinate values directly (Emmett, 1983), a very accurate description of the surface and/or soil path has been implemented.

2.2 Soil-Tool Mechanics

M. L. Nichols was the pioneer in soil tillage mechanics. He worked with various researchers in studying the dynamic properties of soil and their influence on tillage operations. He attempted to explain soil forces using colloid film theory (Nichols, 1931a). Nichols (1931b) also researched the evaluation of soil-metal friction coefficients.

Nichols and Kummer (1932) deduced there are three different frictional areas of a moldboard plow. The lower portion, forming a wedge for breaking the soil loose, a central pulverization area and a turning or inversion area on the upper portion. Using this theory they defined and designed each of these areas to manipulate the soil as desired.

Nichols and Reed (1934) found that two shear planes are formed as a moldboard plow is pulled through the soil. The primary shear plane is 45 degrees to the right of the direction of travel and the secondary plane is 90 degrees from the primary plane which is parallel to the leading edge of the share.

Until the 1950's, the evaluation of moldboard plow performance was largely based on visual observations. Nichols, Reed and Reaves (1958) evaluated plow-share design by observing the soil action in a glass sided soil bin and then measuring the draft forces. The effects of shape, wear, angle of approach and soil conditions were considered.

In 1956, Soehne analyzed the action of a simple inclined plane (Gill and VandenBerg, 1967). Four behavior equations described the tillage action. They are soil-metal friction, shear failure, acceleration of a soil block after forming shear planes and cutting resistance. Rowe and Barnes (1961) added an adhesion term to soil-metal friction.

The inclined plane moving through the soil was assumed to be in equilibrium. Using a force balance, the draft and vertical loads could be computed (Rowe and Barnes, 1961; Gill and VandenBerg, 1967). Several variations of this analysis have been studied.

Rowe and Barnes (1961) concluded that the rate of shearing of the soil was more important than acceleration and should be included in the mathematical model. Hettiaratchi and Reece (1967), Kawamura (Gill and VandenBerg, 1967), McKyes (1977), Perumpral, et al. (1983), and Siemens, et al. (1965) defined the shear planes differently and completed analysis accordingly.

Carlson (1961) fit multi-order polynomial equations to soil path scratches. He then differentiated the soil displacements twice to get accelerations. Emmett and Gunkel (1983) and O'Callaghan and McCoy (1965) also implemented this technique to compute accelerations for draft prediction models.

Chapter 3

SOIL-TOOL MECHANICS MODEL

3.1 Two Dimensional Model

W. Soehne developed the soil-tool mechanics model for an inclined blade using four simple behavior equations. They are soil-metal friction, shear failure, acceleration force of each soil block and soil cutting resistance (Gill and VandenBerg, 1967). Rowe and Barnes (1961) added an adhesion term to the soil-metal interface stress equation. The following assumptions have been made to realize the mechanics (Rowe and Barnes, 1961).

1. The soil failure ahead of an implement is by a series of shear failures.
2. The angle of the forward shear failure plane is dependent only on the soil internal frictional properties.
3. The soil internal shearing strength and the soil-metal shearing strength can be approximated by a linear function of the stress normal to the surface of shear or of sliding.
4. The soil is considered to be a homogeneous and isotropic material.

The following development of the two dimensional soil-tool mechanics is from Gill and VandenBerg (1967). Figure (3.1) shows the forces that act on a soil block as the blade advances. Forces CF_1 and μN_1 are soil cohesion and internal frictional forces. They are due to soil shear.

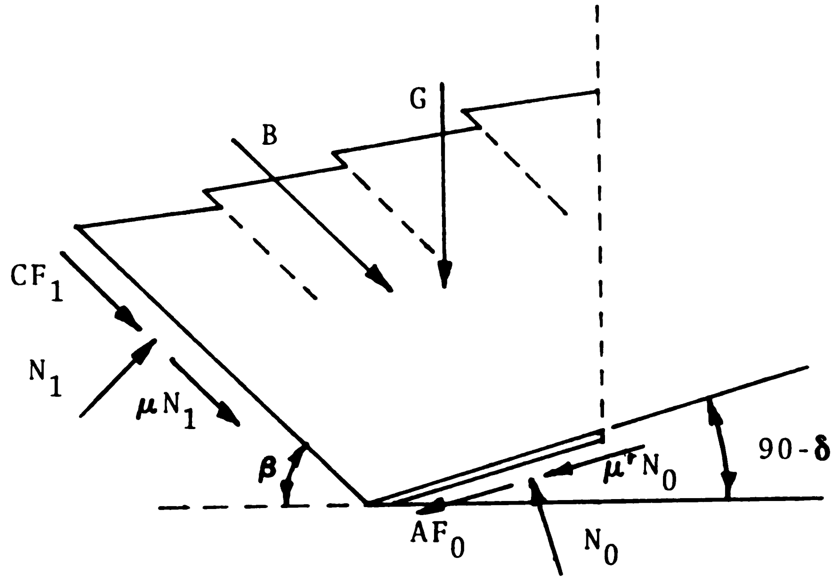

Figure 3.1. Forces on a Soil Segment Reacting on an Inclined Plane Tillage Tool.

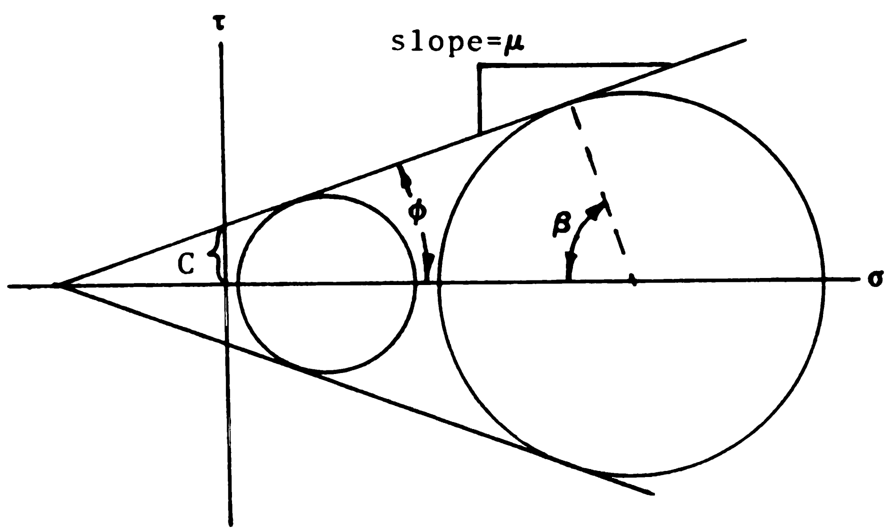

Figure 3.2. A Mohr Envelope of Soil Internal Stresses.

CF_1 is present only at the instant the shear failure plane forms. Forces $\mu' N_0$ and AF_0 are soil-metal friction and adhesion forces. Force B is due to the soil acceleration and G is the soil weight. The pure cutting resistance force occurs at the edge of the tool. Soehne determined that the pure cutting resistance of the soil is very small and only important when the tool is dull or when there are stones or organic matter present. Therefore, it is not included in this analysis.

The shear strength of the soil can be expressed as

$$\tau = C + \mu\,\sigma \tag{3.1}$$

where

τ = soil internal shear strength
C = apparent soil cohesion
μ = soil internal friction coefficient
σ = normal stress on shear surface

Mohr, according to Gill and VandenBerg (1967), proposed the following shear failure theory. Shear strength and normal stress have a functional relationship on the failure plane. If a series of different stress states are applied to the same material and plotted as Mohr's circles, a line drawn tangent to all the circles defines a shear failure envelope (Figure (3.2)). The soil internal friction coefficient can be expressed as

$$\mu = \tan\phi \tag{3.2}$$

where ϕ is the angle of soil internal friction. The shear stress at zero normal stress is the soil cohesion.

The parameter used to geometrically describe the soil internal friction is beta. Figure (3.1) shows the angle beta. Numerically it is

$$\beta = 1/2\ (90 - \tan^{-1} \mu) \tag{3.3}$$

Like the shear strength of soil, the stress at the soil-metal interface can be expressed as

$$\tau = A + \mu'\sigma \tag{3.4}$$

where

τ = shear stress at soil-metal interface
A = adhesion of soil on metal
μ' = soil-metal friction coefficient
σ = normal stress on soil-metal interface

Again plotting several stress states the failure envelope can be drawn (Figure (3.3)). The adhesion and soil-metal friction are the y-intercept and slope, respectively.

The reaction of soil to an inclined plane is assumed to be repeating failure by shear forming small blocks of soil (Figure (3.4)). As one soil block moves onto the incline another soil block moves off. The sum of the forces the soil blocks exert on the incline will be the force required to propel the blade through the soil.

Using the notation in Figure (3.5), an equation can be written for a soil block as follows

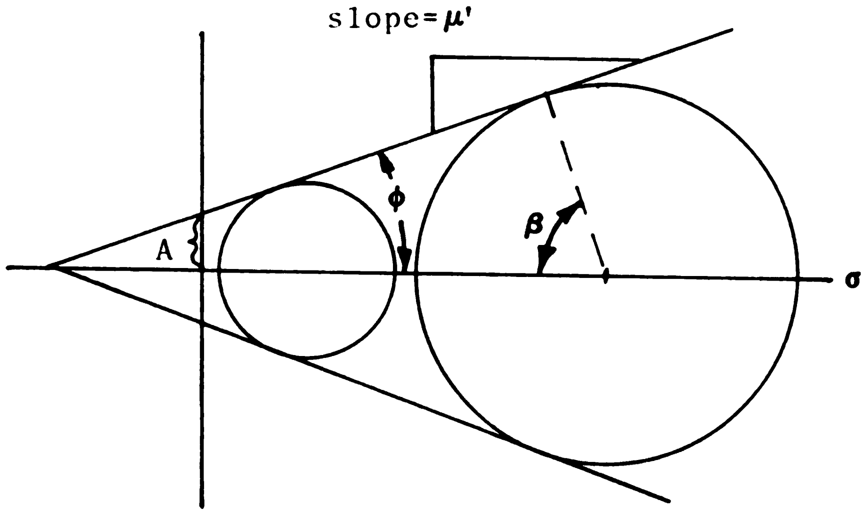

Figure 3.3. A Mohr Envelope of Soil-Metal Stresses.

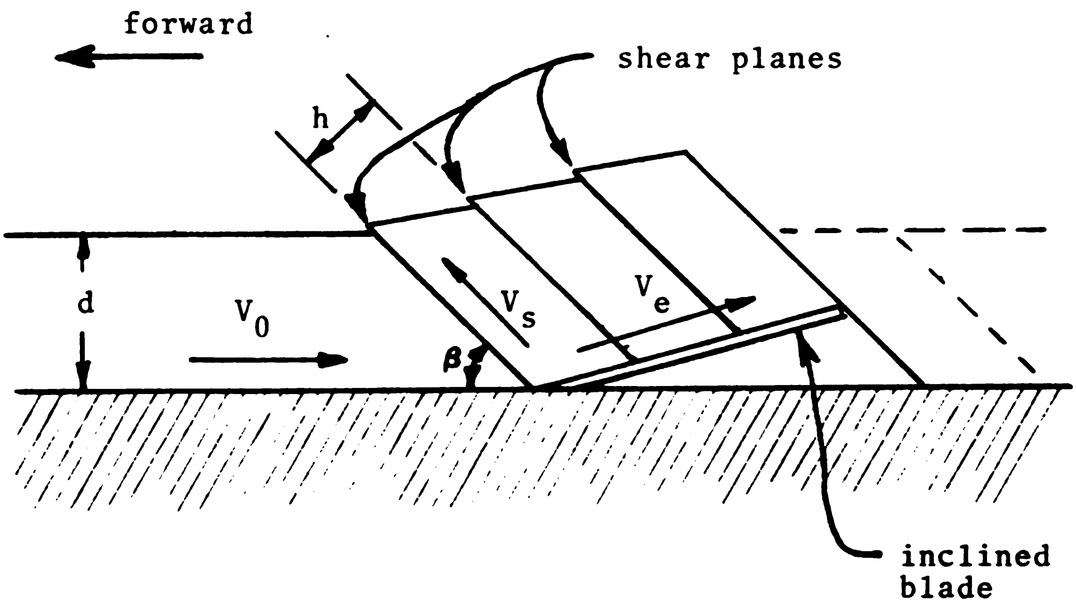

Figure 3.4. Formation of Soil Blocks by Repeated Shear Failure.

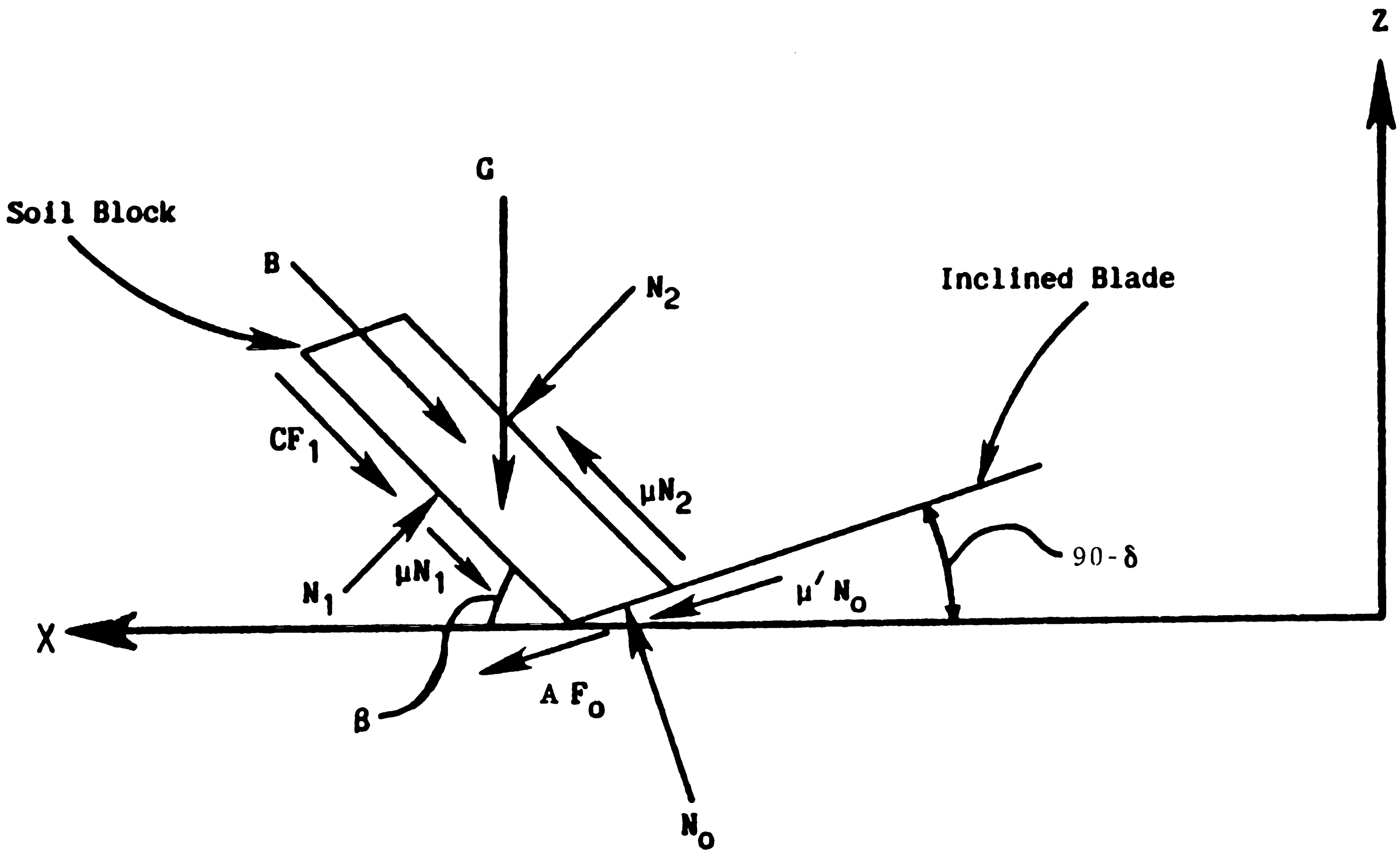

Figure 3.5. A Free Body Diagram of a Soil Slice for the Three Dimensional Model.

$$(B + CF_1 + \mu N_1 - \mu N_2) \cos(\delta - \beta) +$$

$$(N_2 - N_1) \sin(\delta - \beta) + G \sin \delta - N_0 = 0 \qquad (3.5)$$

where

B = acceleration force of soil

C = cohesion of soil

F_1 = area of forward shear failure surface

μ = coefficient of internal soil frictions

N_1 = normal load on leading forward failure surface

N_2 = normal load on trailing forward failure surface

G = weight of soil segment

N_0 = normal load on inclined blade

δ = lift angle of tool

β = angle of forward failure surface

Upon close inspection of the free-body diagram, it can be shown by summing forces perpendicular to the forward failure surface that the difference between N_1 and N_2 is a component of the soil block weight G. As the soil blocks are reduced in size N_1 will equal N_2, thus reducing Equation (3.5) to

$$N_0 = (B + CF_1) \cos (\delta - \beta) + G \sin \delta \qquad (3.6)$$

The weight of soil may be calculated from the volume of soil between shear planes. From Figure (3.4)

$$G = \frac{\gamma\, bhd}{\sin \beta} \qquad (3.7)$$

where

γ = bulk density of soil

b = tool width

h = distance from one shear plane to next

d = depth of tool cut

β = angle of forward failure surface

The area of shear F_1 can be computed from Figure (3.4)

$$F_1 = \frac{bd}{\sin \beta} \tag{3.8}$$

The acceleration force B can be derived from Newton's Second Law of Motion

$$B = m \frac{dv}{dt} \tag{3.9}$$

where

m = mass of accelerated soil

v = velocity of accelerated soil

t = time

If the periodic failure process is considered to be continuous, the total work done for acceleration to occur does not change significantly. Therefore, an average constant force can be determined. The mass to be accelerated in time t is assumed to be the volume of soil disturbed in that time.

$$m = \frac{\gamma}{g} bdtV_0 \tag{3.10}$$

where

γ = wet bulk density

b = width of tool

d = depth of tool

t = time

g = acceleration due to gravity

V_0 = velocity of the tool

The soil can be visualized to be at rest and in time t it reaches a velocity V_s, as shown in Figure (3.4). Therefore,

$$\frac{dv}{dt} = \frac{V}{t} = \frac{V_s - 0}{t - 0} = \frac{V_s}{t} \tag{3.11}$$

The magnitude of the velocity vectors are related so that they form a closed triangle. From Figure (3.4)

$$V_0 = V_s \cos \beta + V_e \cos \delta \tag{3.12}$$

and

$$V_s \sin \beta = V_e \sin \delta \tag{3.13}$$

Eliminating V_e gives

$$V_s = V_0 \frac{\sin \delta}{\sin(\delta + \beta)} \tag{3.14}$$

Substituting Equations (3.10), (3.11) and (3.14) into Equation (3.9) and simplifying gives

$$B = \frac{\gamma}{g} b d V_0^2 \quad \frac{\sin \delta}{\sin(\delta + \beta)} \tag{3.15}$$

C and B are soil parameters determined by soil type. All the variables of Equation (3.6) are now known, therefore, N_0 can be computed.

With N_0 known, a free body diagram of the inclined blade is considered. From Figure (3.6), forces in the x and z direction can be summed.

$$F_x = (AF_0 + \mu' N_0) \cos \delta \tag{3.16}$$

$$F_z = (AF_0 + \mu' N_0) \sin \delta + N_0 \cos \delta \tag{3.17}$$

where

F_x = draft force
F_z = vertical force
A = soil-metal adhesion
F_0 = contact area of soil block and inclined blade
μ' = coefficient of soil-metal friction

Forces computed in Equations (3.16) and (3.17) are due to individual soil blocks. Total inclined blade forces are summations over the number of soil blocks maintaining contact. The thickness of the block, h in Figure (3.4), can be selected for the desired resolution of the model.

The first soil block on the blade is different from the others. The soil cohesion term is due to the force required to form the forward shear plane. Once the shear plane is

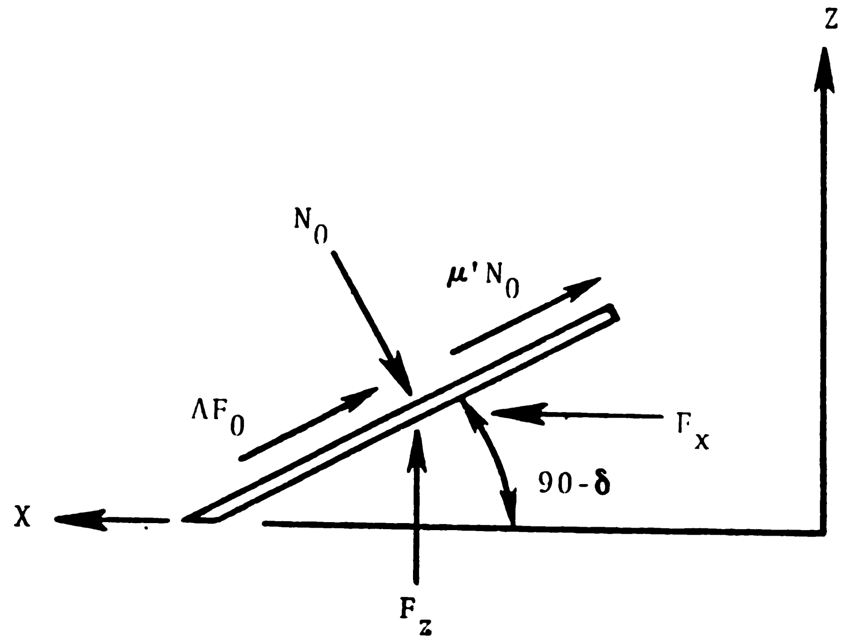

Figure 3.6. A Free Body Diagram of an Inclined Blade.

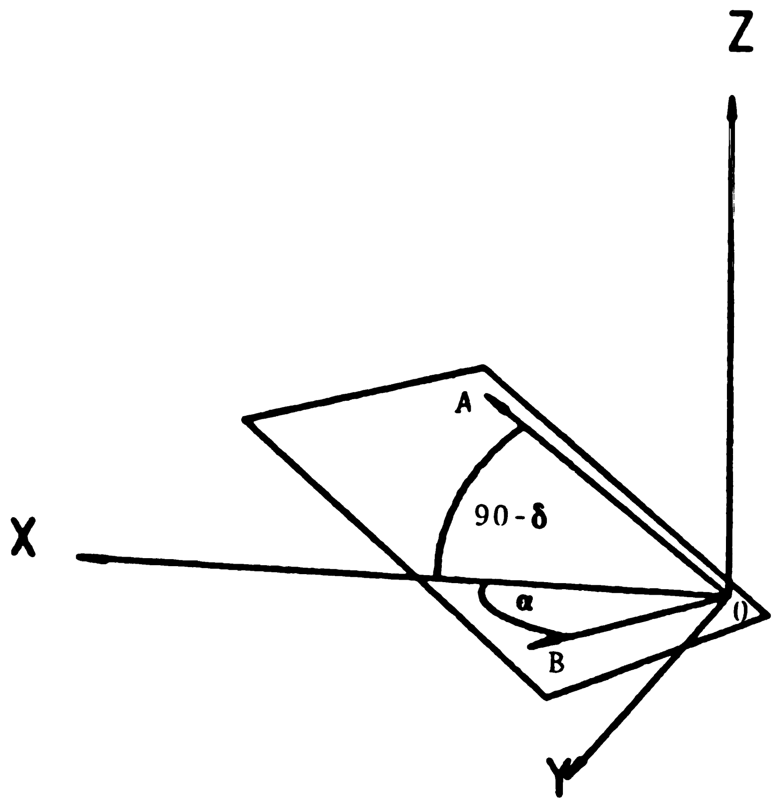

Figure 3.7. Three Dimensional Representation of an Inclined Plane Tillage Tool.

formed, only the soil internal frictional force acts along it.

3.2 Three Dimensional Model

The three dimensional model takes into account the rotation of the inclined blade about z-axis, so that as the blade moves through the soil, there is a side force parallel to the y-axis. Figure (3.7) illustrates this orientation of the blade as it moves in the negative x-direction.

Two angles are used to describe the blade orientation in three space. Delta δ is the angle between OA and the z-axis in Figure (3.7). OA is a line in the plane of the inclined blade and in the x-z plane. Alpha α is the angle between OB and the x-axis. OB is a line in the plane of the inclined blade and in the x-y plane.

To aide in simplifying the force calculations, an angle phi prime ϕ' is defined in terms of alpha α and delta δ. The relationship is

$$\phi' = \tan^{-1} (\sin \alpha \ \tan \delta) \qquad (3.18)$$

Phi prime ϕ' is the angle between CD and CE in Figure (3.8). CD is a line in the plane of the inclined blade and perpendicular to line OB. CE is parallel to the z-axis.

The forces considered in Equation (3.6) are also considered here. The forward failure shear surface is assumed to form perpendicular to line OB of Figure (3.8) and at an angle beta β from the x-y plane. The acceleration and cohesion force act along this plane.

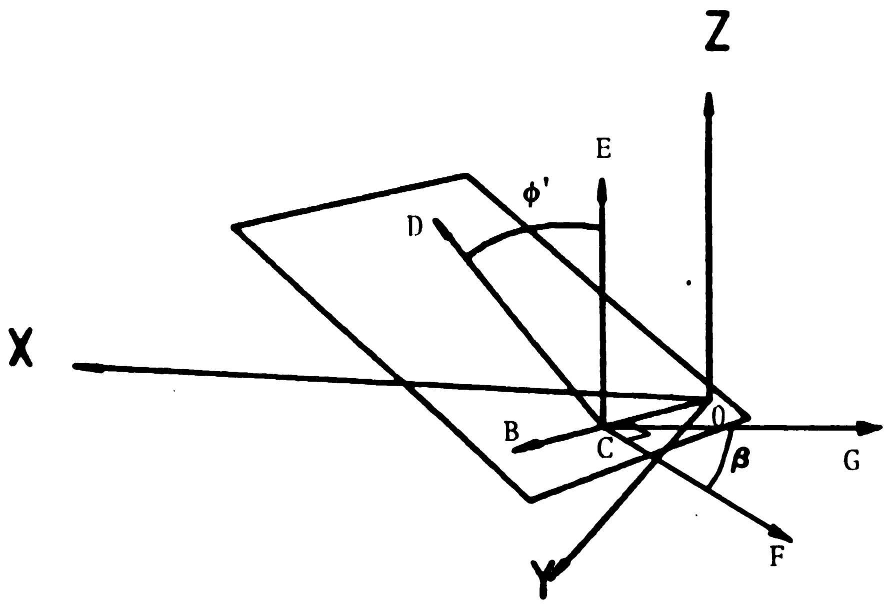

Figure 3.8. Three Dimensional Representation of an Inclined Plane Tillage Tool.

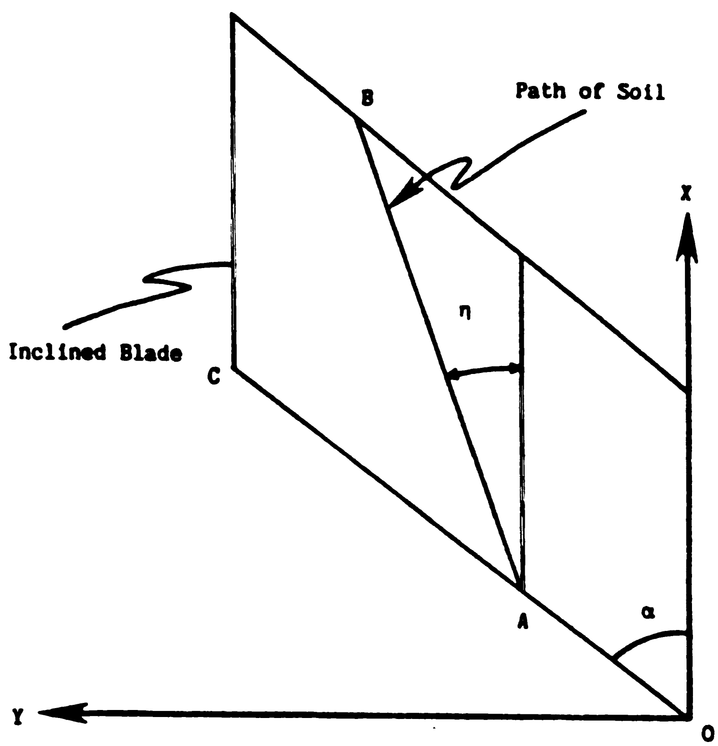

Figure 3.9. Soil Path on the Surface of the Inclined Plane Tillage Tool.

As in the two dimensional case, the soil weight may be calculated from the volume between shear planes

$$G = \gamma h \frac{b}{\sin \alpha} \frac{d}{\sin \beta} \tag{3.19}$$

The area of the forward shear plane F_1 is

$$F_1 = \frac{bd}{\sin \alpha \ \sin \beta} \tag{3.20}$$

Acceleration forces are computed from Newton's Second Law of Motion

$$B = m \frac{dv}{dt}$$

The mass of soil accelerated is

$$m = \frac{G}{g} \tag{3.21}$$

For computation of the change in velocity with time, the displacement of the soil must be considered. The absolute displacement of the soil is along the forward shear plane and in a direction perpendicular to OB of Figure (3.7). The displacement of the soil relative to the incline can be determined empirically with adequate facilities or theoretically if all influencing forces can be quantified. A direction of angle eta η from x-axis and in the plane of the inclined blade was assumed (Figure (3.9)). Further, it was assumed that forward velocity of tool is the velocity at which the soil traverses the blade over the above described path. The time required to traverse the blade is

$$t = \frac{d_p}{v_t} \tag{3.22}$$

where

t = time to traverse blade

d_p = distance the soil travels on the tool over the described path

v_t = forward velocity of the inclined blade

The time required for the soil to traverse the path over the blade (A to B of Figure (3.9)) is the same time required for the soil to displace along the forward failure surface (A to B Figure (3.10)). Therefore, the absolute velocity is

$$V_A = \frac{d_s}{t} \tag{3.23}$$

where

V_A = absolute soil velocity

d_s = distance the soil travels along the forward failure shear plane

t = time to travel d_s

The acceleration is

$$\frac{dv}{dt} = \frac{(V_a - V_0)}{\Delta t} \tag{3.24}$$

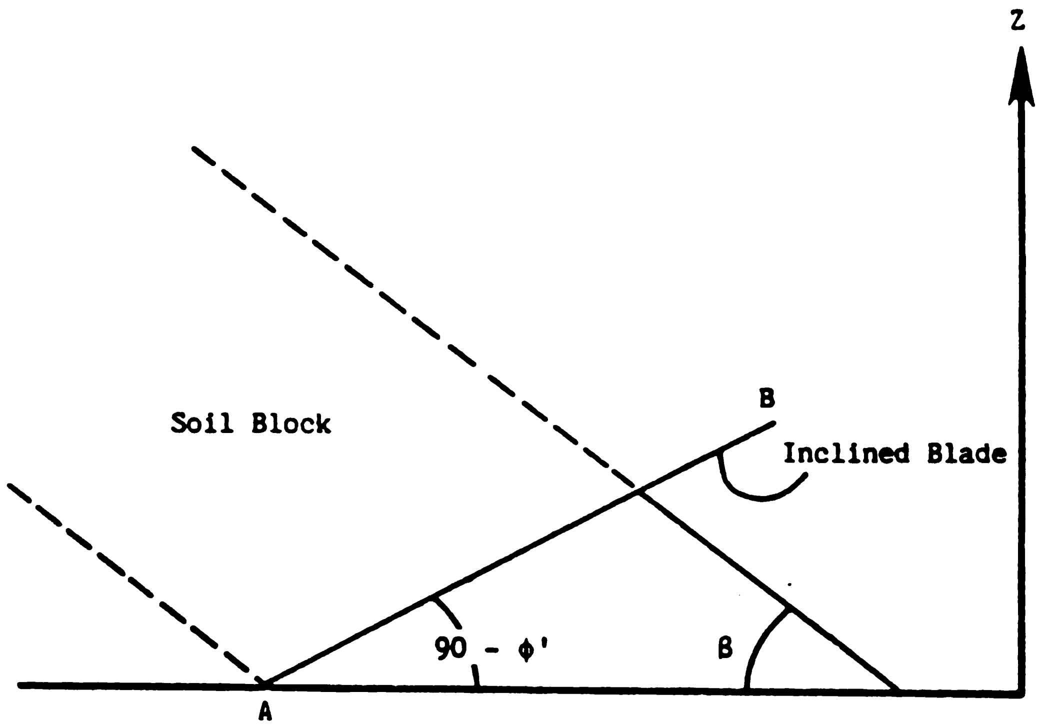

Figure 3.10. Soil and Tillage Tool Orientation in the Three Dimensional Space.

where

V_0 = initial soil velocity

Δt = time for velocity to go from V_0 to V_A

For this case the soil block starts at rest. Substituting Equations (3.20) and (3.23) into Equation (3.8) gives

$$B = \frac{GV_A}{g\Delta t} \tag{3.25}$$

All the variables are now known in Equation (3.6), therefore N_0 can be solved for. As in the two dimensional case, the free body diagram of the inclined blade can be drawn. Note the direction of adhesion and soil-metal frictional forces are in the opposite direction of soil movement. Summation of forces in the x, y and z-directions yields.

$$F_x = -(AF_0 + \mu' N_0) \cos \eta \ \cos \delta + N_0 \cos \phi' \sin \alpha \tag{3.26}$$

$$F_y = -(AF_0 + \mu' N_0) \sin \eta + N_0 \cos \phi' \cos \alpha \tag{3.27}$$

$$F_z = -(AF_0 + \mu' N_0) \cos \eta \ \sin \delta + N_0 \sin \phi' \tag{3.28}$$

CHAPTER 4

GRAPHICS REPRESENTATION

The plow surface is modeled using Bezier parametric bicubic surfaces. A single surface is derived by representing a three dimensional curve and then generalizing the mathematical development to three dimensional curved surfaces. This development is from Foley and VanDam (1982).

4.1 Three Dimensional Curves

There are two ways to represent curves; as functions of x, y and z or as functions of a parameter t, which is a position along a curve. In the first case, it is difficult to represent an infinite slope or a progression of points on a segment if the segment includes a loop. The latter case is a parametric curve that allows closed and multi-valued functions and replaces the slopes with tangent vectors. This eliminates problems with infinite slopes.

Cubic curves are used because they are the lowest order representation of curved segments that provide slope and position continuity at segment endpoints. Higher order representations also tend to have undesirable oscillations. Lower order representations restrict the shape of the curve.

There are many ways to define a cubic parametric curve. Only two are considered here. The Hermite curve representation is shown in Figure (4.1). It uses the

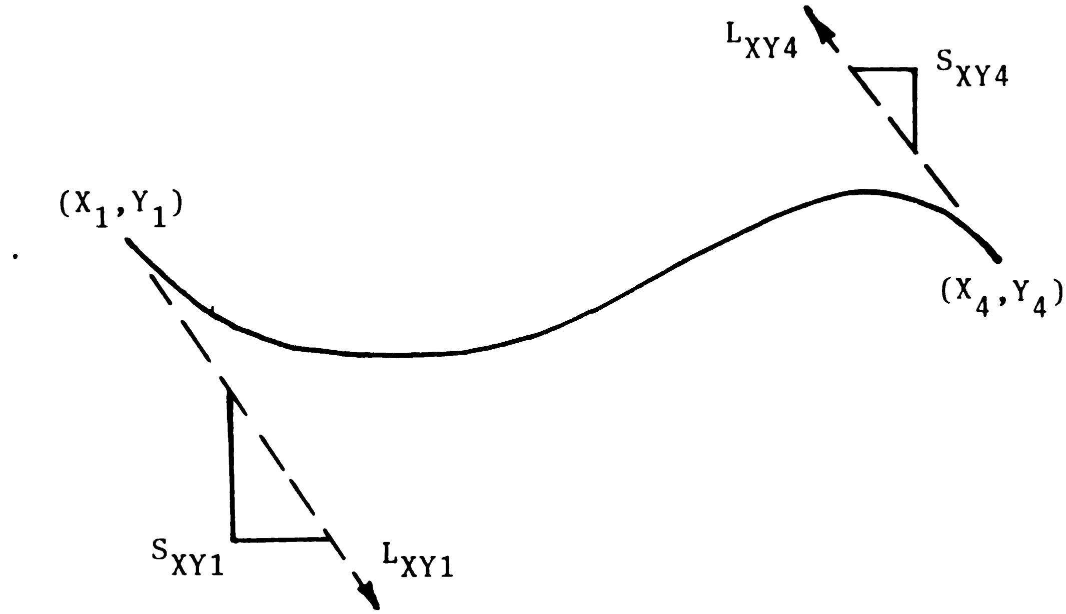

Figure 4.1. Hermite Curve Representation.

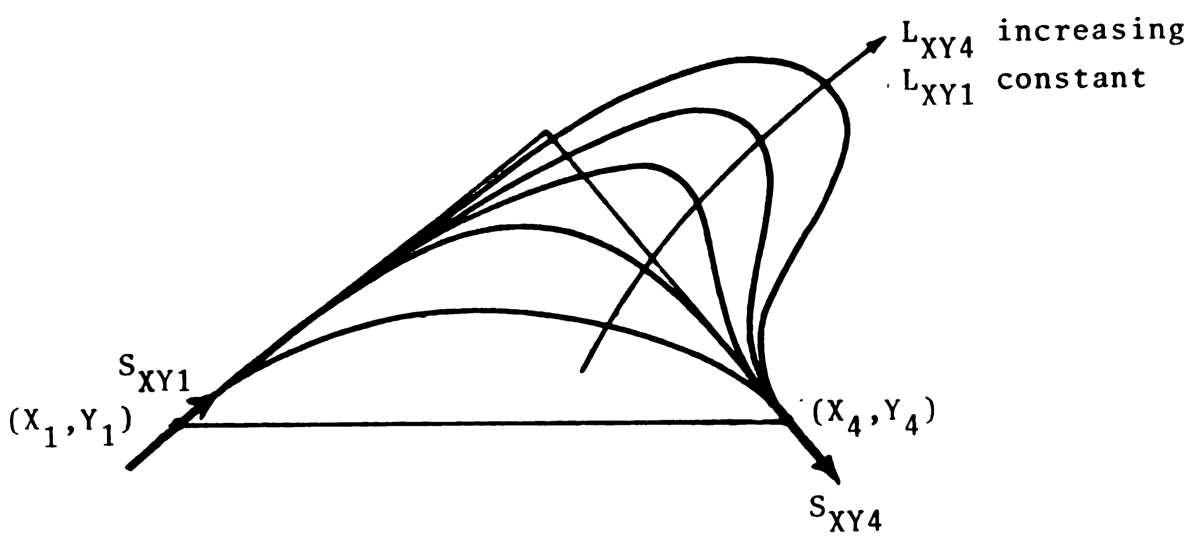

Figure 4.2. Hermite Curve With Varying Tangent Vector Length.

positions of the curved segment endpoints (X_1,Y_1) and (X_4,Y_4); unit tangent vectors from each endpoint in the direction of the other endpoint S_{xy1} and S_{xy4} and tangent vector lengths L_{xy1} and L_{xy4}. The tangent vector lengths describe the 'fullness' of the curve segment. Figure (4.2) shows how the curve changes as $L_{xy}4$ increases with $L_{xy}1$ constant.

The Bezier curve representation is shown in Figure (4.3). It uses the endpoint positions (X_1,Y_1) and (X_4,Y_4) and the position of two control points (X_2,Y_2) and (X_3,Y_3) which, when connected with the endpoints, define both the unit tangent vectors and the tangent vector lengths. The latter method makes the physical meaning of the tangent vectors more apparent. This is important since this model is intended for the design of plow shapes.

The previous description of the Hermite and Bezier curve representations used only two dimensions for clarity. This can be expanded to three dimensions by adding a z coordinate to each point description in the Hermite and Bezier forms and by adding two tangent vectors S_{yz} and S_{zx}, and two tangent vector lengths L_{yz} and L_{zx}, to each endpoint in the Hermite form.

The derivation of the equations for calculating curve point positions follows. The Hermite representation is used in deriving the Bezier form, therefore it is considered first. The derivation uses the subscript x to refer to the x component. The y and z derivations are similar and are

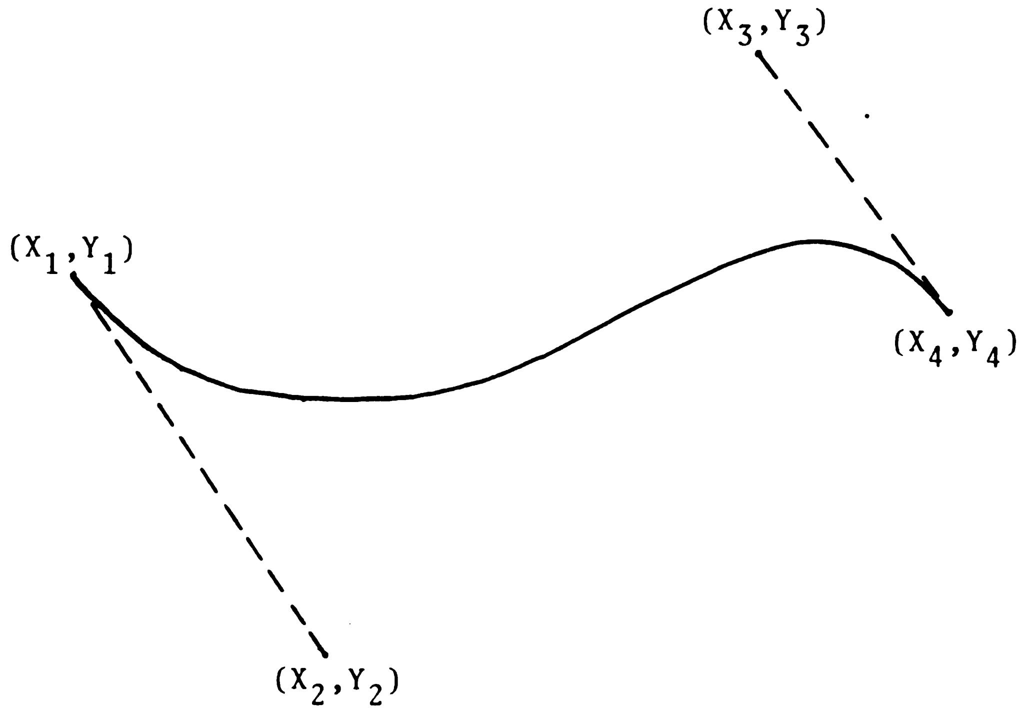

Figure 4.3. Bezier Curve Representation.

not shown here.

A cubic curve parameterized with the variable t is

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \tag{4.1}$$

In matrix form

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \tag{4.2}$$

or

$$x(t) = T\ C_x \tag{4.3}$$

differentiating Equation (4.1) with respect to t

$$\frac{dx}{dt} = 3a_x t^2 + 2b_x t + c_x \tag{4.4}$$

in matrix form

$$x'(t) = \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} \tag{4.5}$$

Subject to Hermite form conditions

$$\begin{aligned} x(0) &= P_{1x} \\ x(1) &= P_{4x} \\ x'(0) &= R_{1x} \\ x'(1) &= R_{4x} \end{aligned} \tag{4.6}$$

P_{1x}, P_{4x} and R_{1x}, R_{4x} are endpoints and tangent vector endpoints, respectively. Index points 1 and 4 are used instead of 1 and 2 for consistency with the notation used for Bezier curves. Substituting these conditions into Equations (4.2) and (4.5)

$$\begin{aligned} x(0) &= P_{1x} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} C_x \\ x(1) &= P_{4x} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} C_x \\ x'(0) &= R_{1x} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} C_x \\ x'(1) &= R_{4x} = \begin{bmatrix} 3 & 2 & 1 & 0 \end{bmatrix} C_x \end{aligned} \tag{4.7}$$

or

$$\begin{Bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} C_x \tag{4.8}$$

Solving for C_x by inverting the 4x4 matrix in Equation (4.8) yields

$$C_x = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{Bmatrix} \tag{4.9}$$

or

$$C_x = M_h \, G_{hx}$$

where M_h is the Hermite matrix and G_{hx} is the Hermite geometry vector. Substituting for C_x in Equation (4.3) gives

$$x(t) = T \, M_h \, G_{hx} \tag{4.10}$$

Similarly

$$y(t) = T\ M_h\ G_{hy} \tag{4.11}$$

$$z(t) = T\ M_h\ G_{hz} \tag{4.12}$$

As discussed previously, Bezier curve endpoint tangent vectors are defined by two points. These points are defined by the following relationships.

$$R_1 = 3(P_2 - P_1) = P'(0) \tag{4.13}$$

$$R_4 = 3(P_4 - P_3) = P'(1) \tag{4.14}$$

The relationship between the Hermite geometry matrix G_h and the Bezier geometry matrix G_b is

$$G_h = \begin{Bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{Bmatrix} = M_{hb}\ G_b \tag{4.15}$$

Substituting for G_h in Equation (4.10) gives

$$x(t) = T\ M_h\ M_{hb}\ G_{bx} \tag{4.16}$$

If we let

$$M_b = M_h\ M_{hb} \tag{4.17}$$

then

$$x(t) = T\ M_b\ G_{bx} \tag{4.18}$$

M_b is called the Bezier matrix. Numerically it is

$$M_b = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.19}$$

4.2 Three Dimensional Surfaces

To move from the cubic curve to the bicubic surface an additional variable s is used. Replacing t with s in Equation (4.18) we obtain

$$x(s) = S\ M_b\ G_{bx} \tag{4.20}$$

Instead of writing the Bezier geometry matrix as a constant, it can be written as a function of t.

$$x(s,t) = S\ M_b\ G_{bx}(t) = S\ M_b \begin{Bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{Bmatrix} \tag{4.21}$$

Now let $P_1(t)$, $P_2(t)$, $P_3(t)$, $P_4(t)$ each be cubes in Bezier form, so that

$$P_{1x}(t) = T\ M_b \begin{Bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \end{Bmatrix} \tag{4.22}$$

$$P_{2x}(t) = T\ M_b \begin{Bmatrix} p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \end{Bmatrix} \tag{4.23}$$

$$P_{3x}(t) = T\ M_b \begin{Bmatrix} p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{Bmatrix} \tag{4.24}$$

$$P_{4x}(t) = T\ M_b \begin{Bmatrix} p_{41} \\ p_{42} \\ p_{43} \\ p_{44} \end{Bmatrix} \tag{4.25}$$

As a row vector

$$P_1(t)\ P_2(t)\ P_3(t)\ P_4(t) =$$

$$T\ M_b \begin{bmatrix} p_{11} & p_{21} & p_{31} & p_{41} \\ p_{12} & p_{22} & p_{32} & p_{42} \\ p_{13} & p_{23} & p_{33} & p_{43} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{bmatrix} \tag{4.26}$$

Transposing both sides of the equation with the identity $(ABC)^T = C^T\ B^T\ A^T$

$$\begin{Bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{Bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} M_b^T\ T^T = P_x\ M_b^T\ T^T \tag{4.27}$$

Substituting into Equation (4.21)

$$x(s,t) = S\ M_b\ P_x\ M_b^T\ T^T \tag{4.28}$$

Similarily

$$y(s,t) = S\ M_b\ P_y\ M_b^T\ T^T \tag{4.29}$$

$$z(s,t) = S\ M_b\ P_z\ M_b^T\ T^T \tag{4.30}$$

The geometry matrix P consists of sixteen control points as shown in Figure (4.4).

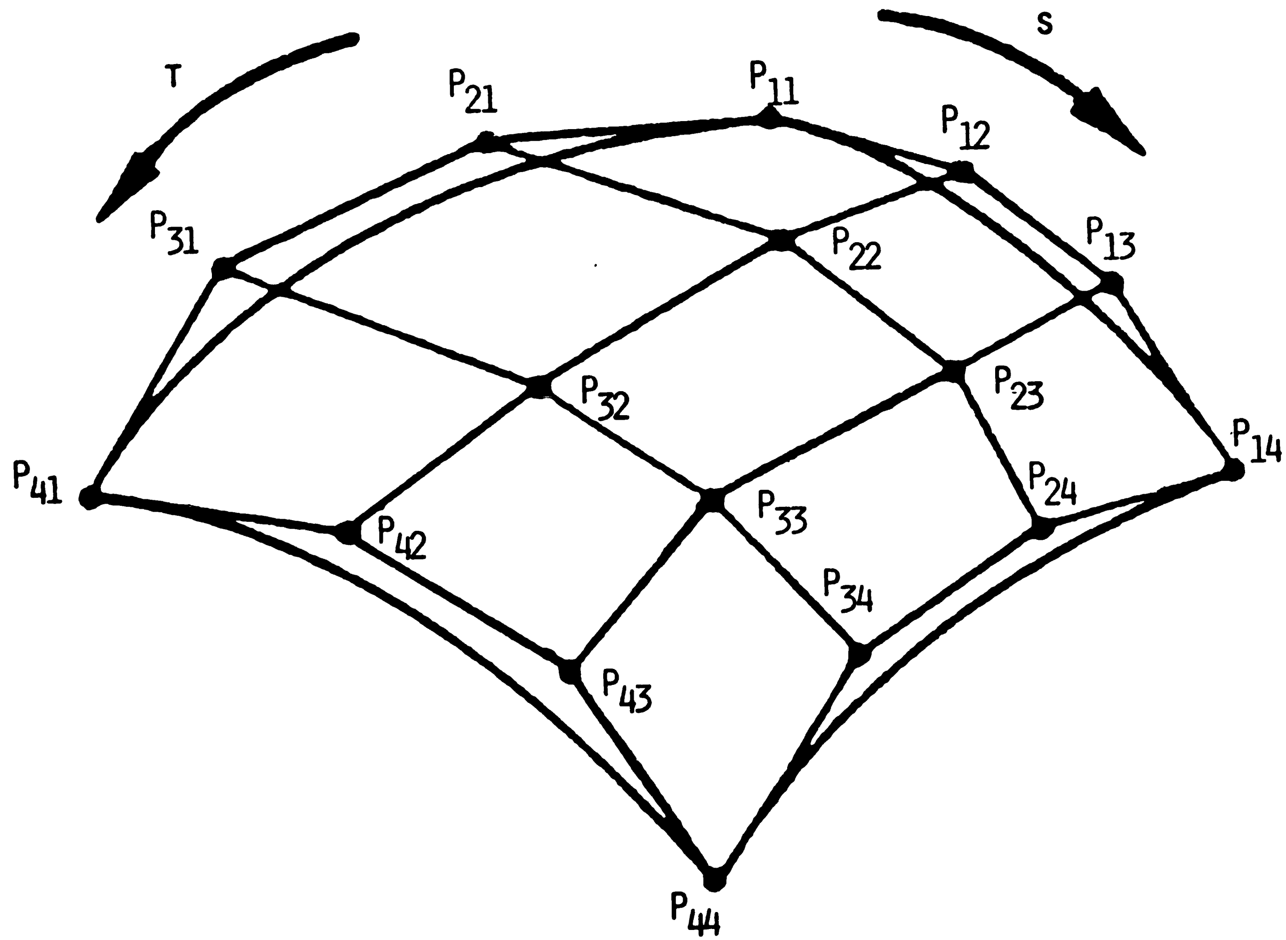

Figure 4.4. Sixteen Control Points for Bezier Bicubic Patch.

Chapter 5

COMPUTER MODEL

The moldboard plow model and graphics representation were implemented on a Prime 750 computer equipped with a Fortran V compiler, a Tektronix Plot-10 Terminal Control System (TCS) and a Tektronix 4000 Series terminal. Also used in this study was three dimensional plotting software written by personnel of The Case Center for Computer Aided Design, Michigan State University.

The moldboard plow was the primary application of this program so plow terminology is used. However, the reader should be aware that the program is also applicable to other fixed surface tillage tools. These applications are discussed in Chapter 6.

5.1 Moldboard Plow Model

Two Bezier surfaces are used to graphically depict a moldboard plow. One for the share and one for the moldboard. Each surface is divided into planar elements with four corner points describing their bounds. Each element has the forces computed in the x, y and z directions as if each were an independent plane. The sum of forces in each direction over the entire surface is the force required to propel the plow through the soil. Moments created by the forces on each element are also summed about a user specified point. This point may correspond to the mounting location of the plow to the implement frame.

The following geometric assumptions were used to implement the moldboard plow model:

1. The point of the share has (x,y,z) coordinates of (0,0,0).

2. The plow moves in the negative x-direction. Therefore, the control points used to describe the surface shapes must also describe the orientation of the plow as it moves through the soil.

3. Each surface is divided into one to one hundred elements; up to ten divisions in each of the s and t directions. See Figure (5.1) for the definition of the s and t directions.

4. The same number of elements in the s and t directions are used. However, the share and moldboard can have a different number of elements.

5. All four edges of the share are straight. Therefore, only the corner point coordinates are used to describe the surface.

6. The edge of the moldboard adjacent to the share is straight.

7. Elements in the curved portion of the surface do not have all four corner points in a plane, since only three points describe a plane. Therefore, the point farthest from the plow point on each element is not used to compute the plane orientation.

8. The four center surface control points are not user specified. They are computed from the specified edge points such that the surface has twist vectors of zero. This eliminates local surface irregularities.

The soil path over a share and moldboard is very important in determining the plow forces. This study assumes the soil path is a function of force exerted on soil by the element geometrically preceding it on the Bezier surface. For example, in Figure (5.1) the soil path over element two is dependent on the forces applied to the soil by element one. The soil path over element three is dependent on the forces applied to the soil by element two. The soil path over element four is dependent on the forces applied to the soil by element three, and so on.

The soil path dependency is determined by the y and z force direction cosines and the geometric orientation of an element (that is, the soil must be in contact with the element).

$$D_y = \frac{F_y}{\sqrt{(F_x^2 + F_y^2 + F_z^2)}} \tag{5.1}$$

$$D_z = \frac{F_z}{\sqrt{(F_x^2 + F_y^2 + F_z^2)}} \tag{5.2}$$

$$D_x = \frac{D_y}{\tan \alpha} + D_z \tan \delta \tag{5.3}$$

where

D = the direction cosine

F = force applied to the soil by the previous element

α = as defined in Chapter 3

δ = as defined in Chapter 3

The x direction cosine is not used because this adds a fourth constraint to a three dimensional system. Relaxing

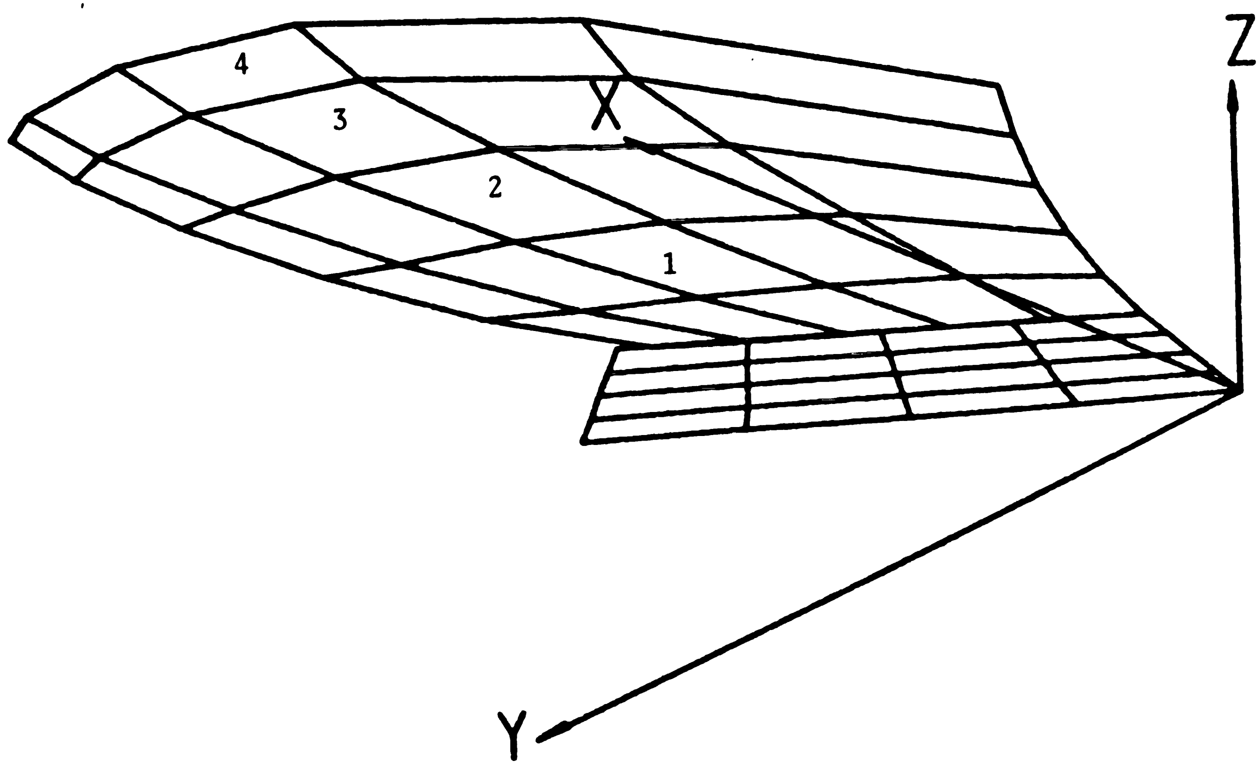

Figure 5.1. Location of Four Elements on Moldboard.

the x direction cosine constraint for a moldboard plow is reasonable since the absolute soil movement is up and to the right (z and y directions) with very little absolute soil movement in the direction of plow travel (x direction).

As stated previously, the soil path is assumed to be a function of the force exerted on the soil by the previous element. Before the soil moves on to the first row of share elements, however, there has not been a force applied to the soil. The path of the soil over the first row of elements was assumed to be parallel to the direction of tool travel (there is no soil movement in the y direction).

5.2 Computer Program

An interactive computer program named PLOW was developed to prove that computer graphic techniques (i.e., Bezier surfaces) could be used to represent tillage tools (primarily moldboard plow surfaces) and then be combined with a soil-tool mechanics to calculate the forces exerted on the tool by the soil. Several 'utilities' were written to realize the implementation. To assure the user that the plow is accurately represented, a picture of the plow is plotted on the terminal using drawing functions. There are also editing functions to change the data files. Lastly, there is a function to calculate the forces on a tillage tool.

Figure (5.2) is the menu displayed by PLOW. Procedures 1 thru 9 are drawing functions. Procedures 10 thru 13 are editing functions. Procedure 14 is the force calculation

```
    1)  Plot
    2)  Change point of view
    3)  Change look at point
    4)  Change magnification factor
    5)  Change center position
    6)  Set plotting conditions to default
    7)  Label control points on plot
    8)  Plot measured points
    9)  Change number of surface elements
   10)  Change operating conditions
   11)  Change share, moldboard or scaling
   12)  Get new tool geometry data
   13)  Get new operating data
   14)  Compute forces
   15)  End
Enter number of desired procedure (99 for menu).
```

Figure 5.2. PLOW Menu.

Tool totals	X	Y	Z
Forces (N)			
Share	23574.	-11936.	-18053.
Moldboard	3988.	-1349.	-951.
Total	27562.	-13285.	-19004.
Moments (N-M)			
Share	-9066.	12647.	-20200.
Moldboard	253.	2613.	-4158.
Total	-8813.	15260.	-24358.
Location of zero moment	0.00	0.00	0.00

Figure 5.3. Example of PLOW Force Output.

function and procedure 15 is the program termination function.

PLOW is setup to run with or without a graphics terminal. The graphics are Tektronix 4000 series terminal compatable. Procedure 1 is for plotting the tillage tool any time the user is prompted for a procedure. Procedures 2 thru 6 are purely for drawing pictures. Procedure 7 is for plotting the control points discussed in Chapter 4. Procedure 8 is for plotting any points the user wishes. Procedure 9 is used to change the number of elements on the tillage tool. This is also considered an editing function for force calculations. The drawing functions are not usable without a graphics terminal.

The editing functions are for changing the information read from two data files. This information is the control points of the tool geometry and operating conditions. As stated above, procedure 9 is for changing the number of elements on the tillage tool. Procedure 10 is for changing the soil and tool operating conditions. Procedure 11 is for changing the control points of the share or moldboard. Procedure 11 is also used to change the overall geometric scaling factors. Procedure 12 is for reading a different tool geometry file and procedure 13 is for reading a different operating conditions file.

Procedure 14 computes the forces on the tillage tool. Force calculations are made on each element and then summed over all the elements of the tool. In addition, the user is prompted for a point that has a zero moment. The distance from each element centroid to the zero moment point is calculated and multiplied by each of the component forces to give a torque. The element torques are summed to give tool torques about each of the component axes. The zero moment point might correspond to the mounting location of the tool to the implement frame. Figure (5.3) shows a typical output.

Finally, procedure 15 ends the program gracefully. The users guide for PLOW installed on the Prime in the Case Center at Michigan State University is in Appendix A.

CHAPTER 6

RESULTS AND VERIFICATION

Verification of the computer program is accomplished by comparing the predicted forces with actual force measurements. The measured force data was extracted from publications of authors with the equipment and resources for the measurements. Two types of tools are considered for comparisons. First, inclined blades of varying widths in four soil conditions are discussed and second, a moldboard plow in one soil type is discussed.

6.1 Inclined Blade

Plasse, Raghavan and McKyes (1983) reported that F. L. Desir measured the draft force for narrow tines in varying soil conditions. In the context of this discussion narrow tines are inclined blades. Two soil types were used; clay and sandy clay, each at two moisture contents. Three blade widths were used; 63, 125 and 200 millimeters. The operating depth and lift angle were held constant at 250 millimeters and 35 degrees, respectively.

Although the computer program was written for modeling a conventional moldboard plow, the inclined blade can be considered geometrically a very simple moldboard without lateral or vertical curvature. For further simplification, when a blade is considered rectangular, in a single plane, and the leading edge is perpendicular to the direction of travel; the soil will not apply a lateral force to the blade. For this tool orientation the model is two dimensional. The same results would be achieved with the two dimensional model developed by Soehne discussed in Chapter 3.

The description of the inclined blade for computer representation was very simple. Like the moldboard plow, the blade was represented as two surfaces; the share and moldboard. However, unlike the moldboard, the surfaces were in the same plane with straight edges. The control points on the moldboard edges were specified to be on the straight lines connecting the corner points. By changing only the y-direction scaling factor the three blades were geometrically represented (Figures (6.1a, b and c)).

When the draft force of each blade was measured, the operating conditions were also measured. These parameters were soil cohesion, soil internal friction, soil bulk density, soil-metal friction and soil-metal adhesion. Table (6.1) shows the parameters and draft forces as they were measured. The vertical and lateral force measurements were not reported.

The right most column of Table (6.1) is the draft force predicted by the model. Comparison of the predicted forces to the measured forces (second from right of Table (6.1)) shows wide variation. This can be explained by considering

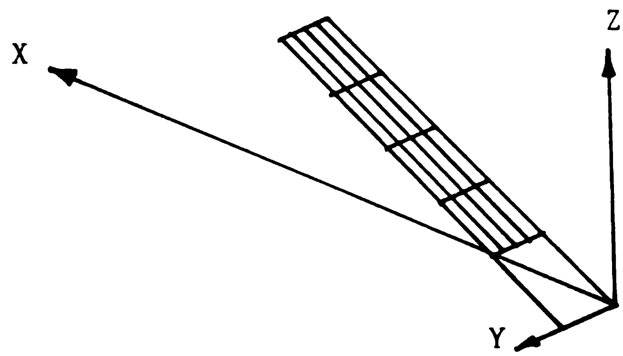

Figure 6.1a. 63 mm Wide Inclined Blade.

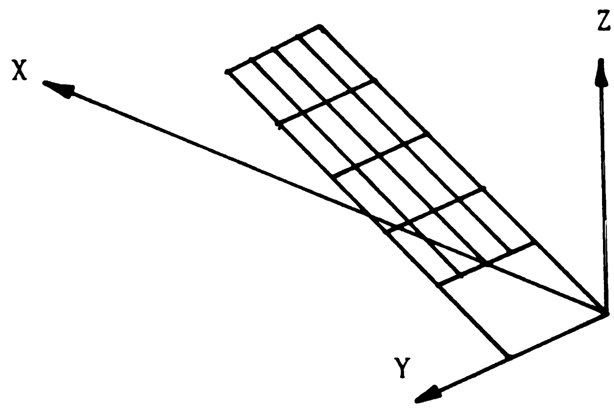

Figure 6.1b. 124 mm Wide Inclined Blade.

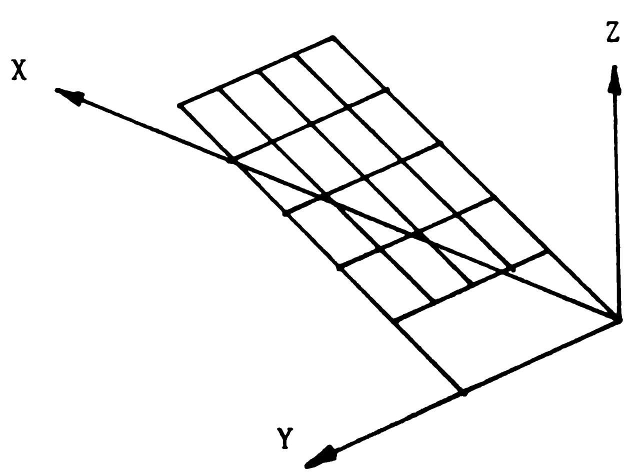

Figure 6.1c. 200 mm Wide Inclined Blade.

TABLE 6.1 Inclined blade operating conditions, measured forces and predicted forces.

Case #	Blade Width (cm)	Cohesion (KPa)	Soil Internal Friction Angle (deg)	Soil Bulk Density (Kg/cu m)	Soil-Metal Friction	Soil-Metal Adhesion (KPa)	Measured Draft Force (KN)	Predicted Draft Force (KN)
1	63	35.9	38	1034	.445	1.8	6.13	1.225
2	125	17.2	45	1007	.625	12.1	8.00	1.718
3	200	48.3	35	1089	.364	11.0	10.70	4.533
4	63	15.5	39	1025	.625	12.1	5.23	.733
5	125	55.2	26	980	.466	6.9	6.10	3.125
6	200	37.9	24	1043	.364	10.5	8.63	3.212
7	63	6.9	36	1379	.445	2.8	1.57	.416
8	125	17.2	34	1361	.306	5.4	2.43	1.201
9	200	2.0	34	1415	.404	3.5	1.67	.834
10	63	10.3	33	1406	.364	4.6	2.70	.461
11	125	10.3	32	1288	.364	9.0	3.13	.871
12	200	17.2	40	1325	.510	3.9	3.77	2.497

the logic of the model.

Since the model was originally intended to predict the forces exterted on a moldboard plow, those forces associated with forming the shear planes, lifting and turning the soil, and resistances to movement in the soil and at the soil-tool interface were included. Three forces were excluded to simplify the calculations.

The first is pure cutting resistance. Soehne determined in his research (Gill and VandenBerg, 1967), that the cutting resistance is only important when the tool is dull or when there are stones or organic matter present. Since these conditions were not present, excluding the pure cutting resistance is appropriate for the inclined blade also.

The other two forces excluded are due to the interaction between the soil moving on the blade and the stationary soil at each side. These forces are soil cohesion and soil internal friction. A moldboard plow does not have either of these forces on the dead furrow side. The soil internal frictional force does not apply to the land side since the soil's first movement is away from the land. The soil cohesion force in the land side shear planes was considered small compared to the cohesion in the forward shear planes and the acceleration forces required to lift and turn the furrow.

The latter two forces discussed may have a significant effect on the total draft force of the inclined blade.

There is stationary soil on each side of the moving soil, adding a soil internal frictional force. Also, as the width of the blade is reduced, the area difference between the cohesion forward and side shear surfaces becomes small, giving the side cohesion a relative larger contribution. In general, the relative contribution of smaller forces becomes more significant when a planar tool is used because the acceleration force, which is the largest force, becomes much smaller.

6.2 Moldboard Plow

Emmet (1983) measured the forces on a moldboard plow at the National Tillage Machinery Laboratory in Auburn, Alabama. The test plow was a 16 inch Ford general purpose bottom with a disposable-type share. The soil and operating conditions are shown in Table (6.2). The soil-metal adhesion is zero because an effective soil-metal friction was measured for the plow material that included the adhesion. Details of the test procedures and results were reported by Emmet (1983).

The shape of the Bezier surfaces in the model had to be manipulated to match the shape of the test plow. This was accomplished by 1) measuring the x, y and z coordinates of eight (8) descriptor points on the share and twenty nine (29) descriptor points on the moldboard, 2) plotting them over the Bezier surfaces on the computer terminal and 3) changing the surfaces to match the descriptor points. Figure (6.2) shows the descriptor points plotted on the

Table 6.2. Soil and Operating Conditions Used for Model Verification.

Soil Type	Norfolk Sandy Loam
Soil Bulk Density (kg/cu m)	1770
Soil Cohesion (kPa)	19.7
Soil Internal Friction Angle (deg)	34.8
Soil-Metal Friction Coefficient	0.4
Soil-Metal Adhesion (kPa)	0.0
Forward Velocity (m/s)	1.34
Tool Depth (m)	.203

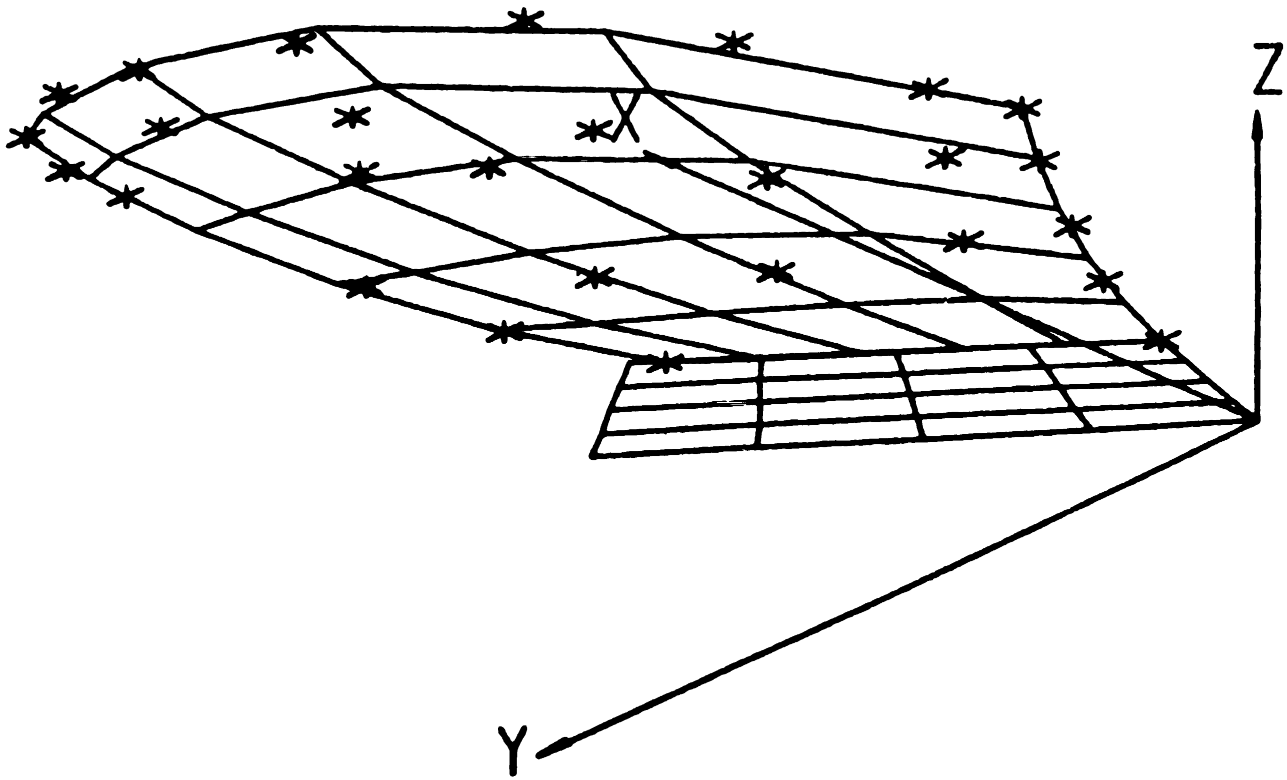

Figure 6.2. Sixteen Inch Ford Plow Descriptor Points.

surfaces.

The plow on which the descriptor points were measured had been used regularly for several years with little maintenance, consequently the plow was worn considerably. With this in mind, descriptor points on the share and moldboard were selected to best describe the plow shape. The plow was then mounted on the table of a Bridgeport vertical spindle mill with the moldboard face up. Each point was moved to a pointer chucked in spindle while the table travel was recorded. Because of the limited travel on the mill table, the plow had to be moved once to measure the whole plow. Datum points located before and after the move were used to fit the two sets of descriptor points together. The author estimated the accuracy of each measurement to be plus or minus two (2) millimeters.

Once the descriptor points were modified to be in the same coordinate system, they had to be manipulated so that the moldboard sat upright as it would be during operation. This required writing a computer program to translate and rotate the points by multiplying them by a transformation matrix. Discussion of this technique can be found in Foley and VanDam (1982) and Faux and Pratt (1979).

Matching the Bezier surfaces to the descriptor points was a tedious trial and error process. Four of the descriptor points were selected so that they fell on the corners of the share surface and four were selected so they fell on the corners of the moldboard surface. The control points were then manually manipulated to provide a visually estimated minimum error between the descriptor points and the Bezier surfaces. Additional interactive software could be developed to greatly simplify this task. Software could also be developed to mathematically quanitify and minimize the error.

The number of elements on each Bezier surface can be changed independent of the other surface. The greater the number of elements, the smaller the size of each element and the better the accuracy of geometric representation. However, when the elements get too small and the soil 'sees' a slightly curved surface section after a sharply curved section, the soil deceleration force is much greater than the soil gravitational force and the calculation of the surface normal load becomes negative. Since this is physically impossible, the normal load is set to zero. However, the path of the soil over the next row of elements is dependent on the forces on that row. Consequently, the user is notified that the soil-tool model can not calculate the forces. The one exception is when a zero normal load is encountered on the last row of elements before the soil leaves the plow.

To calculate the forces for a surface that has the negative normal load problem, the number of surface elements can be decreased until the element representation is course enough to 'hide' the slightly curved surface section after the sharply curved section. The maximum number of elements for the Ford moldboard plow is 36 (that is, six along each edge). Note that this model assumes an equal number of elements along each edge for an individual surface.

As implied previously, when elements too large in size are used, large geometric inaccuracies occur. Also, the soil-tool mechanics eliminates forces with the assumption that the elements are reduced in size. Therefore, the share and moldboard surfaces with one or four elements were not considered.

Figure (6.3) is a graphical representation of the share with 16 elements and the moldboard with 25 elements. Force results were generated for 9, 16, 25 and 36 element share and moldboard combinations. The average of the forces in each of the component directions is shown in Table (6.3). For the x, y and z directions all the forces were within 3.5, 5.1 and 2.4 percent of the averages, respectively.

Table (6.3) also shows the comparisons between the predicted forces and the measured forces. The predicted draft force (F_x) is 15 percent larger than the measured force; the predicted side force (F_y) is 53 percent larger than that measured; and the predicted vertical force (F_z) is 115 percent larger than that measured.

The predictions could be improved by:

1. measuring the descriptor points more accurately.
2. fitting the surfaces to the descriptor points less subjectively.

3. calculating the geometrical properties of each element more accurately.

4. refining the soil-tool mechanics assumptions.

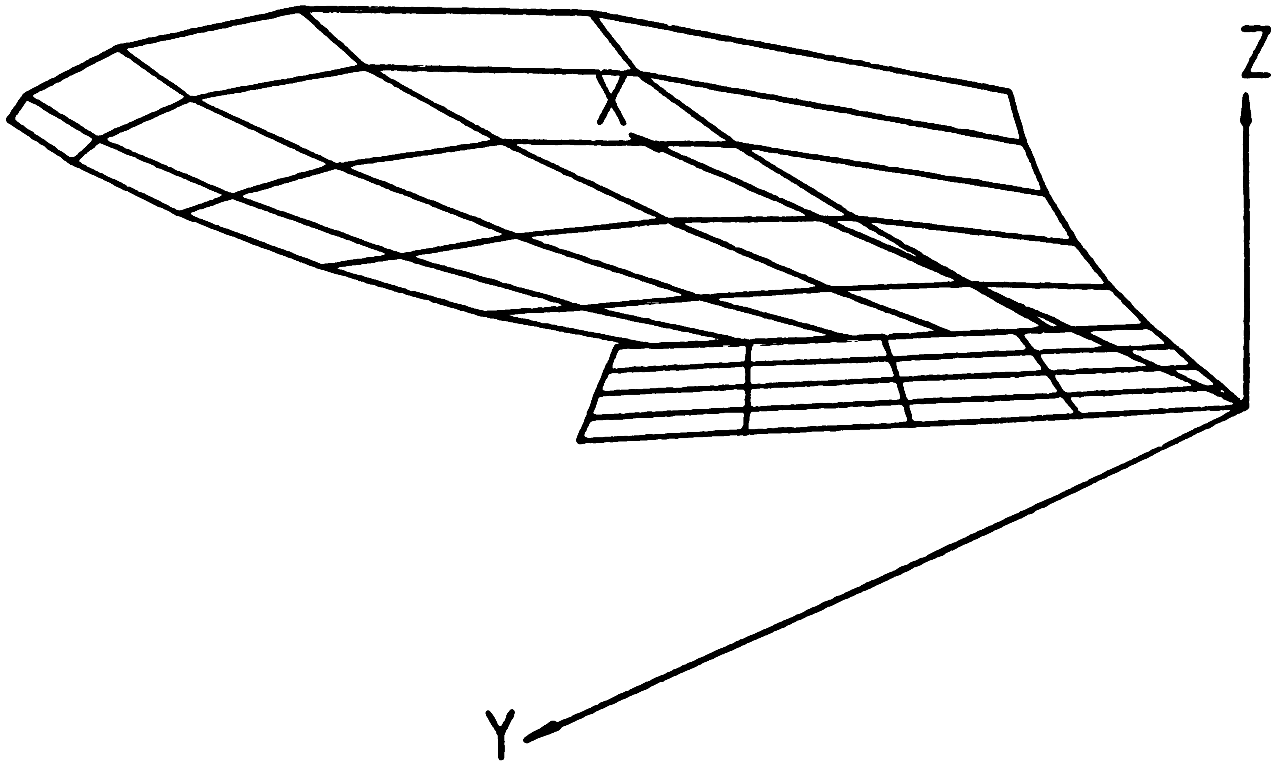

Figure 6.3. Representation of a Moldboard Plow Bottom by Multiple Three Dimensional Inclined Planes.

Table 6.3. Comparison Between the Predicted Plow Forces and the Measured Forces.

Mode	Draft (n) (F_x)	Side (n) (F_y)	Vertical (n) (F_z)
Predicted	3895	-1712	-3319
Measured	3389	-1118	-1542

Chapter 7

CONCLUSIONS

1. Three dimensional graphics techniques (Bezier surfaces) can be used to describe share and moldboard shapes.

2. The user should be aware of the assumptions of the model when applying it to tool geometries other than a moldboard plow. The computer model does not accurately predict draft forces for inclined blades.

3. Three dimensional force reactions can be calculated without prior knowledge of the soil path over the moldboard plow surface. Moldboard plow draft force predictions were within 15 percent of the draft force measurements.

4. Force calculations for other moldboard plow shapes and operating conditions should be checked against measured forces to better determine the validity of the model.

Chapter 8

SUGGESTIONS FOR FURTHER RESEARCH

This research illustrates only one application of computer graphic techniques to modeling. Many others are possible. These techniques are good for any application where complicated and universal surface descriptions are required. The soil tillage tool area has several more applications. Modeling of shallow and deep narrow tines, wide sweep or shovel cultivator tines and disk type tools are all possible.

The computer program PLOW could be used for some of the above applications. PLOW was written in modules to facilitate replacing, adding or deleting functions as required for a particular model. For example, a more accurate inclined blade model could be developed by replacing the force calculation subroutines with those more closely representing that tool.

Another possible application of PLOW is using the output as input to another program for designing an implement hitching system. The three component dirction forces and the three corresponding moments located along a beam or tool bar can be summed to size the implement structural members and hitch orientation to optimize available pulling power.

A very computer time intensive application would be optimization of a tillage tool surface geometry to minimize draft while maximizing the desired soil manipulation. There would be a considerable amount of research required to develop the optimization algorithms, but once in place only computer time is needed to design a new implement.

There are many Computer Aided Design (CAD) software packages available that use very elegant graphics manipulations as part of their drawing functions. If these were married with a force calculation function, the extent of design using computers would be greatly expanded.

APPENDIX A

PLOW USERS GUIDE

PLOW USERS GUIDE

This document describes the operation of the computer program PLOW running on a Prime Computer System. PLOW is comprised of sixteen (16) procedures. This document outlines the steps to execute the program and each procedure within it.

The following conventions are used in this document:

1. Text or numbers enclosed in quotation marks " " represents information displayed by PLOW.

2. Text or numbers enclosed in left and right carrots < > represents information entered from the keyboard by the user. Where more than one response is possible, each response will be followed by an explaination of that action.

3. Text or numbers not falling in areas 1 or 2 are explainations.

To begin execution of the program enter:

<SEG #PLOW> SEG is the system command to execute a run image. #PLOW is the run image. The rest of PLOW is interactive with the user.

"Is a graphics terminal being used? (y or n)"

<y> is the normal response.

<n> will stop PLOW from sending graphics information to the terminal. This severely limits the usefulness of PLOW.

"Enter baud rate."

<1200> can be any of the standard baud rates. This information is needed to initialize the three dimensional graphics drivers.

"Enter file name for writing tool force information."

<D_FORCES> is any file name. The force calculations will be stored here for later printing or review.

"Enter name of geometry data file."

<D_FORD> is file name of geometry previously created by PLOW.

"Enter name of operating conditions file."

<D_LMCLAY1> is file name of operating conditions file previously created by PLOW.

The following main menu is displayed. The remainder of PLOW will work from this menu.

```
"1)  Plot
 2)  Change point of view
 3)  Change look at point
 4)  Change magnification factor
 5)  Change center position
 6)  Set plotting conditions to default
 7)  Label control points on plot
 8)  Plot measured points
 9)  Change number of surface elements
10)  Change operating conditions
11)  Change share, moldboard and scaling
12)  Get new tool geometry data
13)  Get new operating data
14)  Compute forces
15)  End
Enter number of desired procedure. (99 for menu)"
```

<1> plots the active tool. Return to main menu prompt.

<2> "Current view point is (-1.0,1.0,1.0). Enter new X,Y,Z coordinates."

<1.0,.5,.2> The numbers identify the point coordinates you are looking from. Return to main menu prompt.

<3> "Current look at point is (0.0,0.0,0.0). Enter new X,Y,Z coordinates."

<1.,0.,0.> The numbers identify the point coordinates you are looking at. Return to main menu prompt.

<4> "Current magnification factor is 1.0. Enter new factor."

<2.0> The screen is magnified about the screen center by the value entered. Return to main menu prompt.

<5> "Current center point is (0.0,0.0). Enter new X,Y coordinates."

<-1.,.5> The three dimensional image is projected onto a two dimensional plane, so the value of the center point is relative to the size and shape of the tool. The user will have to 'get a feel' for this procedure with each tool. Return to main menu prompt.

<6> "Plotting conditions set to default."

While the user is trying to get a better view of a tool, he may modify several of the display values. This procedure provides an easy way to get back to the starting plot. Return to main menu prompt.

<7> This procedure labels the control points on the tool. Return to main menu prompt.

<8> "Enter measured points data file name."

<D_FORDPLOW> This file has data points that were measured and entered into a file. The file has four values per line. The first value is an integer identifying the point number and the next three are the X,Y,Z coordinates. There can be up to one hundred points in the file. Return to main menu prompt.

<9> "Enter number of elements along the share edge."

<2> This number can range from 1 to 10.

"Enter number of elements along the moldboard edge."

<5> This number can also range from 1 to 10. As with the share, there is an equal number of elements along each surface edge. Return to main menu prompt.

<10> "
```
              Soil Parameters
     1) Soil type:                Norfolk Sandy Loam
     2) Bulk Density (kg/cu m)           1770.
     3) Cohesion (kPa)                   19.7
     4) Internal Friction Angle (deg)    34.8
     5) Soil-Metal Friction Coefficient  0.40
     6) Soil-Metal Adhesion (kPa)        0.0
              Tool Parameters
     7) Forward Velocity (m/s)           1.34
     8) Tool Depth (m)                   0.203
```
Enter parameter number to be changed. (0 to stop)"

<2> "Enter new value for parameter number 2."

<1690.>

"Enter parameter number to be changed. (0 to stop)"

<0> "Store modified data? (y or n)"

<y> "Enter new file name."

<D_NEXT_SOIL> Return to main menu prompt.

<n> Return to main menu prompt.

<11> "Do you want to change:
1) share
2) moldboard
3) scaling
4) stop"

<1> Four share control points will be displayed.

"Enter point # to be edited. (0 to stop)"

<1> "Enter X,Y,Z."

<0.,0.,0.>

<0> Return to prompt at 11.

<2> Twelve moldboard control points will be displayed.

"Enter point # to be edited. (0 to stop)"

<1> "Enter X,Y,Z."

<0.,0.,0.>

<0> Return to prompt at 11.

<3> "X-scale = 1.
Y-scale = 1.
Z-scale = 1.
Enter new X-scale, Y-scale, Z-scale."

<.5,.5,.5> This changes the size of the tool in each of the component directions by the factor entered. Return to prompt at 11.

<4> "Do you want to store modified data in file? (y or n)."

<y> "Enter name of tool geometry data file to be written to."

<D_NEXT_GEOMETRY> Return to main menu prompt.

<n> Return to main menu prompt.

<12> "Enter name of geometry data file to be read."

<D_NEXT_GEOMETRY> Return to main menu prompt.

<13> "Enter name of operating data file to be read."

<D_NEXT_OPERATING> Return to main menu prompt.

<14> "Tool Totals

	X	Y	Z
Forces (N)			
Share	23574.	-11936.	-18053.
Moldboard	3988.	-1349.	-951.
Total	27562.	-13285.	-19004.
Moments (N-m)			
Share	-9066.	12647.	-20200.
Moldboard	253.	2613.	-4158.
Total	-8813.	15260.	-24358.
Location of zero moment	0.	0.	0."

The numbers in the displayed table were just calculated. The time taken to calculate depends on the number of elements on the share and moldboard and the load on the system. These calculations can take several minutes. Return to main menu prompt.

<15> "End PLOW program."

APPENDIX B

PLOW SOURCE CODE

```
C
C                                                                    PLOW
C
C       This program interacts with the user to design a moldboard.
C         It was written to run on the Prime 750 computer at The
C         Case Center for Computer Aided Design, Michigan State
C         University.  The computer was equipped with a Fortran-77
C         compiler and a Tektronix Plot-10 Terminal Control
C         System (TCS).  In addition, three dimensional plotting
C         software developed by Case Center personnel was used.
C         This program has been run with a Tektronix 4100 series, a
C         Tektronix 4000 series, a DEC VT100 with graphics expansion
C         card, and DEC VT240 terminals.
C       Every attempt was made to adhere to Fortran-77 conventions.
C         There are no guarantees expressed or implied.
C
C*************************************************************************
C
C                         Written by:
C
C                      Stephen B. Richey
C
C                        January 1983
C
C*************************************************************************
C
C      Array definitions:
C
C          PSS - coordinates of points blended in s direction describing
C                  share surface.
C          PST - coordinates of points blended in t direction describing
C                  share surface.
C          PMS - coordinates of points blended in s direction describing
C                  moldboard surface.
C          PMT - coordinates of points blended in t direction describing
C                  moldboard surface.
C
C          XS,YX,ZS - coordinates of share control points.
C          XM,YM,ZM - coordinates of moldboard control points.
C
C          DA - coordinates of measured points to be plotted.
C
C********************
C
C      Variable definitions:
C
C          SOILTY - name of soil type (character string).
C          GAMMA - soil bulk density (kg/cu m).
C          COHES - soil cohesion (kpa).
C          PHI - soil internal friction (deg).
C          MUP - soil-metal friction.
C          ADHES - soil-metal adhesion (kpa).
C          VELTL - tool velocity (m/s).
C          DEPTHF - plowing depth (m).
```

```
C
C         SIZEX,SIZEY,SIZEZ - scaling factors for controls points.
C         VX,VY,VZ - view point coordinates.
C         AX,AY,YZ - look at point coordinates.
C         MAG - magnification factor.
C         CX,CY - center position coordinates.
C
C         OLM,JJM - number of lines drawn on moldboard.
C         OLS,JJS - number of lines drawn on share.
C
C         FOFILE - name of data file for writing force info
C                    (character string).
C         DATA - name of data file for reading measured data
C                    (character string).
C
C         LUNTI - logical unit number for terminal input.
C         LUNTO - logical unit number for terminal output.
C         LUNDA,LUNDB,LUNDC - logical unit number for data input/output.
C
C         GRAPH - answer to question of graphics terminal
C                   (character string).
C         Q - answer to question (character string).
C         Q1 - numerical answer to question.
C
C         BAUD - baud rate.
C
C     Subroutines called:
C
C         GETDAT - reads tool geometry data.
C         STODAT - stores tool geometry data.
C         OPERAT - reads soil and operating conditions.
C         DEFPLT - sets plotting parameters to default.
C         EDGE - calculates control points for a straight edge.
C         SURFACE - calculates point coordinates to be plotted.
C         SETPLT - sets up plot.
C         PLOT - plots surfaces.
C         LABPTS - labels control points on moldboard.
C         PARAM - edits soil and tool parameters.
C         NEWPS - edits share control points.
C         NEWPM - edits moldboard control points.
C         NEWSCA - edits scaling factors.
C         COMFOR - computes forces.
C
C     The following subroutines are on the Prime system at The
C       Case Center at Michigan State University.
C
C         INITT3 - initializes graphics file.
C         CLOSTK - closes graphics file.
C         MOVER3 - relative move (3-D).
C         MOVEA3 - absolute move (3-D).
C         DRAWR3 - relative draw (3-D).
C         RECOVR - recover to graphics mode.
C         HOME - move curser to screen upper left.
C         NEWPAG - page screen.
```

```
C          CARTN - generates carriage return.
C
C          CARTVP - sets cartisan view point.
C          LOOKAT - set lookat point.
C          PERSPT - turns on or off perspective viewing.
C          XUP,YUP,ZUP - defines which axis is up on plot.
C          PPCENT - sets distance to 2-D viewing plane from
C                     look at point to view point as a
C                     percentage.
C          DWINDO - sets bounds of area displayed on screen.
C          CENTER - sets center point of display window.
C          MAGNFY - sets display magnification factor.
C          MOVEA3 - move to specified coordinates (3-D).
C          VECTOR - draw line to specified coordinates and
C                     draw an arrowhead (3-D).
C          CHARTK - write text on plot.
C
C*********************
C
      IMPLICIT REAL (A-H,O-Z)
      IMPLICIT INTEGER (I-N)
      COMMON /A/GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      COMMON /B/VX,VY,VZ,AX,AY,AZ,MAG,CX,CY
      COMMON /LUN/LUNTI,LUNTO
      COMMON /LUND/LUNDA,LUNDB
C
      DIMENSION PMS(11,3,11),PMT(11,3,11),PST(11,3,11),PSS(11,3,11),
     +XM(4,4),YM(4,4),ZM(4,4),XS(4,4),YS(4,4),ZS(4,4)
      DIMENSION DA(3,50)
C
      CHARACTER*1 Q,GRAPH
      CHARACTER SOILTY*40
      CHARACTER*80 FOFILE,DATA
      REAL VX,VY,VZ,AX,AY,AZ,MAG,CX,CY
      REAL GAMMA,COHES,PHI,MUP,VELTL,DEPTHF
      INTEGER OLM,OLS,BAUD,Q1
C
      DATA LUNTI,LUNTO/1,1/
      DATA LUNDA,LUNDB,LUNDC/12,11,13/
C
      WRITE (LUNTI,*) 'Is a grpahics terminal being used?(y/n).'
      READ (LUNTO,'(A3)') GRAPH
      IF (GRAPH.EQ.'Y'.OR. GRAPH.EQ.'y') CALL NEWPAG
      WRITE (LUNTI,*) 'Enter baud rate.'
      READ (LUNTO,*) BAUD
      BAUD=BAUD/10
C
C        Open file for writing tool info.
C
4     WRITE (LUNTI,*) ' Enter file name for writing tool force',
     +  ' information.'
      READ (LUNTO,'(A)') FOFILE
      OPEN (LUNDA,FILE=FOFILE,STATUS='NEW',ERR=4)
C
```

```
C       Get tool geometry data, soil and tool operating conditions
C         and set plotting conditions to default.
C
      CALL GETDAT (SIZEX,SIZEY,SIZEZ,XS,YS,ZS,XM,YM,ZM)
      CALL OPERAT (SOILTY)
      CALL DEFPLT(OLM,JJM,OLS,JJS)
C
C       Calculate points for share.
C
5     CALL EDGE (XS,1,1)
      CALL EDGE (XS,1,4)
      CALL EDGE (XS,2,1)
      CALL EDGE (XS,2,4)
      CALL EDGE (YS,1,1)
      CALL EDGE (YS,1,4)
      CALL EDGE (YS,2,1)
      CALL EDGE (YS,2,4)
      CALL EDGE (ZS,1,1)
      CALL EDGE (ZS,1,4)
      CALL EDGE (ZS,2,1)
      CALL EDGE (ZS,2,4)
      CALL SURFAC (OLS,JJS,PSS,PST,XS,YS,ZS)
C
C       Calculate points for moldboard.
C
      CALL EDGE (XM,1,1)
      CALL EDGE (YM,1,1)
      CALL EDGE (ZM,1,1)
      CALL SURFAC (OLM,JJM,PMS,PMT,XM,YM,ZM)
C
C       Write menu to lunti.
C
99    WRITE (LUNTI,*) '     1)  Plot'
      WRITE (LUNTI,*) '     2)  Change point of view.'
      WRITE (LUNTI,*) '     3)  Change look at point.'
      WRITE (LUNTI,*) '     4)  Change magnification factor.'
      WRITE (LUNTI,*) '     5)  Change center position.'
      WRITE (LUNTI,*) '     6)  Set plotting conditions to default.'
      WRITE (LUNTI,*) '     7)  Label control points on plot.'
      WRITE (LUNTI,*) '     8)  Plot measured points.'
      WRITE (LUNTI,*) '     9)  Change number of surface elements.'
      WRITE (LUNTI,*) '    10)  Change operating conditions.'
      WRITE (LUNTI,*) '    11)  Change share, moldboard, or scaling.'
      WRITE (LUNTI,*) '    12)  Get new tool geometry data.'
      WRITE (LUNTI,*) '    13)  Get new operating data.'
      WRITE (LUNTI,*) '    14)  Compute forces.'
      WRITE (LUNTI,*) '    15)  Store plot in file and end.'
C
10    IF (GRAPH.EQ.'Y'.OR. GRAPH.EQ.'y') CALL ANMODE
C
      WRITE (LUNTI,*) 'Enter number of desired procedure',
     +  '(99 for menu).'
      READ (LUNTO,*,ERR=10) Q1
C
```

```
      IF (Q1.EQ.99) GOTO 99
      GOTO (6,15,16,25,30,35,40,75,45,50,60,55,80,70,100),Q1
      GOTO 10
C
C       Set up and plot tool.
C
6     IF (GRAPH.EQ.'N'.OR.GRAPH.EQ.'n') GOTO 10
      CALL INITT3 (BAUD)
7     CALL SETPLT (SIZEX,SIZEY,SIZEZ)
      CALL PLOT (PSS,PST,OLS,JJS)
      CALL PLOT (PMS,PMT,OLM,JJM)
      CALL ANMODE
      CALL HOME
      GOTO 10
C
C       Change view point.
C
15    WRITE (LUNTI,'(A24,F4.1,2(A1,F4.1),A32)')
     +  ' Current view point is (',VX,',',VY,',',VZ,
     +  ').  Enter new X,Y,Z coordinates.'
      READ (LUNTO,*) VX,VY,VZ
      GOTO 10
C
C       Change lookat point.
C
16    WRITE (LUNTI,'(A26,F4.1,2(A1,F4.1),A32)')
     +  ' Current lookat point is (',AX,',',AY,',',AZ,
     +  ').  Enter new X,Y,Z coordinates.'
      READ (LUNTO,*) AX,AY,AZ
      GOTO 10
C
C       Change magnification factor.
C
25    WRITE (LUNTI,'(A33,F3.1,A20)')
     +  ' Current magnification factor is ',MAG,
     +  '.  Enter new factor.'
      READ (LUNTO,*) MAG
      GOTO 10
C
C       Change screen center point.
C
30    WRITE (LUNTI,'(A25,F4.1,A1,F4.1,A19)')
     +  'Current center point is (',CX,',',CY,
     +  ').  Enter new X,Y coordinates.'
      READ (LUNTO,*) CX,CY
      GOTO 10
C
C       Reset plotting conditions
C
35    WRITE (LUNTI,*) 'Plotting conditions set to default.'
      CALL DEFPLT (OLM,JJM,OLS,JJS)
      GOTO 5
C
C       Label moldboard surface control points.
```

```
C
40    IF (GRAPH.EQ.'N'.OR.GRAPH.EQ.'n') GOTO 10
      CALL RECOVR
      CALL LABPTS (XM,YM,ZM)
      CALL CARTN
      CALL ANMODE
      GOTO 10
C
C       Change number of surface elements.
C
45    WRITE (LUNTI,*) ' Enter number of elements along the',
     +  ' share edge.'
      READ (LUNTO,'(I2)',ERR=45) OLS
      IF (OLS.LT.1.OR.OLS.GT.10) GOTO 45
      OLS=OLS+1
      JJS=OLS
46    WRITE (LUNTI,*) ' Enter number of elements along the',
     +  ' moldboard edge.'
      READ (1,'(I2)',ERR=46) OLM
      IF (OLM.LT.1.OR.OLM.GT.10) GOTO 46
      OLM=OLM+1
      JJM=OLM
      IF (GRAPH.EQ.'Y'.OR.GRAPH.EQ.'y') GOTO 5
      GOTO 10
C
C       Change operating conditions.
C
50    CALL PARAM (SOILTY)
      IF (GRAPH.EQ.'Y'.OR.GRAPH.EQ.'y') CALL NEWPAG
      GOTO 10
C
C       Get new tool geometry data.
C
55    CALL GETDAT (SIZEX,SIZEY,SIZEZ,XS,YS,ZS,XM,YM,ZM)
      IF (GRAPH.EQ.'Y'.OR.GRAPH.EQ.'y') GOTO 5
      GOTO 10
C
C       Get new operating data.
C
80    CALL OPERAT (SOILTY)
      WRITE (LUNTI,*) ' New data read.'
      GOTO 10
C
C       Change share, moldboard, or scaling.
C
60    WRITE (LUNTI,*) ' Do you want to change:'
      WRITE (LUNTI,*) '     1) share'
      WRITE (LUNTI,*) '     2) moldboard'
      WRITE (LUNTI,*) '     3) scaling'
      WRITE (LUNTI,*) '4) stop'
      READ (LUNTO,*) Q1
      GOTO (61,62,63,64),Q1
      GOTO 60
61    CALL NEWPS (XS,YS,ZS)
```

```
      GOTO 60
62    CALL NEWPM (XM,YM,ZM)
      GOTO 60
63    CALL NEWSCA (XS,YS,ZS,XM,YM,ZM,SIZEX,SIZEY,SIZEZ)
      GOTO 60
64    WRITE (LUNTI,*) ' Do you want to store modified data',
     +  ' in file?(y/n)'
      READ(LUNTO,'(A3)') Q
      IF (Q.EQ.'Y'.OR.Q.EQ.'y') THEN
        CALL STODAT (SIZEX,SIZEY,SIZEZ,XS,YS,ZS,XM,YM,ZM)
      ENDIF
      IF (GRAPH.EQ.'Y'.OR.GRAPH.EQ.'y') GOTO 5
      GOTO 10
C
C       Calculate forces.
C
70    IF (GRAPH.EQ.'Y'.OR.GRAPH.EQ.'y') CALL NEWPAG
      CALL COMFOR (PMS,PSS,OLS,JJS,OLM,JJM,SOILTY)
      GOTO 10
C
C       Plot measured points.
C
75    IF (GRAPH.EQ.'N'.OR.GRAPH.EQ.'n') GOTO 10
      WRITE (LUNTI,*) ' Enter data file name.'
      READ (LUNTO,'(A10)',ERR=75) DATA
      OPEN (LUNDC,FILE=DATA,STATUS='OLD',ERR=75)
      NUMP=1
1     READ (LUNDC,*,END=2) J,(DA(I,NUMP),I=1,3)
      NUMP=NUMP+1
      GOTO 1
2     NUMP=NUMP-1
      CLOSE (LUNDC)
      CALL RECOVR
      CALL NEWPAG
      CALL SETPLT (SIZEX,SIZEY,SIZEZ)
C       Plot points.
      DO 3 I=1,NUMP
        CALL MOVEA3 (DA(1,I),DA(2,I),DA(3,I))
        CALL MOVER3 (-.05*SIZEX,0.,0.)
        CALL DRAWR3 (.1*SIZEX,0.,0.)
        CALL MOVER3 (-.05*SIZEX,-.05*SIZEY,0.)
        CALL DRAWR3 (0.,.1*SIZEY,0.)
        CALL MOVER3 (0.,-.05*SIZEY,-.05*SIZEZ)
        CALL DRAWR3 (0.,0.,.1*SIZEZ)
3     CONTINUE
C
      GOTO 7
C
C       End program.
C
100   CALL CLOSTK (IERR)
      CALL HOME
      CALL ANMODE
      WRITE (LUNTI,*) ' End PLOW program.'
```

```
C
      CLOSE (LUNDA)
      CLOSE (LUNTI)
      CLOSE (LUNTO)
C
      END
C
C                                                                   INIT
C
C********************
C
      SUBROUTINE DEFPLT (OLM,JJM,OLS,JJS)
C
C              This subroutine sets plotting parameters to default
C
C    Variable definitions:
C
C         VX,VY,VZ - view point coordinates.
C         AX,AY,AZ - look at point coordinates.
C         MAG - magnification factor.
C         CX,CY - center position coordinates.
C         OLM,OLS - number of horizontal lines drawn on moldboard
C                     and share respectively.
C         JJM,JJS - number of vertical lines drawn on moldboard
C                     and share respectively.
C
      COMMON /B/VX,VY,VZ,AX,AY,AZ,MAG,CX,CY
      REAL VX,VY,VZ,AX,AY,AZ,MAG,CX,CY
      INTEGER OLM,OLS
C
      VX=-1.
      VY=1.
      VZ=1.
      AX=0.
      AY=0.
      AZ=0.
      MAG=1.5
      CX=-1.
      CY=.5
      OLM=4
      JJM=4
      OLS=2
      JJS=2
      RETURN
      END
C
C********************
C
      SUBROUTINE OPERAT (SOILTY)
C
C           This subroutine sets soil and operating conditions to
C             values read from user specified file.
C
C     Variable definitions:
```

```
C
C            DFILE - name of operating conditions file
C                    (character string).
C
C            SOILTY - name of soil type (character string).
C            GAMMA - soil bulk density (kg/cu m).
C            COHES - soil cohesion (kpa).
C            PHI - soil internal friction (deg).
C            MUP - soil-metal friction.
C            ADHES - soil-metal adhesion (kpa).
C            VELTL - tool velocity (m/s).
C            DEPTHF - plowing depth (m).
C
C            LUNTO - logical unit number for terminal output.
C            LUNTI - logical unit number for terminal input.
C            LUNDI - logical unit number for data input.
C
      COMMON /A/GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      COMMON /LUN/LUNTI,LUNTO
      CHARACTER SOILTY*40
      CHARACTER DFILE*80
      REAL GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      DATA LUNDI/10/
C
10    WRITE (LUNTI,*) 'ENTER NAME OF OPERATING CONDITIONS FILE.'
      READ (LUNTO,'(A10)') DFILE
C
      OPEN (UNIT=LUNDI,FILE=DFILE,STATUS='OLD',ERR=10)
      READ (LUNDI,'(A20)') SOILTY
      READ (LUNDI,*) GAMMA
      READ (LUNDI,*) COHES
      READ (LUNDI,*) PHI
      READ (LUNDI,*) MUP
      READ (LUNDI,*) ADHES
      READ (LUNDI,*) VELTL
      READ (LUNDI,*) DEPTHF
C
      CLOSE(LUNDI)
      RETURN
      END
C
C********************
C
      SUBROUTINE GETDAT (SIZEX,SIZEY,SIZEZ,XS,YS,ZS,XM,YM,ZM)
C
C            This subroutine reads tool geometry data from user
C              specified file.
C
C      Array definitions:
C
C            XS,YX,ZS - coordinates of share descritive points.
C            XM,YM,ZM - coordinates of moldboard descriptive points.
C
C      Variable definitions:
```

```
C
C           DFILE - name of file to be read (character string).
C           SIZEX,SIZEY,SIZEZ - scaling factors for x,y,z data.
C
C           LUNTO - logical unit number for terminal output.
C           LUNTI - logical unit number for terminal input.
C           LUNDI - logical unit number for data input.
C
      COMMON /LUN/LUNTI,LUNTO
      DIMENSION XS(4,4),YS(4,4),ZS(4,4),XM(4,4),YM(4,4),ZM(4,4)
      CHARACTER DFILE*80
      DATA LUNDA/10/
C
10    WRITE (LUNTI,*) 'ENTER NAME OF GEOMETRY DATA FILE TO BE READ.'
      READ (LUNTO,'(A20)') DFILE
C
      OPEN (LUNDI,FILE=DFILE,STATUS='OLD',ERR=10)
      READ (LUNDI,'(3(F9.3))') SIZEX,SIZEY,SIZEZ
C
C
      DO 20 I=1,4
        DO 20 J=1,4
          READ (LUNDI,'(3(F9.3))') XS(J,I),YS(J,I),ZS(J,I)
20    CONTINUE
C
C
      DO 30 I=1,4
        DO 30 J=1,4
          READ (LUNDI,'(3(F9.3))') XM(J,I),YM(J,I),ZM(J,I)
30    CONTINUE
C
C
      CLOSE (LUNDI)
      RETURN
      END
C
C********************
C
      SUBROUTINE STODAT (SIZEX,SIZEY,SIZEZ,XS,YS,ZS,XM,YM,ZM)
C
C           This subroutine writes current tool geometry data
C             to user specified file.
C
C      Array definitions:
C
C           XS,YX,ZS - coordinates of share descritive points.
C           XM,YM,ZM - coordinates of moldboard descriptive points.
C
C      Variable definitions:
C
C           DATA - name of file to be read (character string).
C           SIZEX,SIZEY,SIZEZ - scaling factors for x,y,z data.
C
C           LUNTO - logical unit number for terminal output.
```

```
C            LUNTI - logical unit number for terminal input.
C            LUNDI - logical unit number for data input.
C
      COMMON /LUN/LUNTI,LUNTO
      DIMENSION XS(4,4),YS(4,4),ZS(4,4),XM(4,4),YM(4,4),ZM(4,4)
      CHARACTER DFILE*80
      DATA LUNDI/10/
C
10    WRITE (LUNTI,*) ' ENTER NAME OF TOOL GEOMETRY DATA FILE TO ',
     +   'BE WRITTEN TO.'
      READ (LUNTO,'(A10)') DFILE
C
      OPEN (LUNDI,FILE=DFILE,STATUS='NEW',ERR=10)
C
      WRITE (LUNDI,'(3(F9.3))') SIZEX,SIZEY,SIZEZ
C
C
      DO 20 I=1,4
        DO 20 J=1,4
          WRITE (LUNDI,'(3(F9.3))') XS(J,I),YS(J,I),ZS(J,I)
20    CONTINUE
C
C
      DO 30 I=1,4
        DO 30 J=1,4
          WRITE (LUNDI,'(3(F9.3))') XM(J,I),YM(J,I),ZM(J,I)
30    CONTINUE
C
C
      CLOSE (LUNDI)
      RETURN
      END
C
C
C                                                              SURFACE
C
C*********************
C
      SUBROUTINE SURFAC (OL,JJ,PTS,PTT,PX,PY,PZ)
C
C      This subroutine calculates point coordinates to be plotted.
C
C
C      Array definitions:
C
C          PTS,PTT - coordinates of points for describing surface.
C          PX,PY,PZ - coordinates of control points.
C
C      Variable definitions:
C
C          OL,JJ - number of splines used to draw surface.
C
C      Subroutines called:
C
```

```
C          INTNOD - calculates internal control points.
C          POINT - calculates point for describing surface.
C
      DIMENSION PTS(11,3,11),PTT(11,3,11),PX(4,4),PY(4,4),PZ(4,4)
      REAL U,V
      INTEGER OL,JJ
C
      CALL INTNOD (PX)
      CALL INTNOD (PY)
      CALL INTNOD (PZ)
C
C          BLEND IN T-DIRECTION
C
      DO 10 I=1,OL
        U=(I-1)/REAL(OL-1)
        DO 10 J=1,JJ
          V=(J-1)/REAL(JJ-1)
          CALL POINT (PX,U,V,PTT(J,1,I))
          CALL POINT (PY,U,V,PTT(J,2,I))
          CALL POINT (PZ,U,V,PTT(J,3,I))
10    CONTINUE
C
C          BLEND IN S-DIRECTION
C
      DO 20 I=1,OL
        V=(I-1)/REAL(OL-1)
        DO 20 J=1,JJ
          U=(J-1)/REAL(JJ-1)
          CALL POINT (PX,U,V,PTS(J,1,I))
          CALL POINT (PY,U,V,PTS(J,2,I))
          CALL POINT (PZ,U,V,PTS(J,3,I))
20    CONTINUE
      RETURN
      END
C
C*********************
C
      SUBROUTINE INTNOD (P)
C
C      This subroutine calculates the four internal control points.
C
C      Array definition:
C
C          P - coordinates of control points
C
      DIMENSION P(4,4)
      P(2,2)=P(2,1)+P(1,2)-P(1,1)
      P(2,3)=P(1,3)-P(1,4)+P(2,4)
      P(3,2)=P(3,1)-P(4,1)+P(4,2)
      P(3,3)=P(4,3)+P(3,4)-P(4,4)
      RETURN
      END
C
C*********************
```

```
C
      SUBROUTINE POINT (P,S,T,Q)
C
C      This subroutine calculates a point for surface description.
C
C      Array definitions:
C
C          P - control points.
C          SMAT - S matrix.
C          TMAT - T matrix.
C          M - M matrix.
C          SM - S*M
C          SMP - S*M*P
C          MT - M*T
C          Q - calculated surface point.
C
C      Subroutine called:
C
C          MMLT - matrix multiplication (Case Center library
C                   function).
C
      DIMENSION P(4,4),SMAT(4),TMAT(4),SM(4),SMP(4),Q(1)
      REAL M(4,4),MT(4)
      DATA M/-1.,3.,-3.,1.,3.,-6.,3.,0.,-3.,3.,0.,0.,1.,0.,0.,0./
C
      SMAT(1)=S*S*S
      SMAT(2)=S*S
      SMAT(3)=S
      SMAT(4)=1
C
      TMAT(1)=T*T*T
      TMAT(2)=T*T
      TMAT(3)=T
      TMAT(4)=1
C
      CALL MMLT (SM,SMAT,M,1,4,4)
      CALL MMLT (MT,M,TMAT,4,4,1)
      CALL MMLT (SMP,SM,P,1,4,4)
      CALL MMLT (Q,SMP,MT,1,4,1)
C
      RETURN
      END
C
C********************
C
      SUBROUTINE EDGE (P,I,J)
C
C      This subroutine calculates control points along a
C        straight edge.
C
C      Array definitions:
C
C          P - control point coordinates.
C
```

```
      DIMENSION P(4,4)
      IF (I.EQ.1) THEN
        P(J,2)=P(J,1)+((P(J,4)-P(J,1))/3.)
        P(J,3)=P(J,1)+((P(J,4)-P(J,1))*2./3.)
      ELSE
        P(2,J)=P(1,J)+((P(4,J)-P(1,J))/3.)
        P(3,J)=P(1,J)+((P(4,J)-P(1,J))*2./3.)
      ENDIF
      RETURN
      END
C
C                                                                   PLOT
C
C*********************
C
      SUBROUTINE SETPLT (SIZEX,SIZEY,SIZEZ)
C
C      This subroutine sets up the plot and draws the axes for
C        subroutine PLOT.
C
C      Variable definitions:
C
C          SIZEX,SIZEY,SIZEZ - scaling factors for controls points.
C          VX,VY,VZ - view point coordinates.
C          AX,AY,YZ - look at point coordinates.
C          MAG - magnification factor.
C          CX,CY - center position coordinates.
C
C          SIZE - largest of SIZEX,SIZEY,SIZEZ.
C          SIZEA - average of SIZEX,SIZEY,SIZEZ.
C
C      The following subroutines are on the Prime system at The
C        Case Center at Michigan State University.
C
C          CARTVP - sets cartisan view point.
C          LOOKAT - set lookat point.
C          PERSPT - turns on or off perspective viewing.
C          XUP,YUP,ZUP - defines which axis is up on plot.
C          PPCENT - sets distance to 2-D viewing plane from
C                     look at point to view point as a
C                     percentage.
C          DWINDO - sets bounds of area displayed on screen.
C          CENTER - sets center point of display window.
C          MAGNFY - sets display magnification factor.
C          MOVEA3 - move to specified coordinates (3-D).
C          VECTOR - draw line to specified coordinates and
C                     draw an arrowhead (3-D).
C          CHARTK - write text on plot.
C
      COMMON /B/VX,VY,VZ,AX,AY,AZ,MAG,CX,CY
      REAL MAG
      SIZE=-10000.
      IF (SIZEX.GE.SIZE) SIZE=SIZEX
      IF (SIZEY.GE.SIZE) SIZE=SIZEY
```

```
      IF (SIZEZ.GE.SIZE) SIZE=SIZEZ
      SIZEA=(SIZEX+SIZEY+SIZEZ)/3
C
C          SET UP PLOT
C
      CALL CARTVP (VX,VY,VZ)
      CALL LOOKAT (AX,AY,AZ)
      CALL PERSPT (.FALSE.)
      IF (VX.EQ.0..AND.VY.EQ.0.) THEN
        CALL XUP
      ELSE
        CALL ZUP
      ENDIF
      CALL PPCENT (100.)
      CALL DWINDO (-3.*SIZE,2.*SIZE,-2.*SIZE,3.*SIZE)
      CALL CENTER (CX*SIZE,CY*SIZE)
      CALL MAGNFY (MAG)
C
C          DRAW AXES
C
      CALL MOVEA3 (0.,0.,0.)
      CALL VECTOR (2.*SIZEX,0.,0.)
      CALL MOVEA3 (2.1*SIZEX,0.,0.)
      CALL CHARTK ('X',1.5)
      CALL MOVEA3 (0.,0.,0.)
      CALL VECTOR (0.,2.*SIZEY,0.)
      CALL MOVEA3 (0.,2.1*SIZEY,0.)
      CALL CHARTK ('Y',1.5)
      CALL MOVEA3 (0.,0.,0.)
      CALL VECTOR (0.,0.,1.5*SIZEZ)
      CALL MOVEA3 (0.,0.,1.6*SIZEZ)
      CALL CHARTK ('Z',1.5)
      RETURN
      END
C
C********************
C
      SUBROUTINE PLOT (PS,PT,OL,JJ)
C
C      This subroutine plots the surfaces on the region set up
C        in subroutine SETPLT.
C
C      Array definitions:
C
C          PS,PT - contains point locations for 3-D plots
C
C      Variable definitions:
C
C          OL,JJ - number of splines used in PS and PT to
C                      draw the surface.
C
      DIMENSION PS(11,3,11),PT(11,3,11)
      INTEGER OL,JJ
C
```

```
C           DRAW SURFACE
C
C      First draw splines in S direction
C
       DO 10 I=1,OL
         DO 10 J=1,JJ
           IF (J.EQ.1) THEN
             CALL MOVEA3 (PS(J,1,I),PS(J,2,I),PS(J,3,I))
           ELSE
             CALL DRAWA3 (PS(J,1,I),PS(J,2,I),PS(J,3,I))
           ENDIF
10     CONTINUE
C
C      Then in T direction
C
       DO 20 I=1,OL
         DO 20 J=1,JJ
           IF (J.EQ.1) THEN
             CALL MOVEA3 (PT(J,1,I),PT(J,2,I),PT(J,3,I))
           ELSE
             CALL DRAWA3 (PT(J,1,I),PT(J,2,I),PT(J,3,I))
           ENDIF
20     CONTINUE
       RETURN
       END
C
C********************
C
       SUBROUTINE LABPTS (PX,PY,PZ)
C
C       This subroutine labels the moldboard control points.
C
C       Array definitions:
C
C           PX,PY,PZ - contains coordinates of control points.
C
C       Variable definitions:
C
C           Q - answer to question (character string).
C           Q1 - numbers to plotted (character string).
C
C
       DIMENSION PX(4,4),PY(4,4),PZ(4,4)
       CHARACTER Q*3,Q1*32
       Q1=' 1 2 3 4 5 6 7 8 910111213141516'
C
C            LABEL POINTS
C
       DO 2 I=1,4
         DO 2 J=1,4
           CALL MOVEA3 (PX(J,I),PY(J,I),PZ(J,I))
           Q=Q1((((I-1)*4+J)*2)-1):(((I-1)*4+J)*2))
           CALL CHARTK (Q,1.)
2      CONTINUE
```

```
      RETURN
      END
C
C                                                            TRANSFORM
C
C*********************
C
      SUBROUTINE DISTAN (PS,I,J,X,Y,Z,ANGLE1,ANGLE2,DIST)
C
C      This subroutine calculates the distance the soil
C        travels over an element.  These calculations
C        require translating and rotating elements.
C
C      Array definitions:
C
C          PS - coordinates of surface description points.
C
C      Variable definitions:
C
C          I,J - pointers indicating which element in PS is
C                  to be translated and rotated.  Points
C                  PS(J,*,I),PS(J+1,*,I),PS(J,*,I+1) and
C                  PS(J+1,*,I+1) form the element.
C          X - x direction cosine of element orientation.
C          Y - y direction cosine of element orientation.
C          Z - z direction cosine of element orientation.
C          ANGLE1 - the first angle of rotation.
C          ANGLE2 - the second angle of rotation.
C          DIST - the distance the soil travels over the element.
C
C      Subroutines called:
C
C          TRANSL - translates point.
C          ROTATE - rotates point.
C
      DIMENSION PS(11,3,11)
      REAL LEN
C
      X5=X
      Y5=Y
      Z5=Z
C
      CALL TRANSL (PS(J,1,I),PS(J,2,I),PS(J,3,I),X1,Y1,Z1,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J+1,1,I),PS(J+1,2,I),PS(J+1,3,I),X2,Y2,Z2,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J+1,1,I+1),PS(J+1,2,I+1),PS(J+1,3,I+1),X3,Y3,Z3,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J,1,I+1),PS(J,2,I+1),PS(J,3,I+1),X4,Y4,Z4,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
C
      CALL ROTATE (X1,Y1,Z1,X1,Y1,Z1,'Z',ANGLE1)
      CALL ROTATE (X2,Y2,Z2,X2,Y2,Z2,'Z',ANGLE1)
      CALL ROTATE (X3,Y3,Z3,X3,Y3,Z3,'Z',ANGLE1)
```

```
      CALL ROTATE (X4,Y4,Z4,X4,Y4,Z4,'Z',ANGLE1)
      CALL ROTATE (X5,Y5,Z5,X5,Y5,Z5,'Z',ANGLE1)
C
      CALL ROTATE (X1,Y1,Z1,X1,Y1,Z1,'Y',ANGLE2)
      CALL ROTATE (X2,Y2,Z2,X2,Y2,Z2,'Y',ANGLE2)
      CALL ROTATE (X3,Y3,Z3,X3,Y3,Z3,'Y',ANGLE2)
      CALL ROTATE (X4,Y4,Z4,X4,Y4,Z4,'Y',ANGLE2)
      CALL ROTATE (X5,Y5,Z5,X5,Y5,Z5,'Y',ANGLE2)
C
      DIMY=(ABS(Y1-Y4)+ABS(Y2-Y3))/2.
      DIMZ=(ABS(Z1-Z2)+ABS(Z3-Z4))/2.
      LEN=SQRT(DIMY**2+DIMZ**2)
      DIST=DIMZ/Z5
      IF (DIST.GT.LEN) DIST=DIMY/Y5
      IF (DIST.GT.LEN) DIST=LEN
C
      RETURN
      END
C
C********************
C
      SUBROUTINE TRANEL (PS,I,J,ANGLE1,ANGLE2,DIMY,DIMZ)
C
C      This subroutine calculates the y and z dimension of an
C        element.
C
C      Array definitions:
C
C          PS - surface desciption coordinates.
C
C      Variable definitions:
C
C          I,J - pointers indicating which element in PS is
C                  to be translated and rotated.
C          ANGLE1 - the first angle of rotation.
C          ANGLE2 - the second angle of rotation.
C          DIMY - calculated y dimension.
C          DIMZ - calculated z dimension.
C
C      Subroutines called:
C
C          TRANSL - translates element.
C          ROTATE - rotates element.
C
      DIMENSION PS(11,3,11)
C
      CALL TRANSL (PS(J,1,I),PS(J,2,I),PS(J,3,I),X1,Y1,Z1,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J+1,1,I),PS(J+1,2,I),PS(J+1,3,I),X2,Y2,Z2,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J+1,1,I+1),PS(J+1,2,I+1),PS(J+1,3,I+1),X3,Y3,Z3,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
      CALL TRANSL (PS(J,1,I+1),PS(J,2,I+1),PS(J,3,I+1),X4,Y4,Z4,
     +-PS(J,1,I),-PS(J,2,I),-PS(J,3,I))
```

```
C
      CALL ROTATE (X1,Y1,Z1,X1,Y1,Z1,'Z',ANGLE1)
      CALL ROTATE (X2,Y2,Z2,X2,Y2,Z2,'Z',ANGLE1)
      CALL ROTATE (X3,Y3,Z3,X3,Y3,Z3,'Z',ANGLE1)
      CALL ROTATE (X4,Y4,Z4,X4,Y4,Z4,'Z',ANGLE1)
C
      CALL ROTATE (X1,Y1,Z1,X1,Y1,Z1,'Y',ANGLE2)
      CALL ROTATE (X2,Y2,Z2,X2,Y2,Z2,'Y',ANGLE2)
      CALL ROTATE (X3,Y3,Z3,X3,Y3,Z3,'Y',ANGLE2)
      CALL ROTATE (X4,Y4,Z4,X4,Y4,Z4,'Y',ANGLE2)
C
      DIMY=(ABS(Y1-Y4)+ABS(Y2-Y3))/2.
      DIMZ=(ABS(Z1-Z2)+ABS(Z3-Z4))/2.
C
      RETURN
      END
C
C********************
C
      SUBROUTINE ROTATE (X,Y,Z,X1,Y1,Z1,AXIS,ANG)
C
C      This subroutine rotates a point.
C
C      Variable definitions:
C
C          X,Y,Z - point to be rotated.
C          X1,Y1,Z1 - point after rotation.
C          AXIS - axis point is rotated about (character
C                   string).
C          ANG - angle point is to be rotated.
C
C      Subroutines called:
C
C          ROT - calculates T matrix.
C          MMLT - matrix multiplication (Case Center Math
C                   library function).
C
      DIMENSION T(4,4),P(4),RP(4)
      CHARACTER AXIS*1
      DO 10 I=1,4
        DO 10 J=1,4
          IF (I.EQ.J) THEN
            T(I,J)=1.
          ELSE
            T(I,J)=0.
          ENDIF
10    CONTINUE
C
      P(1)=X
      P(2)=Y
      P(3)=Z
      P(4)=1.
C
      IF (AXIS.EQ.'X') CALL ROT (T,2,3,ANG)
```

```
      IF (AXIS.EQ.'Y') CALL ROT (T,1,3,ANG)
      IF (AXIS.EQ.'Z') CALL ROT (T,1,2,ANG)
C
      CALL MMLT(RP,P,T,1,4,4)
C
      X1=RP(1)
      Y1=RP(2)
      Z1=RP(3)
C
      RETURN
      END
C
C*********************
C
      SUBROUTINE ROT (T,I,J,ANG)
C
C      This subroutine calculates T matrix.
C
C      Array definitions:
C
C          T - work array for assigning values to used in
C                subroutine ROTATE.
C
C      Variable definitions:
C
C          I,J - pointers in T matrix.
C          ANG - angle of rotation.
C
C
      DIMENSION T(4,4)
C
      T(I,I)=COS(ANG)
      T(I,J)=SIN(ANG)
      T(J,I)=-SIN(ANG)
      T(J,J)=COS(ANG)
C
      IF (I.EQ.1.AND.J.EQ.3) THEN
        T(I,J)=-SIN(ANG)
        T(J,I)=SIN(ANG)
      ENDIF
C
      RETURN
      END
C
C********************
C
      SUBROUTINE TRANSL (X,Y,Z,X1,Y1,Z1,X2,Y2,Z2)
C
C      This subroutine translated a point.
C
C      Variable definitions:
C
C          X,Y,Z - point to be translated.
C          X1,Y1,Z1 - translated point.
```

```
C          X2,Y2,Z2 - distance to translated.
C
C      Subroutine called:
C
C          MMLT - matrix multiplication (Case Center Math
C                  library function).
C
      DIMENSION T(4,4),P(4),TP(4)
C
      DO 10 I=1,4
        DO 10 J=1,4
          IF (I.EQ.J) THEN
            T(I,J)=1.
          ELSE
            T(I,J)=0.
          ENDIF
10    CONTINUE
C
      P(1)=X
      P(2)=Y
      P(3)=Z
      P(4)=1.
C
      T(4,1)=X2
      T(4,2)=Y2
      T(4,3)=Z2
C
      CALL MMLT (TP,P,T,1,4,4)
C
      X1=TP(1)
      Y1=TP(2)
      Z1=TP(3)
C
      RETURN
      END
C
C                                                                    NEWPTS
C
C*************************
C
      SUBROUTINE NEWSCA (XS,YS,ZS,XM,YM,ZM,SIZEX,SIZEY,SIZEZ)
C
C       This subroutine changes the scaling factors of the
C         control points.
C
C      Array definitions:
C
C            XS,YX,ZS - coordinates of share control points.
C            XM,YM,ZM - coordinates of moldboard control points.
C
C      Variable definitions:
C
C            SIZEX,SIZEY,SIXEZ - scaling factors for control
C              points.
```

```
C           SIZEX1,SIZEY1,SIZEZ1 - intermediate values for
C             SIZEX,SIZEY,SIZEZ.
C
C           LUNTO - logical unit number for terminal output.
C           LUNTI - logical unit number for terminal input.
C
      COMMON /LUN/LUNTI,LUNTO
      DIMENSION XS(4,4),YS(4,4),ZS(4,4),XM(4,4),YM(4,4),ZM(4,4)
      SIZEX1=SIZEX
      SIZEY1=SIZEY
      SIZEZ1=SIZEZ
      WRITE (LUNTI,'(A,F9.3)') 'X-SCALE= ',SIZEX1
      WRITE (LUNTI,'(A,F9.3)') 'Y-SCALE= ',SIZEY1
      WRITE (LUNTI,'(A,F9.3)') 'Z-SCLAE= ',SIZEZ1
5     WRITE (LUNTI,*) 'ENTER NEW X-SCALE, Y-SCALE, Z-SCALE.'
      READ (LUNTO,*,ERR=5) SIZEX,SIZEY,SIZEZ
C
      DO 10 I=1,4
        DO 10 J=1,4
          XS(I,J)=XS(I,J)*SIZEX/SIZEX1
          YS(I,J)=YS(I,J)*SIZEY/SIZEY1
          ZS(I,J)=ZS(I,J)*SIZEZ/SIZEZ1
          XM(I,J)=XM(I,J)*SIZEX/SIZEX1
          YM(I,J)=YM(I,J)*SIZEY/SIZEY1
          ZM(I,J)=ZM(I,J)*SIZEZ/SIZEZ1
10    CONTINUE
C
      RETURN
      END
C
C**********************
C
      SUBROUTINE NEWPM (PX,PY,PZ)
C
C        This subroutine changes the moldboard control points.
C
C       Array definitions:
C
C           PX,PY,PZ - coordinates of moldboard control
C                        points.
C
C           LUNTO - logical unit number for terminal output.
C           LUNTI - logical unit number for terminal input.
C
      COMMON /LUN/LUNTI,LUNTO
      DIMENSION PX(4,4),PY(4,4),PZ(4,4)
      WRITE (LUNTI,*) '   MOLDBOARD DESCRIPTIVE POINTS'
      WRITE (LUNTI,'(A24)') 'PT#       X       Y       Z'
C
      DO 10 I=1,4
        DO 10 J=1,4
          IF (((I-1)*4+J).EQ.5) GOTO 10
          IF (((I-1)*4+J).EQ.6) GOTO 10
          IF (((I-1)*4+J).EQ.7) GOTO 10
```

```
          IF (((I-1)*4+J).EQ.9) GOTO 10
          IF (((I-1)*4+J).EQ.10) GOTO 10
          IF (((I-1)*4+J).EQ.11) GOTO 10
          WRITE (LUNTI,'(I2,3(3X,F6.3))') ((I-1)*4+J),PX(J,I),
     +      PY(J,I),PZ(J,I)
10    CONTINUE
C
15    WRITE (LUNTI,'(/,A34)') 'ENTER PT# TO BE EDITED.(0 TO STOP)'
      READ (LUNTO,*,ERR=15) J
      IF (J.GT.16) GOTO 15
      IF (J.EQ.0) GOTO 20
      I=((J-1)/4)+1
      J=MOD(J-1,4)+1
      WRITE (LUNTI,'(A11)') 'ENTER X,Y,Z'
      READ (LUNTO,*,ERR=15) PX(J,I),PY(J,I),PZ(J,I)
      GOTO 15
20    CONTINUE
      RETURN
      END
C
C*********************
C
      SUBROUTINE NEWPS (PX,PY,PZ)
C
C        This subroutine changes the share control points.
C
C      Array definitions:
C
C            PX,PY,PZ - coordinates of moldboard control
C                         points.
C
C            LUNTO - logical unit number for terminal output.
C            LUNTI - logical unit number for terminal input.
C
      COMMON /LUN/LUNTI,LUNTO
      DIMENSION PX(4,4),PY(4,4),PZ(4,4)
      WRITE (LUNTI,*) '  SHARE DESCRIPTIVE POINTS.'
      WRITE (LUNTI,'(A24)') 'PT#      X      Y      Z'
C
      DO 10 I=1,4
        DO 10 J=1,4
          IF (((I-1)*4+J).EQ.1) GOTO 5
          IF (((I-1)*4+J).EQ.4) GOTO 5
          IF (((I-1)*4+J).EQ.13) GOTO 5
          IF (((I-1)*4+J).EQ.16) GOTO 5
          GOTO 10
5         WRITE (LUNTI,'(I2,3(3X,F6.3))') ((I-1)*4+J),PX(J,I),
     +      PY(J,I),PZ(J,I)
10    CONTINUE
C
15    WRITE (LUNTI,'(/,A34)') 'ENTER PT# TO BE EDITED.(0 TO STOP)'
      READ (LUNTO,*,ERR=15) J
      IF (J.GT.16) GOTO 15
      IF (J.EQ.0) GOTO 20
```

```
      I=((J-1)/4)+1
      J=MOD(J-1,4)+1
      WRITE (LUNTI,'(A11)') 'ENTER X,Y,Z'
      READ (LUNTO,*,ERR=15) PX(J,I),PY(J,I),PZ(J,I)
      GOTO 15
20    CONTINUE
      RETURN
      END
C
C                                                                 PARAM
C
C********************
C
      SUBROUTINE PARAM (SOILTY)
C
C      This subroutine allow editing of the soil and tool
C        parameters.
C
C      Variable definitions:
C
C           DFILE - name of operating conditions file
C                     (character string).
C
C           SOILTY - name of soil type (character string).
C           GAMMA - soil bulk density (kg/cu m).
C           COHES - soil cohesion (kpa).
C           PHI - soil internal friction (deg).
C           MUP - soil-metal friction.
C           ADHES - soil-metal adhesion (kpa).
C           VELTL - tool velocity (m/s).
C           DEPTHF - plowing depth (m).
C
C           LUNTI - logical unit number for terminal input.
C           LUNTO - logical unit number for terminal output.
C           LUNDI - logical unit number for data input.
C           Q - answer to question (character string).
C
      COMMON /A/GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      COMMON /LUN/LUNTI,LUNTO
      CHARACTER SOILTY*40
      CHARACTER Q*3
      CHARACTER DFILE*80
      REAL GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      DATA LUNDI/10/
C
      CALL NEWPAG
      WRITE (LUNTI,'(10X,A)') 'SOIL PARAMETERS'
      WRITE (LUNTI,'(A,T60,A)')' 1)  SOIL TYPE: ',SOILTY
      WRITE (LUNTI,100)' 2)  BULK DENSITY (KG/CU-M) ',GAMMA
      WRITE (LUNTI,100)' 3)  COHESION (kPA) ',COHES
      WRITE (LUNTI,100)' 4)  INTERNAL SOIL FRICTION ANGLE (DEG) ',PHI
      WRITE (LUNTI,100)' 5)  SOIL-METAL FRICTION COEFFICIENT ',MUP
      WRITE (LUNTI,100)' 6)  SOIL ADHESION (KPA)',ADHES
      WRITE (LUNTI,'(10X,A)')'TOOL PARAMETERS'
```

```
      WRITE (LUNTI,100)' 7)  FORWARD VELOCITY (M/S) ',VELTL
      WRITE (LUNTI,100)' 8)  TOOL DEPTH (M) ',DEPTHF
10    WRITE (LUNTI,'(A)') 'ENTER PARAMETER NUMBER TO BE CHANGED. ',
     +'(0 TO STOP)'
C
      READ (LUNTO,*) I
      IF (I.EQ.0) GOTO 20
      WRITE (LUNTI,*) 'ENTER NEW VALUE FOR PARAMETER NUMBER ',I
C
      IF (I.EQ.1) READ (LUNTO,'(A20)') SOILTY
      IF (I.EQ.2) READ (LUNTO,*) GAMMA
      IF (I.EQ.3) READ (LUNTO,*) COHES
      IF (I.EQ.4) READ (LUNTO,*) PHI
      IF (I.EQ.5) READ (LUNTO,*) MUP
      IF (I.EQ.6) READ (LUNTO,*) ADHES
      IF (I.EQ.7) READ (LUNTO,*) VELTL
      IF (I.EQ.8) READ (LUNTO,*) DEPTHF
      GOTO 10
C
20    WRITE (LUNI,*) ' STORE MODIFIED DATA?(Y OR N)'
      READ (LUNTO,'(A3)') Q
      IF (Q.EQ.'N') GOTO 30
      WRITE (LUNTI,*) 'ENTER NEW FILE NAME.'
      READ (LUNTO,'(A20)') DFILE
C
      OPEN (UNIT=LUNDI,FILE=DFILE,STATUS='NEW',ERR=20)
      WRITE (LUNDI,'(A20)') SOILTY
      WRITE (LUNDI,*) GAMMA
      WRITE (LUNDI,*) COHES
      WRITE (LUNDI,*) PHI
      WRITE (LUNDI,*) MUP
      WRITE (LUNDI,*) ADHES
      WRITE (LUNDI,*) VELTL
      WRITE (LUNDI,*) DEPTHF
      CLOSE (LUNDI)
C
30    RETURN
100   FORMAT (A,T60,F9.3)
      END
C
C                                                                    ANGLE
C
C********************
C
      SUBROUTINE ANGLE (PS,I,J,X,Y,ALPHA)
C
C       This subroutine calculates the angle between one of
C         axes and the element.
C
C       Array definitions:
C
C           PS - coordinates of surface description points.
C
C       Variable definitions:
```

```
C
C          I,J - pointers indicating which element in PS is
C                  to have the orientation computed for.
C          X,Y - numbers indicating the plane from which the
C                  angle is calculated.
C          ALPHA - calculated angle.
C
C      Subroutines called:
C
C          INTERP - interpolates between to points.
C
      DIMENSION PS(11,3,11)
      INTEGER X,Y,Z,O,P
      DATA PI/3.14159265359/
C
C      Check which plane is to be considered.
C
      IF (X.EQ.1.AND.Y.EQ.2) Z=3
      IF (X.EQ.2.AND.Y.EQ.3) Z=1
      IF (X.EQ.3.AND.Y.EQ.1) Z=2
C
C       Find point closest to 0,0,0
C
      IF (PS(J,Z,I).GE.PS(J,Z,I+1).AND.PS(J,Z,I).GE.PS(J+1,Z,I)) THEN
        IF (PS(J,Z,I+1).GE.PS(J+1,Z,I)) THEN
          K=J+1
          L=J
          M=J
          N=I
          O=I
          P=I+1
        ELSE
          K=J
          L=J
          M=J+1
          N=I+1
          O=I
          P=I
        ENDIF
      ELSE IF (PS(J,Z,I+1).GE.PS(J,Z,I).AND.PS(J,Z,I+1).GE.PS(J+1,Z,I))
     +  THEN
        IF (PS(J,Z,I).GE.PS(J+1,Z,I)) THEN
          K=J+1
          L=J
          M=J
          N=I
          O=I+1
          P=I
        ELSE
          K=J
          L=J
          M=J+1
          N=I
          O=I+1
```

```
          P=I
        ENDIF
      ELSE IF (PS(J+1,Z,I).GE.PS(J,Z,I).AND.PS(J+1,Z,I).GE.PS(J,Z,I+1))
     +  THEN
        IF (PS(J,Z,I).GE.PS(J,Z,I+1)) THEN
          K=J
          L=J+1
          M=J
          N=I+1
          O=I
          P=I
        ELSE
          K=J
          L=J+1
          M=J
          N=I
          O=I
          P=I+1
        ENDIF
      ENDIF
C
C       Interpolate to find the point used to calculate the
C         angle.
C
      CALL INTERP (PS(K,Z,N),PS(L,Z,O),PS(M,Z,P),
     +  PS(K,X,N),PS(L,X,O),XX)
      CALL INTERP (PS(K,Z,N),PS(L,Z,O),PS(M,Z,P),
     +  PS(K,Y,N),PS(L,Y,O),YY)
C
      IF ((XX-PS(M,X,P)).LT..001.AND.(XX-PS(M,X,P)).GT.-.001) THEN
        ALPHA=PI/2.
      ELSE
        RATYX=((YY-PS(M,Y,P))/(XX-PS(M,X,P)))
        IF ((YY-PS(M,Y,P))/(XX-PS(M,X,P)).LT.0.) THEN
            ALPHA=-ATAN(-RATYX)
        ELSE
            ALPHA=ATAN (RATYX)
        ENDIF
      ENDIF
C
      RETURN
      END
C
C*******************
C
      SUBROUTINE INTERP (MIN1,MAX1,MID1,MIN2,MAX2,MID2)
C
C       This subroutine interpolates between to points.
C
C       Variable definitions:
C
C         MIN1 - minimum independent value.
C         MAX1 - maximum independent value.
C         MID1 - middle independent value.
```

```
C         MIN2 - minimum dependent value.
C         MAX2 - maximum dependent value.
C         MID2 - middle dependent value (calculated).
C
      REAL MIN1,MAX1,MID1,MIN2,MAX2,MID2
C
      IF ((MAX1-MIN1).EQ.0.) THEN
        MID2=MAX1
      ELSE
        MID2=MIN2+((MID1-MIN1)/(MAX1-MIN1)*(MAX2-MIN2))
      ENDIF
C
      RETURN
      END
C
C                                                           FORCES
C
C********************
C
      SUBROUTINE COMFOR (PMS,PSS,OLS,JJS,OLM,JJM,SOILTY)
C
C      This subroutine calculates the forces the soil applies
C        to the tool.
C
C      Array definitions:
C
C          PMS - coordinates of moldboard surface description
C                  points.
C          PSS - coordinates of share surface description
C                  points.
C          SFORCE - Share forces.
C          MFORCE - Moldboard forces.
C          TOTFOR - Total forces.
C          TOTMOM - Total moments.
C
C      Variable definitions:
C
C          OLM,OLS - number of horizontal lines drawn on
C                      moldboard and share respectively.
C          JJM,JJS - number of vertical lines drawn on
C                      moldboard and share respectively.
C          SOILTY - soil type (character string).
C
C          LUNTI - logical unit number for terminal input.
C          LUNTO - logical unit number for terminal output.
C          LUNDA - logical unit number for file data storage.
C          LUNDB - logical unit number for file data storage.
C
C      Subroutines called:
C
C          ELEFOR - calculates element forces.
C          SUMFOR - sums the forces over all elements.
C
      COMMON /A/GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
```

```
      COMMON /LUN/LUNTI,LUNTO
      COMMON /LUND/LUNDA,LUNDB
C
      DIMENSION SFORCE(11,3,11,10),MFORCE(11,3,11,10)
     +,PMS(11,3,11),PSS(11,3,11),TOTFOR(3,2),TOTMOM(3,2)
     +,XNS(11),YNS(11),ZNS(11),XNM(11),YNM(11),ZNM(11)
      REAL MUP,MFORCE
      INTEGER OLS,JJS,OLM,JJM
      CHARACTER SOILTY*40
C
      IC=0
      OPEN (LUNDB,FILE='D_FORCES',STATUS='UNKNOWN')
      WRITE (LUNDB,'(//A)') 'SOIL AND OPERATING CONDITIONS'
      WRITE (LUNDB,'(A,T20,A)') 'SOIL TYPE',SOILTY
      WRITE (LUNDB,5) 'SOIL BULK DENSITY (KG/CU M)',GAMMA
      WRITE (LUNDB,5) 'SOIL COHESION (KPA)',COHES
      WRITE (LUNDB,5) 'SOIL INTERNAL FRICTION ANGLE (DEG)',PHI
      WRITE (LUNDB,5) 'SOIL-METAL FRICTION COEFFICIENT',MUP
      WRITE (LUNDB,5) 'SOIL-METAL ADHESION (KPA)',ADHES
      WRITE (LUNDB,5) 'FORWARD VELOCITY (M/S)',VELTL
      WRITE (LUNDB,5) 'TOOL DEPTH (M)',DEPTHF
C
      WRITE (LUNDA,'(////A)') '       SOIL AND OPERATING CONDITIONS'
      WRITE (LUNDA,'(A,T30,A)') 'SOIL TYPE',SOILTY
      WRITE (LUNDA,5) 'SOIL BULK DENSITY (KG/CU M)',GAMMA
      WRITE (LUNDA,5) 'SOIL COHESION (KPA)',COHES
      WRITE (LUNDA,5) 'SOIL INTERNAL FRICTION ANGLE (DEG)',PHI
      WRITE (LUNDA,5) 'SOIL-METAL FRICTION COEFFICIENT',MUP
      WRITE (LUNDA,5) 'SOIL-METAL ADHESION (KPA)',ADHES
      WRITE (LUNDA,5) 'FORWARD VELOCITY (M/S)',VELTL
      WRITE (LUNDA,5) 'TOOL DEPTH (M)',DEPTHF
      WRITE (LUNDA,6) 'NUMBER OF SHARE ELEMENTS',(JJS-1)*(OLS-1)
      WRITE (LUNDA,6) 'NUMBER OF MOLDBOARD ELEMENTS',(JJM-1)*(OLM-1)
C
5     FORMAT (A,T40,F10.4)
6     FORMAT (A,T40,I3)
C
      WRITE (LUNTI,*) 'ENTER LOCATION OF ZERO MOMENT.(X,Y,Z)'
      READ (LUNTO,*) X,Y,Z
C
C            COMPUTE FORCES ON SHARE
C
      WRITE (LUNDB,'(////A)') '     SHARE ELEMENT FORCES'
      DO 10 J=1,JJS-1
        DO 10 I=1,OLS-1
C
C       Assign force cosines.
C
            IF (J.EQ.1) THEN
              XN=1.
              YN=0.
              ZN=0.
            ELSE
              XN=XNS(I)
```

```
            YN=YNS(I)
            ZN=ZNS(I)
          ENDIF
C
          IF (XNS(I).LE.0..AND.YNS(I).LE.0..AND.ZNS(I).LE.0.) THEN
            II=I+1
            JJ=J
            GOTO 100
          ENDIF
C
C     Calculate element forces.
C
        CALL ELEFOR (PSS,I,J,SFORCE,'S',XN,YN,ZN,IC)
C
        AVEX=(PSS(J,1,I)+PSS(J+1,1,I)+PSS(J+1,1,I+1)+PSS(J,1,I+1))/4.
        AVEY=(PSS(J,2,I)+PSS(J+1,2,I)+PSS(J+1,2,I+1)+PSS(J,2,I+1))/4.
        AVEZ=(PSS(J,3,I)+PSS(J+1,3,I)+PSS(J+1,3,I+1)+PSS(J,3,I+1))/4.
C
C     Sum forces.
C
        CALL SUMFOR (SFORCE,I,J,X,Y,Z,AVEX,AVEY,AVEZ)
C
C         Calculate forces cosines for next row.
C
        IF (SFORCE(J,1,I,10).LE..0001.AND.SFORCE(J,2,I,10).LE..0001
     +    .AND.SFORCE(J,3,I,10).LE..0001) THEN
          XNS(I)=0.
          YNS(I)=0.
          ZNS(I)=0.
        ELSE
          XNS(I)=-(SFORCE(J,1,I,10))/SQRT(SFORCE(J,1,I,10)**2+
     +      SFORCE(J,2,I,10)**2+SFORCE(J,3,I,10)**2)
          YNS(I)=-(SFORCE(J,2,I,10))/SQRT(SFORCE(J,1,I,10)**2+
     +      SFORCE(J,2,I,10)**2+SFORCE(J,3,I,10)**2)
          ZNS(I)=-(SFORCE(J,3,I,10))/SQRT(SFORCE(J,1,I,10)**2+
     +      SFORCE(J,2,I,10)**2+SFORCE(J,3,I,10)**2)
        ENDIF
C
10    CONTINUE
C
C       Compute forces on moldboard.
C
      WRITE (LUNDB,'(////A)') '     MOLDBOARD ELEMENT FORCES'
      DO 20 J=1,JJM-1
        DO 20 I=1,OLM-1
C
C      Assign force cosines.
C
          IF (J.EQ.1) THEN
            IC=1
15          IF ((PMS(J,2,I).GE.PSS(JJS,2,IC).AND.PMS(J,2,I).LT.
     +        PSS(JJS,2,IC+1)).OR.I.EQ.1) GOTO 16
            IC=IC+1
            GOTO 15
```

```
16            XN=XNS(IC)
              YN=YNS(IC)
              ZN=ZNS(IC)
            ELSE
              XN=XNM(I)
              YN=YNM(I)
              ZN=ZNM(I)
            ENDIF
C
C           Calculate element forces.
C
           CALL ELEFOR (PMS,I,J,MFORCE,'M',XN,YN,ZN,IC)
C
           AVEX=(PMS(J,1,I)+PMS(J+1,1,I)+PMS(J+1,1,I+1)+PMS(J,1,I+1))/4.
           AVEY=(PMS(J,2,I)+PMS(J+1,2,I)+PMS(J+1,2,I+1)+PMS(J,2,I+1))/4.
           AVEZ=(PMS(J,3,I)+PMS(J+1,3,I)+PMS(J+1,3,I+1)+PMS(J,3,I+1))/4.
C
C           Sum forces.
C
           CALL SUMFOR (MFORCE,I,J,X,Y,Z,AVEX,AVEY,AVEZ)
C
C           Calculate force cosines for next row.
C
           IF (MFORCE(J,1,I,10).LE..0001.AND.MFORCE(J,2,I,10).LE.0.
     +       .AND.MFORCE(J,3,I,10).LE..0001) THEN
             XNM(I)=0.
             YNM(I)=0.
             ZNM(I)=0.
           ELSE
             XNM(I)=-(MFORCE(J,1,I,10))/SQRT(MFORCE(J,1,I,10)**2+
     +         MFORCE(J,2,I,10)**2+MFORCE(J,3,I,10)**2)
             YNM(I)=-(MFORCE(J,2,I,10))/SQRT(MFORCE(J,1,I,10)**2+
     +         MFORCE(J,2,I,10)**2+MFORCE(J,3,I,10)**2)
             ZNM(I)=-(MFORCE(J,3,I,10))/SQRT(MFORCE(J,1,I,10)**2+
     +         MFORCE(J,2,I,10)**2+MFORCE(J,3,I,10)**2)
           ENDIF
C
20      CONTINUE
C
C         Sum forces over entire tool.
C
        DO 30 K=1,3
          TOTFOR(K,1)=0.
          TOTMOM(K,1)=0.
          TOTFOR(K,2)=0.
          TOTMOM(K,2)=0.
C
          DO 40 J=1,JJS-1
            DO 40 I=1,OLS-1
              TOTMOM(K,1)=TOTMOM(K,1)+SFORCE(J,K,I,9)
              TOTFOR(K,1)=TOTFOR(K,1)+SFORCE(J,K,I,10)
40        CONTINUE
C
        DO 30 J=1,JJM-1
```

```
          DO 30 I=1,OLM-1
            TOTMOM(K,2)=TOTMOM(K,2)+MFORCE(J,K,I,9)
            TOTFOR(K,2)=TOTFOR(K,2)+MFORCE(J,K,I,10)
30      CONTINUE
C
        WRITE (LUNDB,'(////)')
        WRITE (LUNDB,50) '      TOOL TOTALS','X','Y','Z'
        WRITE (LUNDB,'(/,A)') '   FORCES (N)'
        WRITE (LUNDB,60) 'SHARE',(TOTFOR(K,1),K=1,3)
        WRITE (LUNDB,60) 'MOLDBOARD',(TOTFOR(K,2),K=1,3)
        WRITE (LUNDB,'(T33,A)') '------------------------------------'
        WRITE (LUNDB,60) 'TOTAL',(TOTFOR(K,1)+TOTFOR(K,2),K=1,3)
        WRITE (LUNDB,'(//A)') '   MOMENTS (N-M)'
        WRITE (LUNDB,60) 'SHARE',(TOTMOM(K,1),K=1,3)
        WRITE (LUNDB,60) 'MOLDBOARD',(TOTMOM(K,2),K=1,3)
        WRITE (LUNDB,'(T33,A)') '------------------------------------'
        WRITE (LUNDB,60) 'TOTAL',(TOTMOM(K,1)+TOTMOM(K,2),K=1,3)
        WRITE (LUNDB,60) 'LOCATION OF ZERO MOMENT',X,Y,Z
C
        WRITE (LUNDA,'(//)')
        WRITE (LUNDA,50) '      TOOL TOTALS','X','Y','Z'
        WRITE (LUNDA,'(/,A)') '   FORCES (N)'
        WRITE (LUNDA,60) 'SHARE',(TOTFOR(K,1),K=1,3)
        WRITE (LUNDA,60) 'MOLDBOARD',(TOTFOR(K,2),K=1,3)
        WRITE (LUNDA,'(T33,A)') '------------------------------------'
        WRITE (LUNDA,60) 'TOTAL',(TOTFOR(K,1)+TOTFOR(K,2),K=1,3)
C
        WRITE (LUNTI,50) '      TOOL TOTALS','X','Y','Z'
        WRITE (LUNTI,'(/,A)') '   FORCES (N)'
        WRITE (LUNTI,60) 'SHARE',(TOTFOR(K,1),K=1,3)
        WRITE (LUNTI,60) 'MOLDBOARD',(TOTFOR(K,2),K=1,3)
        WRITE (LUNTI,'(T33,A)') '------------------------------------'
        WRITE (LUNTI,60) 'TOTAL',(TOTFOR(K,1)+TOTFOR(K,2),K=1,3)
        WRITE (LUNTI,'(//A)') '   MOMENTS (N-M)'
        WRITE (LUNTI,60) 'SHARE',(TOTMOM(K,1),K=1,3)
        WRITE (LUNTI,60) 'MOLDBOARD',(TOTMOM(K,2),K=1,3)
        WRITE (LUNTI,'(T33,A)') '------------------------------------'
        WRITE (LUNTI,60) 'TOTAL',(TOTMOM(K,1)+TOTMOM(K,2),K=1,3)
        WRITE (LUNTI,60) 'LOCATION OF ZERO MOMENT',X,Y,Z
50      FORMAT (A,T28,3(TR11,A))
60      FORMAT (A,T33,3(F10.2,2X))
C
        CLOSE (LUNDB)
        RETURN
C
C         Write message.
C
100     WRITE (LUNTI,1000) II,JJ
        WRITE (LUNDA,1000) II,JJ
        WRITE (LUNDB,1000) II,JJ
1000    FORMAT ( '******************** ATTENTION ********************',/,
     +    ' SOIL IS NOT IN CONTACT WITH SURFACE AT POINT ',I2,I2)
C
        CLOSE (LUNDB)
```

```
      RETURN
      END
C
C*********************
C
      SUBROUTINE ELEFOR (PMS,I,J,FORCE,TOOLP,XN,YN,ZN,IC)
C
C       This subroutine calculates the soil forces on an element.
C
C       Array definitions:
C
C           PMS - coordinates of surface description points.
C           I,J - pointers indicating which element in PMS is
C                   to have forces calculated for.
C           FORCE - calculated forces for elements where
C                     1st and 3rd dimensions indicate element
C                     2nd dimension indicates coordinate
C                       1 - x; 2 - y; 3 - z
C                     4th indicates which force
C                       1 - normal load
C                       2 - soil-metal friction
C                       3 - adhesion
C                       4 - moments for each element
C                       5 - total force for each element
C           VELM - velocity of soil over moldboard element.
C           VELS - velocity of soil over share element.
C
C       Variable definitions:
C
C           TOOLP - indicates which tool part is being calculated.
C                     S for share, M for moldboard (character string)
C            SOILTY - name of soil type (character string).
C            GAMMA - soil bulk density (kg/cu m).
C            COHES - soil cohesion (kpa).
C            PHI - soil internal friction (deg).
C            MUP - soil-metal friction.
C            ADHES - soil-metal adhesion (kpa).
C            VELTL - tool velocity (m/s).
C            DEPTHF - plowing depth (m).
C
C           XN,YN,ZN - force direction cosines.
C           IC - pointer to indicate which share element is before
C                  moldboard element.
C
C           MU - internal soil friction coefficient.
C           FO - contact area of soil block and tool.
C           F1 - area of forward shear failure surface.
C           WEIGHT - weight of soil block.
C           ACCEL - soil acceleration.
C           NO - normal load on tool.
C
C           LUNTI - logical unit number for terminal input.
C           LUNTO - logical unit number for terminal output.
C           LUNDA - logical unit number for data file.
```

```
C          LUNDB - logical unit number for data file.
C
C      Subroutines called:
C
C          ANGLE - computes angles of angles elements.
C          TRANEL - translates elements.
C          DISTAN - computes distance soil travels over element.
C
      COMMON /A/GAMMA,COHES,PHI,MUP,ADHES,VELTL,DEPTHF
      COMMON /LUN/LUNTI,LUNTO
      COMMON /LUND/LUNDA,LUNDB
C
      DIMENSION PMS(11,3,11),FORCE(11,3,11,5),VELM(11),VELS(11)
      REAL MU,MUP,NO
      CHARACTER TOOLP*1
      SAVE VELM,VELS
      DATA PI/3.141592654/
C
C      Calculate element orientation
C
      CALL ANGLE (PMS,I,J,1,2,ALPHA)
      CALL ANGLE (PMS,I,J,3,1,DELTA)
      PHIP=ATAN(SIN(ALPHA)*TAN(DELTA))
      BETA=.5*(90.-PHI)/(180./PI)
C
C       Compute other parameters
C
      IF (J.EQ.1.AND.TOOLP.EQ.'S') THEN
        COHESE=COHES*1000.
      ELSE
        COHESE=0.
      ENDIF
C
      MU=TAN(PHI/(180./PI))
      CALL TRANEL (PMS,I,J,(90./(180./PI)-ALPHA),-PHIP,DIMY1,DIMZ1)
      FO=DIMY1*DIMZ1
      CALL TRANEL (PMS,I,J,(90./(180./PI)-ALPHA),-BETA,DIMY,DIMZ)
      F1=DEPTHF/SIN(BETA)*DIMY
      WEIGHT=GAMMA*9.81*F1*DIMZ
C
C      Calculate force direction cosines.
C
      IF (J.EQ.1.AND.TOOLP.EQ.'S') THEN
        XN1=SIN(DELTA)
        YN1=0.
        ZN1=COS(DELTA)
      ELSE
        XN=YN/TAN(ALPHA)+ZN*TAN(DELTA)
        XN1=XN/SQRT(XN**2+YN**2+ZN**2)
        YN1=YN/SQRT(XN**2+YN**2+ZN**2)
        ZN1=ZN/SQRT(XN**2+YN**2+ZN**2)
      ENDIF
C
C       Compute accelerations.
```

```
C
      CALL DISTAN (PMS,I,J,XN1,YN1,ZN1,(90./(180./PI)-ALPHA),
     +  -PHIP,DIST)
      SDIST=DIMZ1*COS(PHIP)/SIN(BETA)
      TIME=DIST/VELTL
      IF (J.EQ.1) THEN
        IF (TOOLP.EQ.'S') THEN
          VEL1=0.
          VELS(I)=SDIST/TIME
          VEL2=VELS(I)
        ELSE
          VEL1=VELS(IC)
          VELM(I)=SDIST/TIME
          VEL2=VELM(I)
        ENDIF
      ELSE
        IF (TOOLP.EQ.'S') THEN
          VEL1=VELS(I)
          VELS(I)=SDIST/TIME
          VEL2=VELS(I)
        ELSE
          VEL1=VELM(I)
          VELM(I)=SDIST/TIME
          VEL2=VELM(I)
        ENDIF
      ENDIF
      ACCEL=GAMMA*F1*DIMZ*(VEL2-VEL1)/TIME
C
      NO=(COHESE*F1+ACCEL)*COS(PHIP-BETA)+WEIGHT*SIN(PHIP)
      IF (NO.LE.0.) NO=0.
C
C      Calculate normal load.
C
      FORCE(J,1,I,1)=NO*COS(PHIP)*SIN(ALPHA)
      FORCE(J,2,I,1)=-NO*COS(PHIP)*COS(ALPHA)
      FORCE(J,3,I,1)=-NO*SIN(PHIP)
C
C      Calculate Soil-metal frictional force.
C
      FORCE(J,1,I,2)=NO*MUP*(XN1)
      FORCE(J,2,I,2)=NO*MUP*(YN1)
      FORCE(J,3,I,2)=NO*MUP*(ZN1)
C
C      Calculate adhesion force.
C
      FORCE(J,1,I,3)=ADHES*F0*(XN1)
      FORCE(J,2,I,3)=ADHES*F0*(YN1)
      FORCE(J,3,I,3)=ADHES*F0*(ZN1)
C
C      Print out info to LUNDB.
C
      WRITE (LUNDB,'(//A)') 'ELEMENT COORDINATES'
      WRITE (LUNDB,'(11X,3(A,9X))') 'X','Y','Z'
      WRITE (LUNDB,35) I,J,PMS(J,1,I),PMS(J,2,I),PMS(J,3,I)
```

```
      WRITE (LUNDB,35) I,J+1,PMS(J+1,1,I),PMS(J+1,2,I),PMS(J+1,3,I)
      WRITE (LUNDB,35) I+1,J+1,PMS(J+1,1,I+1),PMS(J+1,2,I+1),
     +  PMS(J+1,3,I+1)
      WRITE (LUNDB,35) I+1,J,PMS(J,1,I+1),PMS(J,2,I+1),PMS(J,3,I+1)
C
      WRITE (LUNDB,'(A,T28,3(TR11,A))') '     FORCES (N)','X','Y','Z'
      WRITE (LUNDB,50)'NORMAL LOAD',(FORCE(J,K,I,1),K=1,3)
      WRITE (LUNDB,50)'SOIL-METAL FRICTION',(FORCE(J,K,I,2),K=1,3)
      WRITE (LUNDB,50)'ADHESION',(FORCE(J,K,I,3),K=1,3)
C
35    FORMAT (I2,3X,I2,3(F9.3))
50    FORMAT (A,T33,3(F10.2,2X))
C
      RETURN
      END
C
C********************
C
      SUBROUTINE SUMFOR (FORCE,I,J,X,Y,Z,AVEX,AVEY,AVEZ)
C
C      This subroutine sums forces for an element.
C
C      Array definitions:
C
C          FORCE - forces values to be summed.
C
C      Variable definitions:
C
C          I,J - pointers indication which element is being
C                  considered.
C          X,Y,Z - coordinates of point force moments are being
C                    summed about.
C          AVEX,AVEY,AVEZ - coordinates of center of point.
C
C          LUNDA - logical unit number for data file.
C          LUNDB - logical unit number for data file.
C
      COMMON /LUND/LUNDA,LUNDB
      DIMENSION FORCE(11,3,11,5)
C
      DO 5 K=1,3
        FORCE(J,K,I,5)=0.
5     CONTINUE
C
C       Sum forces.
C
      DO 10 K=1,3
        FORCE(J,1,I,5)=FORCE(J,1,I,5)+FORCE(J,1,I,K)
        FORCE(J,2,I,5)=FORCE(J,2,I,5)+FORCE(J,2,I,K)
        FORCE(J,3,I,5)=FORCE(J,3,I,5)+FORCE(J,3,I,K)
10    CONTINUE
C
C       Calculate moments.
C
```

```
      FORCE(J,1,I,4)=(FORCE(J,2,I,5)*(Z-AVEZ))
     +  -(FORCE(J,3,I,10)*(Y-AVEY))
      FORCE(J,2,I,4)=-(FORCE(J,1,I,5)*(Z-AVEZ))
     +  +(FORCE(J,3,I,5)*(X-AVEX))
      FORCE(J,3,I,4)=(FORCE(J,1,I,5)*(Y-AVEY))
     +  -(FORCE(J,2,I,5)*(X-AVEX))
C
      WRITE (LUNDB,'(T33,36A)')'-----------------------------------'
      WRITE (LUNDB,30) 'TOTALS',(FORCE(J,K,I,10),K=1,3)
      WRITE (LUNDB,'(/)')
      WRITE (LUNDB,30) 'DISTANCE TO ZERO MOMENT',X-AVEX,Y-AVEY,Z-AVEZ
      WRITE (LUNDB,30) 'MOMENT (N-M)',(FORCE(J,K,I,9),K=1,3)
30    FORMAT (A,T33,3(F10.2,2X))
C
      RETURN
      END
```

APPENDIX C

PLOW DATA FILES

PLOW DATA FILES

This section contains the data files used to generate the information for this research. Each file will be referred to by the name as it exists on the Prime 750 located at The Case Center for Computer Aided Design, Michigan State University. A brief explanation of the function of each file will also be included. Those interested in a detailed description of each value are encouraged to review the fortran source code for the variable name the values are read into.

The first file is D_TINE. It is the geometry file for the 63 mm wide inclined blade.

```
0.357    0.063    0.250
0.000    0.000    0.000
0.083    0.000    0.083
0.167    0.000    0.167
0.089    0.000    0.062
0.000    0.333    0.000
0.083    0.333    0.083
0.167    0.333    0.167
0.250    0.333    0.250
0.000    0.667    0.000
0.083    0.667    0.083
0.167    0.667    0.167
0.250    0.667    0.250
0.000    0.063    0.000
0.083    1.000    0.083
0.167    1.000    0.167
0.089    0.063    0.062
0.089    0.000    0.062
0.178    0.000    0.125
0.268    0.000    0.187
0.357    0.000    0.250
0.089    0.021    0.062
0.500    0.333    0.500
```

0.750	0.333	0.750
0.357	0.021	0.250
0.089	0.042	0.062
0.500	0.667	0.500
0.750	0.667	0.750
0.357	0.042	0.250
0.089	0.063	0.062
0.178	0.063	0.125
0.268	0.063	0.187
0.357	0.063	0.250

The following file is D_FORD. It is the geometry file for the 16 inch wide Ford moldboard plow.

0.3459	0.4074	0.2594
0.0000	0.0000	0.0000
0.0359	0.0000	0.0160
0.0719	0.0000	0.0320
0.1078	0.0000	0.0480
0.1153	0.1358	0.0000
0.1513	0.1358	0.0160
0.1658	0.1064	0.0321
0.2018	0.1064	0.0481
0.2306	0.2716	0.0000
0.2452	0.2422	0.0160
0.2811	0.2422	0.0321
0.2957	0.2128	0.0481
0.3459	0.4074	0.0000
0.3605	0.3780	0.0160
0.3751	0.3486	0.0321
0.3896	0.3192	0.0481
0.1078	0.0000	0.0480
0.1842	0.0000	0.0826
0.2476	0.0000	0.1651
0.2682	-0.0025	0.2594
0.1958	0.0991	0.0474
0.2722	0.0991	0.0819
0.7034	0.4597	0.4391
0.7239	0.4572	0.5334
0.2838	0.1981	0.0467
0.4962	0.3073	0.0467
0.8185	0.5343	0.1671
0.8128	0.6096	0.2794
0.3718	0.2972	0.0461
0.5842	0.4064	0.0461
0.8382	0.4890	0.1270
0.8325	0.5643	0.2393

The following file is D_FORDPLOW. It contains the descriptor points for the 16 inch wide Ford moldboard plow.

1	0.1984	0.1035	0.0489
2	0.0000	0.0000	0.0000
3	0.1285	0.1502	0.0014
4	0.2301	0.2673	0.0006
5	0.3459	0.4074	0.0000
6	0.3896	0.3192	0.0481
7	0.2886	0.2053	0.0487
8	0.1078	0.0000	0.0480
9	0.1723	0.0024	0.0918
10	0.2124	-0.0030	0.1387
11	0.2680	0.0635	0.1266
12	0.3604	0.1827	0.1056
13	0.4627	0.2861	0.1029
14	0.6070	0.4034	0.0800
15	0.4955	0.3557	0.0551
16	0.3718	0.2972	0.0461
17	0.5638	0.3059	0.2002
18	0.6394	0.3773	0.1941
19	0.7817	0.5011	0.1581
20	0.2492	-0.0039	0.2017
21	0.2682	-0.0025	0.2594
22	0.3033	0.0494	0.2067
23	0.3235	0.0475	0.2830
24	0.4340	0.1578	0.3440
25	0.5453	0.2841	0.3816
26	0.6663	0.4223	0.3664
27	0.7501	0.5167	0.3402
28	0.7530	0.4896	0.2538
29	0.6472	0.3783	0.2608
30	0.5081	0.2418	0.2418
31	0.4003	0.1542	0.1868
32	0.7990	0.5605	0.3096
33	0.8325	0.5643	0.2393
34	0.8181	0.5338	0.1893

The thirteen files that follow are operating conditions file. The first file is the operating conditions for the moldboard plow model verification (Table 6.2). For the remainder of the files, the number at the end of each file name corresponds to the case number of Figure 6.1.

D_SANDYLOAM
Norfolk Sandy Loam
1770.
19.7
34.8
.4
0.
1.34
.203

D_LMCLAY1
Low Moisture Clay-Case 1
1034.
35.9
38.
.445
1.8
1.34
.25

D_LMCLAY2
Low Moisture Clay-Case 2
1007.
17.2
45.
.625
12.1
1.34
.25

D_LMCLAY3
Low Moisture Clay-Case 3
1089.
48.3
35.
.364
11.
1.34
.25

D_HMCLAY4
High Moisture Clay-Case 4
1025.
15.5
39.
.625
12.1
1.34
.25

D_HMCLAY5
High Moisture Clay-Case 5
980.
55.2
26.
.466
6.9
1.34
.25

D_HMCLAY6
High Moisture Clay-Case 6
1043.
37.9
24.
.364
10.5
1.34
.25

D_LMSAND7
Low Moisture Sandy Clay-Case 7
1379.
6.9
36.
.445
2.8
1.34
.25

D_LMSAND8
Low Moisture Sandy Clay-Case 8
1361.
17.2
34.
.306
5.4
1.34
.25

D_LMSAND9
Low Moisture Sandy Clay-Case 9
1415.
2.
34.
.404
3.5
1.34
.25

D_HMSAND10
High Moisture Sandy Clay-Case 10
1406.
10.3
33.
.364
4.6
1.34
.25

D_HMSAND11
High Moisture Sandy Clay-Case 11
1288.
10.3
32.
.364
9.
1.34
.25

D_HMSAND12
High Moisture Sandy Clay-Case 12
1325.
17.2
40.
.510
3.5
1.34
.25

APPENDIX D

MEASURED DATA

MEASURED DATA

The following file contains the data points measured on the 16 inch wide Ford moldboard plow. To adequately measure the entire surface, the plow had to be positioned three times. The point numbers measured before and after the moves were used to piece together the data. The file listed as D_FORDPLOW in Appendix C is the result of the manipulation of this file.

Points 1-18 were taken at the first plow position.

1	0.000	0.000	5.200
2	8.605	2.327	6.568
3	0.999	2.111	8.203
4	-4.980	2.016	9.437
5	-11.951	1.901	11.050
6	-11.135	-0.218	7.380
7	-5.253	-0.109	6.232
8	5.305	0.149	4.127
9	3.296	-1.742	2.781
10	2.237	-3.699	1.780
11	-1.110	-3.290	2.500
12	-6.875	-2.566	3.965
13	-12.548	-2.607	4.753
14	-19.894	-1.952	4.917
15	-15.303	-0.732	6.017
16	-9.937	-0.123	6.958
17	-16.046	-6.679	3.220
18	-20.123	-6.551	3.634

Points 11 and 17-31 were taken at the second position.

11	0.000	0.000	8.000
17	15.201	2.210	7.291
18	19.251	1.754	6.882
19	26.578	-0.030	6.500
20	-2.035	3.176	9.467
21	-1.305	5.438	9.724
22	0.936	3.216	9.077

23	1.658	6.218	9.417
24	7.852	8.283	8.441
25	14.407	9.380	7.050
26	21.455	8.372	5.658
27	26.281	7.061	4.774
28	25.572	3.757	5.918
29	19.625	4.360	6.855
30	11.987	4.022	7.855
31	6.431	2.120	8.169

Points 17-19, 27, 28 and 32-34 were taken at the third position.

17	-11.361	2.202	7.288
18	-7.303	1.762	6.880
19	0.000	0.000	6.500
27	-0.281	7.103	4.780
28	-0.980	3.792	5.914
32	2.227	5.768	4.653
33	3.253	3.005	5.520
34	1.994	1.134	6.260

LIST OF REFERENCES

LIST OF REFERENCES

Carlson, E.C., 1961. Plow and Computers. Agricultural Engineering. 42(6):292-295, 307.

Emmett, W.T. Jr., 1983. Computer-Aided Performance Predictions of Coated Moldboard Plows. M.S. Thesis Cornell University, Ithaca, NY.

Emmett, W.T. Jr. and W.W. Gunkel, 1983. A Computer-Aided Analysis of the Performance of Coated Moldboard Plow Surfaces. American Society of Agricultural Engineers Paper No. 83-1046.

Faux, I.D. and M.J. Pratt, 1979. Computational Geometry for Design and Manufacture. Ellis Horwood Limited, West Sussex, England.

Foley, J.D. and A. VanDam, 1982. Fundamentals of Interactive Computer Graphics. Addison Wesley Publishing Company, Reading, Massachusetts.

Gill, W.R. and F.E. VandenBerg, 1967. Soil Dynamics in Tillage and Traction. Agricultural Handbook No. 316. U.S. Government Printing Office, Washington, D.C.

Hettiaratchi, D.R.P. and A.R. Reece, 1967. Symmetrical Three-Dimensional Soil Failure. Journal of Terramechanics. 4(3):45-67.

McKyes, E., 1977. The Calculation of Draft Forces and Soil Failure Boundaries of Narrow Cutting Blades. Transactions of American Society of Agricultural Engineers. 15(2):111-116.

Nichols, M.L., 1931a. Dynamic Properties of Soil. I. An Explanation of the Dynamic Properties of Soils by Means of Colloidal Films. Agricultural Engineering. 12(7):259-264.

Nichols, M.L., 1931b. The Dynamic Properties of Soil. II. Soil and Metal Friction. Agricultural Engineering. 12(8):321-324.

Nichols, M.L. and T.H. Kummer, 1932. The Dynamic Properties of Soil. IV. A Method of Analysis of Plow Moldboard Design Based upon Dynamic Properties of Soil. Agricultural Engineering. 13(11):279-285.

Nichols, M.L. and I.F. Reed, 1934. Soil Dynamics: VI. Physical Reactions of Soils to Moldboard Surfaces. Agricultural Engineering. 15(6):187-190.

Nichols, M.L., I.F. Reed and C.A. Reaves, 1958. Soil Reaction: To Plow Share Design. Agricultural Engineering. 39(6):336-339.

O'Callaghan, J.R. and J.G McCoy, 1965. The Handling of Soil by Mouldboard Ploughs. Journal of Agricultural Engineering Research. 10(1):23-25.

Orlandea, N., S.H. Chen and T. Berenyi, 1983. A Study of Soil-Tool Interactions. Transactions of American Society of Agricultural Engineers. 21(5):1619-1625.

Perumpral, J.V., R.D. Grisso and C.S. Desai, 1983. A Soil-Tool Model Based on Limit Equilibrium Analysis. Transactions of American Society of Agricultural Engineers. 21(3):991-995.

Plasse, R., G.S.V. Raghavan and F. McKyes, 1983. Simulation of Narrow Blade Performance in Different Soils. American Society of Agricultural Engineers Paper No. 83-1539.

Rowe, R.J. and K.K. Barnes, 1961. Influence of Speed on Elements of Draft of a Tillage Tool. Transactions of the American Society of Agricultural Engineers. 6(1):55-57.

Siemens, J.C., J.A. Weber and T.H. Thorburn, 1965. Mechanics of Soil as Influenced by Model Tillage Tools. Transactions of American Society of Agricultural Engineers. 8(1):1-7.

Soehne, W., 1959. [Investigations on the Shape of Plough Bodies for High Speeds.] Grundlagen der Landtechnik. 11:22-39, (National Institute of Agricultural Engineering., English Translation 87.)

White, E.A., 1918. A Study of the Plow Bottom and its Action Upon the Furrow Slice. Journal of Agricultural Research. 12:149-182, illus.