4

AAMRL-TR-88-009

ARTICULATED TOTAL BODY MODEL ENHANCEMENTS
Volume 1: Modifications

LOUISE A. OBERGEFELL

BIODYNAMICS AND BIOENGINEERING DIVISION
HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY

JOHN T. FLECK

J&J TECHNOLOGIES INC.
92 HENNING DRIVE
ORCHARD PARK, NY 14127

INTS KALEPS

BIODYNAMICS AND BIOENGINEERING DIVISION
HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY

THOMAS R. GARDNER

SYSTEMS RESEARCH LABORATORIES, INC.
2800 INDIAN RIPPLE ROAD
DAYTON, OH 45440

JANUARY 1988

FINAL REPORT

Approved for public release; distribution is unlimited.

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

Please do not request copies of this report from the Armstrong Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-88-009

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Henning E. von Gierke

HENNING E. VON GIERKE, Dr Ing
Director
Biodynamics and Bioengineering Division
Armstrong Aerospace Medical Research Laboratory

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS AD-A198 726
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AAMRL-TR-88-009	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Harry G. Armstrong Aerospace Medical Research Laboratory 6b. OFFICE SYMBOL (If applicable) AAMRL/BBM	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433-6573	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62202F	7231	23	01

11. TITLE (Include Security Classification)
ARTICULATED TOTAL BODY MODEL ENHANCEMENTS
VOLUME 1: MODIFICATIONS

12. PERSONAL AUTHOR(S) Obergefell, Louise A., AAMRL; Fleck, John T., J&J Technologies Inc.; Kaleps, Ints, AAMRL; Gardner, Thomas R., Systems Research Laboratories, Inc.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 83 NOV TO 87 DEC	88 JAN	91

16. SUPPLEMENTARY NOTATION
Effort partially funded by the National Highway Traffic Safety Administration.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computer Simulation, Rigid Body Dynamics, Mathematical Model, Three-Dimensional Dynamics, Articulated Total Body Model, Crash Victim Simulator
20	11		
12	05		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Articulated Total Body (ATB) Model is used at the Armstrong Aerospace Medical Research Laboratory (AAMRL) to study human body biomechanics in various dynamic environments, especially aircraft ejection with windblast exposure. In order to improve the model's predicted results and capabilities, a number of modifications have been made. These modifications include the capability to have segment contact ellipsoids block the wind from other segments, an option to prescribe velocity dependent wind forces, a correction to prevent angular drift in the joints, improved contact force calculations for segment contact near a plane's edge, the capability to specify as input multi-axis angular displacements to describe the vehicle motion, a slip joint capability and a hyperellipsoid option. Along with these major changes, a number of minor corrections and clarifications have been included to form the ATB-IV version. The results of these modifications have been documented in three volumes of which this is Volume 1, Modifications. It contains a

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Louise A. Obergefell	22b. TELEPHONE (Include Area Code) (513) 255-3665 22c. OFFICE SYMBOL AAMRL/BBM

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Block 19 ABSTRACT (Continued)

description of the ATB-IV modifications and the theory used to develop them. Volume 2 is an updated user's guide containing the new input description and Volume 3 is an updated programmer's guide containing a listing of all the ATB-IV subroutines.

PREFACE

This report incorporates the work done in a number of different efforts to improve the Articulated Total Body (ATB) Model's capability to simulate human body biomechanics in various dynamic environments, especially aircraft ejection with windblast exposure.

The majority of modifications to the model fall into six categories:

- wind force option
- joint drift correction
- edge effect option
- multi-axis angular displacement vehicle motion prescription
- slip joint option
- hyperellipsoid option

These improvements have been combined to form the ATB-IV version on the Armstrong Aerospace Medical Research Laboratory's (AAMRL) Concurrent computer system at Wright Patterson Air Force Base. AAMRL, Systems Research Laboratories Inc., J & J Technologies Inc., and the National Highway Traffic Safety Administration have all contributed to the technical work described herein.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

The Articulated Total Body (ATB) Model is used at the Armstrong Aerospace Medical Research Laboratory (AAMRL) for predicting gross human body response in various dynamic environments, especially aircraft ejection with windblast exposure. Aerodynamic force application and a harness belt capability were added to the Crash Victim Simulation (CVS) Program (Ref. 1), by Calspan Corporation in 1975 for AMRL (Ref 2.), and the resulting program became known as the ATB model. In 1980, Calspan made a number of modifications to the ATB model combining it with the then current 3-D Crash Victim Simulation program to form the ATB-II model (Ref. 3). Complete documentation of the program through the ATB-II version was performed by Calspan Corp. (Ref. 4). A new version, ATB-III, was generated which included the improvements made by J & J Technologies Inc to model the body response to windblast for AMRL (Ref. 5).

A number of additional efforts have been made to improve various aspects of the ATB-III model, with emphasis on its capability to simulate aircraft ejection with windblast exposure as well as complex automobile accidents.

This volume, Modifications, contains a description of the major changes made to create ATB-IV and the theory used to develop them.

Section Two of this volume includes a new wind force option allowing segment contact ellipsoids to block the wind as well as other aerodynamic force improvements. Corrections to prevent angular drift in the joints are described in Section Three. The edge effect option in Section Four ensures that a contact of a plane with an ellipsoid will not be ignored and that a smaller force will be applied when only part of the contact area is within the plane boundaries. Section Five contains an improvement allowing the prescription of multi-axis angular displacements to describe the vehicle motion. A new option allowing a joint to slide along an axis is explained in Section Six. Section Seven contains a new hyperellipsoid option. A summary of other modifications that form the ATB-IV version is included in Section Eight.

These changes have been made so that previous input decks are valid with changes required only in the H cards. The updated input description outlining any changes needed and describing the use of the new options is described in Volume 2, the User's Guide. Sample input decks using the new options and the resulting output are also included in Volume 2, along with an updated list of numbered stops. Volume 3, the Programmer's Guide, contains the listing of the updated ATB-IV program.

2.0 AERODYNAMIC FORCES

The aerodynamic routines added to the model in 1975 (Ref. 2) have some limitations which make simulating wind forces difficult. For example the aerodynamic pressure is prescribed tabularly as a function of time, but this requires knowing the velocity profile of the seat before a simulation is made. Since the seat's motion may depend on the wind forces, estimation or trial and error has to be used in defining the aerodynamic pressure.

Also the aerodynamic forces are applied to the entire projected contact ellipsoid area that has penetrated the wind plane even if the ellipsoid is partially or fully blocked by another ellipsoid. This causes a disproportionate amount of force to be applied in many cases. This is especially significant for the torso segments where the ellipsoids substantivally overlap. More than 30 percent of three torso segment's combined area is within another ellipsoid, resulting in the aerodynamic force on these segments being much too large.

The original aerodynamic forces are applied to any segment by specifying an aerodynamic pressure, a boundary plane, and a contact ellipsoid associated with the segment. When the ellipsoid penetrates the boundary plane, the wetted area is estimated and a pressure from the tabular data, defining the time dependent aerodynamic pressure, is used to calculate the force and torque that is applied to the segment.

Three changes have been made to the routines to allow more flexibility in applying aerodynamic forces.

1. The aerodynamic pressure can be a function of a segment's velocity.

2. A time dependent drag coefficient can be included in calculating the wind force.

3. An additional method of calculating the wetted area, that allows segments to be defined which block the ellipsoid from the wind, has been added as an option.

2.1. VELOCITY DEPENDENT PRESSURE

To allow for a velocity dependent aerodynamic pressure, subroutine KINPUT is altered to read in E.6 cards that contain the specific heat ratio, the speed of sound, and the absolute pressure for the altitude which the simulation is to represent along with the definitions of two segments. The aerodynamic pressure will depend on the velocity of the first segment with respect to the second segment.

The aerodynamic pressure, $\overline{FT}$, used in WINDY to determine the aerodynamic forces, is calculated from the definition of dynamic pressure:

$$\overline{FT} = (1/2)kP_a\ (\overline{V}/c)^2$$

where k is the ratio of specific heats

c is the speed of sound

P_a is the absolute pressure

$\overline{V}$ is the velocity of the first segment with respect to the reference segment

Note that $\overline{FT}$ is a pressure and is multiplied by a wetted area in subroutine WINDY to determine the wind force applied to a segment. $\overline{FT}$ can be defined as time dependent using the same input cards as before, or as velocity dependent by specifying a specific heat ratio. How $\overline{FT}$ is applied to a segment in subroutine WINDY, has no functional dependence on the method used to define $\overline{FT}$.

2.2 DRAG COEFFICIENT

Time dependent drag coefficient functions can be defined as wind force functions on the E.6 cards. They follow the same format as the time dependent wind force functions, although the drag coefficient is a scalar quantity rather than a vector. Before the aerodynamic pressure is used in WINDY, it is multiplied by the drag coefficient, C_D.

$$\overline{FT}_{new} = C_D \, \overline{FT}$$

This can be used to simulate the effects of the drogue chute opening or other events that effect the drag. If there is no drag coefficient defined, the default value is 1.0.

2.3 BLOCKED WIND

To allow for blocking of the wind, a second method of applying the wind force has been added to subroutine WINDY. This involves projecting the ellipsoid, to which the aerodynamic force is being applied, as an ellipse to define the wetted area. Then this ellipse is divided into incremental areas, whose center points are checked for penetration of the wind plane and for blockage by other segments. Each area that passes these tests has the wind force applied at its center point. This allows for overlapping and connected segments. Since this new grid method can increase run time significantly, the original method can still be used for any or all of the segments to which a wind force is being applied, without any changes to previous input decks.

Subroutine WINDY contains the major changes that incorporate this new method for applying the aerodynamic forces. Much of the analysis needed for this method is based on the derivations developed for the VIEW program (Ref. 6). In WINDY, after checking if there is any penetration of the segment through the wind plane, and getting the wind pressure from the wind force functions, the program chooses a method for the wind force calculations depending on the input. The original method uses a calculated area of the ellipsoid, while the new method allows for blocking of the wind by other segments, by using a grid to determine the area.

2.3.1 Project Ellipsoid

For the grid method, the first step is to set up a coordinate system associated with the wind. This viewpoint coordinate system is located at an assumed origin of the wind with it's z-axis directed towards the origin of the inertial coordinate system.

Define,

$\overline{FT}$ as the wind force vector (inertial coord.)

$\overline{VP}$ as the origin of the viewpoint coordinate system (inertial coord.), which is set equal to $-10000\overline{FT}$.

$\underline{DVP}$ as the direction cosine matrix for transformation of vector components from the inertial to the viewpoint coordinate system.

The $\underline{DVP}$ transformation is chosen such that the X-axis of the viewpoint coordinate system is parallel to the X-Y plane of the inertial coordinate system. $\underline{DVP}$ can be calculated as follows:

Let

$\hat{ft} = \overline{FT} / |\overline{FT}|$, which is the unit wind force vector

form

$XNORM = \sqrt{ft_1{}^2 + ft_2{}^2}$, which is the projected length of $\hat{ft}$ on the X-Y plane.

Let

$$\hat{Z}_{vp} = \hat{ft}$$

then it can be shown that

$\hat{X}_{vp} = (ft_2\hat{i} - ft_1\hat{j})/XNORM$, is a unit vector normal to $\hat{Z}_{vp}$ and parallel to the X-Y plane of the inertial coordinate system.

The third unit vector can then be obtained using the vector product

$$\hat{Y}_{vp} = \hat{Z}_{vp} \times \hat{X}_{vp}$$

The transformation matrix is then formed by placing these unit vectors in row form

$$\underline{DVP} = \begin{bmatrix} ft_2/XNORM & -ft_1/XNORM & 0 \\ ft_1 ft_3/XNORM & ft_2 ft_3/XNORM & -XNORM \\ ft_1 & ft_2 & ft_3 \end{bmatrix}$$

The contact ellipsoid is projected onto a plane parallel to the X-Y plane of the viewpoint coordinate system. Since the viewpoint is far away from the ellipsoid and the Z-axis is nearly directed at the ellipsoid, the projection is assumed to be elliptical. To solve for an ellipse matrix, three radial vectors of the ellipsoid, pointing to a surface point that forms the contour of the projected shadow, must be determined. $\underline{BD}$(7-15,M) is the matrix that defines the surface points of ellipsoid M with respect to its principal axes. First, this ellipsoid matrix, $\underline{BD}$(7-15,M), is transformed to the viewpoint coordinate system and designated as $\underline{AM}$(3,3). This is accomplished by:

$$\underline{AM} = \underline{DVP}\ \underline{D}^T\ \underline{BD}\ \underline{D}\ \underline{DVP}^T$$

where $\underline{D}$ is the direction cosine matrix that transforms from the inertial to the ellipsoid principal coordinate system.

In order to define the projected ellipse, three vectors are chosen which lie respectively in the X-Z plane, the Y-Z plane, and the (X=Y)-Z plane of a coordinate system parallel to the viewpoint system but with its origin at the ellipsoid center. These vectors are shown in Fig. 1, and have components

$$\bar{R}_1 = \begin{pmatrix} R_{1X} \\ 0 \\ R_{1Z} \end{pmatrix} \quad \bar{R}_2 = \begin{pmatrix} 0 \\ R_{2Y} \\ R_{2Z} \end{pmatrix} \quad \bar{R}_3 = \begin{pmatrix} R_{3X} \\ R_{3Y} \\ R_{3Z} \end{pmatrix}$$

with $R_{3X} = R_{3Y}$

As seen in Fig. 1 the associated vector $\bar{P}_1$ from the viewpoint to the tip of $\bar{R}_1$ is normal to the normal vector $\bar{n}_1$ for the point defined by $\bar{R}_1$ on the ellipsoid. Therefore,

$$\bar{n}_1 \cdot \bar{P}_1 = 0 \quad \text{and} \quad \bar{n}_1 = \underline{AM}\,\bar{R}_1 \qquad 1)\ \&\ 2)$$

Combining equations 1 and 2,

$$\underline{AM}\,\bar{R}_1 \cdot \bar{P}_1 = \bar{P}_1 \cdot \underline{AM}\,\bar{R}_1 = 0 \qquad 3)$$

Also from the figure,

$$\bar{P}_1 = \overline{SM} + \bar{R}_1 \qquad 4)$$

Substituting in eq. 3 for $\bar{P}_1$ from eq. 4,

$$(\overline{SM} + \bar{R}_1)^T\,\underline{AM}\,\bar{R}_1 = 0 = \overline{SM}^T\,\underline{AM}\,\bar{R}_1 + \bar{R}^T{}_1\,\underline{AM}\,\bar{R}_1 \qquad 5)$$

$\bar{R}^T{}_1\,\underline{AM}\,\bar{R}_1 = 1$ from the definition of an ellipsoid. 6)

then,

$$\overline{SM}^T\,\underline{AM}\,\bar{R}_1 = -1 \qquad 7)$$

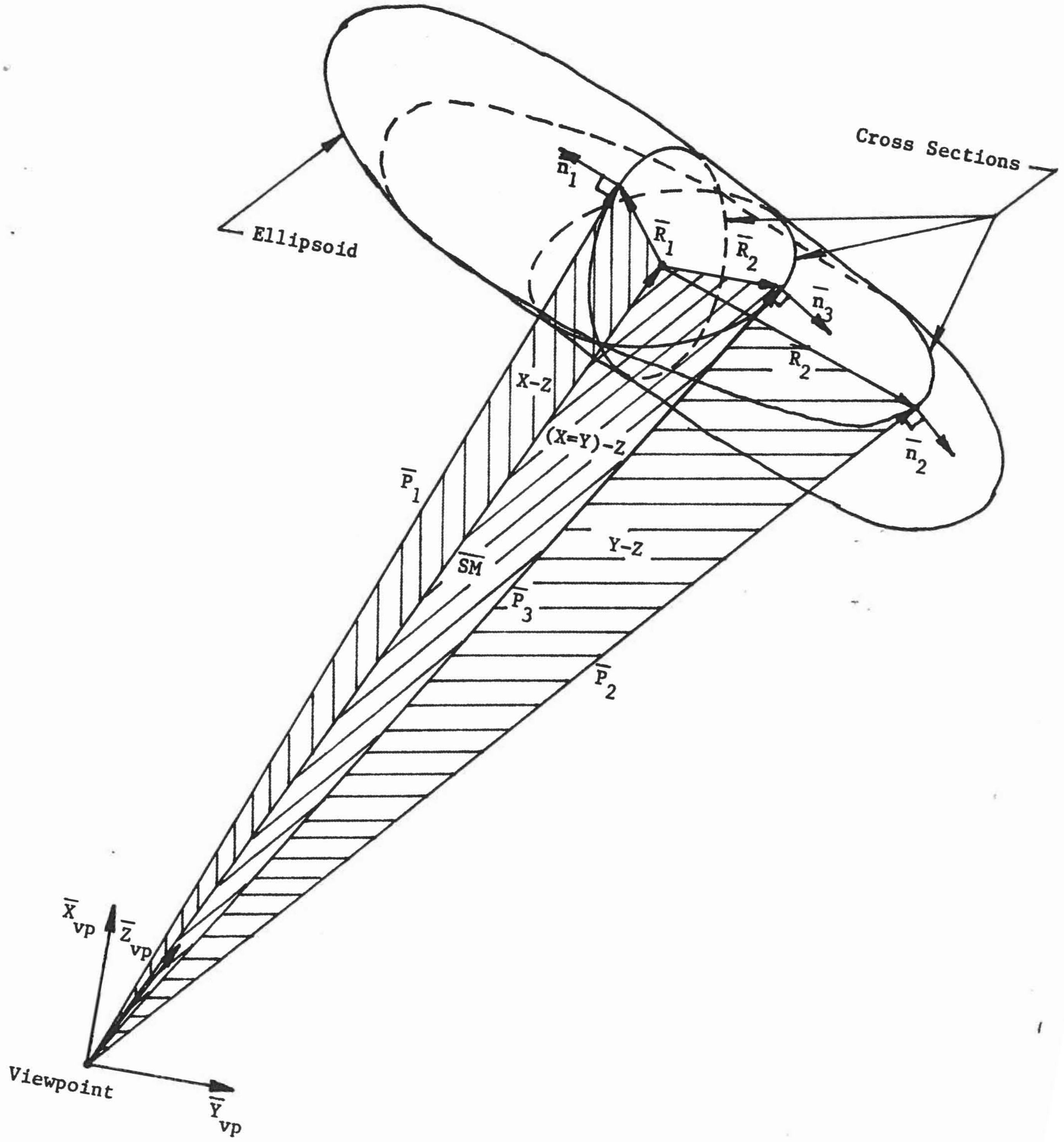

Figure 1 Three Radial Vectors

Subroutine SOLVR solves this equation for the components of $\bar{R}_1$. The same procedure is used to solve for the other components of $\bar{R}_2$ and $\bar{R}_3$.

To solve for $\bar{R}_1$, $\bar{R}_2$, and $\bar{R}_3$, we will respectively treat these as Case No. 1, Case No. 2, and Case No. 3.

For all three cases expanding eq. 7

$$(SM_X,\ SM_Y,\ SM_Z) \begin{bmatrix} AM_{11} & AM_{12} & AM_{13} \\ AM_{21} & AM_{22} & AM_{23} \\ AM_{31} & AM_{32} & AM_{33} \end{bmatrix} \qquad \bar{R} = -1$$

or

$$(SM_XAM_{11} + SM_YAM_{21} + SM_ZAM_{31},\ SM_XAM_{12} + SM_YAM_{22} + SM_ZAM_{32},$$
$$SM_XAM_{13} + SM_YAM_{23} + SM_ZAM_{33})\ \bar{R} = -1$$

This can be further reduced to

$$(A1SM_X + A2SM_Y + A3SM_Z)\ R_{XorY} + (A4SM_X + A5SM_Y + A6SM_Z)\ R_Z = -1$$

where

<u>Case No. 1</u>	<u>Case No. 2</u>	<u>Case No. 3</u>
$A1 = AM_{11}$	$A1 = AM_{12}$	$A1 = AM_{11} + AM_{12}$
$A2 = AM_{21}$	$A2 = AM_{22}$	$A2 = AM_{21} + AM_{22}$
$A3 = AM_{31}$	$A3 = AM_{32}$	$A3 = AM_{31} + AM_{32}$

$$\left.\begin{array}{l} A4 = AM_{13} \\ \\ A5 = AM_{23} \\ \\ A6 = AM_{33} \end{array}\right\} \text{for all cases}$$

Making the further substitutions;

$$B = A1SM_X + A2SM_Y + A3SM_Z$$

$$D = A4SM_X + A5SM_Y + A6SM_Z$$

we get

$$B\ R_{XorY} + D\ R_Z = -1.$$

Solving for R_{XorY}

$$R_{XorY} = -(D/B)R_Z \ -(1/B). \qquad (8)$$

Now the $\bar{R}$ vectors can be written

$$\bar{R}_1 = \begin{bmatrix} -(D/B)R_Z \ -1/B \\ 0 \\ R_Z \end{bmatrix}$$

$$\bar{R}_2 = \begin{bmatrix} 0 \\ -(D/B)R_Z \ -1/B \\ R_Z \end{bmatrix}$$

$$\bar{R}_3 = \begin{bmatrix} -(D/B)R_Z \ -1/B \\ -(D/B)R_Z \ -1/B \\ R_Z \end{bmatrix}$$

Substituting $\overline{R}$ into eq. 6 and expanding,

$$A7\ R_{XorY}^2 + 2\ A8\ R_{XorY}\ R_Z + A6\ R_Z^2 = 1 \qquad (9)$$

where

Case No. 1	Case No. 2	Case No. 3
$A6 = AM_{33}$	$A6 = AM_{33}$	$A6 = AM_{33}$
$A7 = AM_{11}$	$A7 = AM_{22}$	$A7 = AM_{11} + 2AM_{12} + AM_{22}$
$A8 = AM_{13}$	$A8 = AM_{23}$	$A8 = AM_{13} + AM_{23}$

Substituting eq. 8 into eq. 9,

$$A7[-(D/B)R_Z - 1/B]^2 + 2A8R_Z[-(D/B)R_Z-1/B] + A6R_Z^2 = 1$$

Expanding and combining like terms,

$$[A7(D/B)^2+A6-2A8(D/B]R_Z^2 + [2A7(D/B^2)-2A8(1/B)]R_Z + A7/B^2-1 = 0$$

Therefore,

$$R_Z = \frac{-T2 \pm \sqrt{T2^2 - 4T1T3}}{2T1} \quad \text{and } R_{XorY} = -(D/B)R_Z - 1/B$$

where

$$T1 = A7(D/B)^2 + A6 -2A8(D/B)$$

$$T2 = 2A7(D/B^2) - 2A8(1/B)$$

$$T3 = A7(1/B^2) - 1.$$

WINDY calls subroutine SOLVR which requires as input variables the values of A1, A2, A3, A4, A5, A6, A7, A8, and $\overline{SM}$ for each case. Subroutine SOLVR returns corresponding values for $R_{X or Y}$ and R_Z. The three $\bar{R}$ vectors obtained satisfy the three dimensional ellipsoid, and lie in the appropriate planes.

These $\bar{R}$ vectors are then projected as if the viewpoint was an infinite distance away. Therefore, the X and Y components of $\bar{R}$ are the two-dimensional projected vectors, $\overline{R2}$.

$$R2_X = R_X \qquad\qquad R2_Y = R_Y$$

The equation for the ellipse is,

$$\overline{R2}^T \underline{AS}\ \overline{R2} = 1 \quad \text{where} \quad \underline{AS} = \begin{bmatrix} AS_{11} & AS_{12} \\ AS_{21} & AS_{22} \end{bmatrix}$$

Since there are three $\overline{R2}$ vectors and three independent components of $\underline{AS}$, $\underline{AS}$ can be obtained by solving three equations simultaneously which is done in subroutine SOLVA.

2.3.2 Set-up Grid Pattern

To set up the grid pattern for the ellipse, the major and minor axis vectors are needed. These vectors are found by solving for the eigenvalues of the ellipse matrix, $\underline{AS}$, by imposing the condition

$$\underline{AS}\ \bar{R} = \lambda \bar{R}.$$

This condition is true only for vectors that represent the major and minor axes of the ellipse.

$$\begin{bmatrix} AS_{11} & AS_{12} \\ AS_{12} & AS_{22} \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \end{bmatrix} = \begin{bmatrix} \lambda R_X \\ \lambda R_Y \end{bmatrix}.$$

Or,

$$\begin{vmatrix} AS_{11}-\lambda & AS_{12} \\ AS_{12} & AS_{22}-\lambda \end{vmatrix} = 0 \qquad (10)$$

Therefore,

$$\lambda = \frac{AS_{11} + AS_{22} \pm \sqrt{(AS_{11} + AS_{22})^2 - 4\ (AS_{11}\ AS_{22} - A_{12}{}^2)^{1/2}}}{2}$$

With the eigen vectors,

$$\bar{R}_1 = \frac{1}{\sqrt{\lambda 1}} \begin{bmatrix} AS_{12} \\ \lambda 1\ -AS_{11} \end{bmatrix} = \begin{bmatrix} R_X \\ R_Y \end{bmatrix}$$

$$\bar{R}_2 = \frac{1}{\sqrt{\lambda 2}} \begin{bmatrix} \lambda 2\ -AS_{22} \\ AS_{12} \end{bmatrix} = \begin{bmatrix} R_X \\ R_Y \end{bmatrix} \qquad (11)$$

These are the major and minor axes vectors of the ellipse.

2.3.3 Three Checks

With the major and minor axes of the ellipse found, a grid is laid over the projected ellipse (Fig. 2) and each corner point of the grid is checked to see if it is in the ellipse, through the wind plane or not behind a blocking segment. If all are true, then a wind force is applied to the incremental area (AREA) shown shaded in Fig. 2.

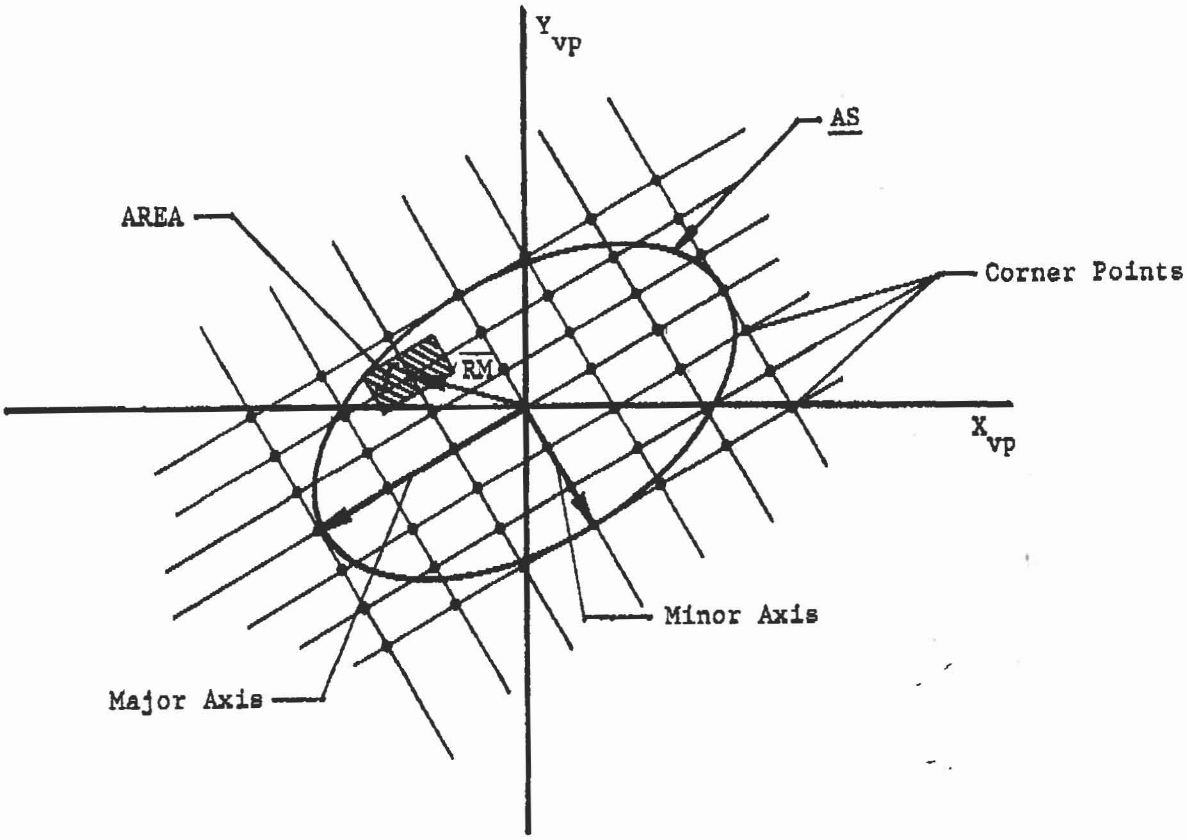

Figure 2 Grid Overlay

2.3.3.1 Check if Corner Point Is Within Projected Ellipse. $\overline{RM}$ is the two dimensional vector to a corner point. If $\overline{RM}^T \underline{AS}\ \overline{RM} \leq 1$, then $\overline{RM}$ is within the ellipse.

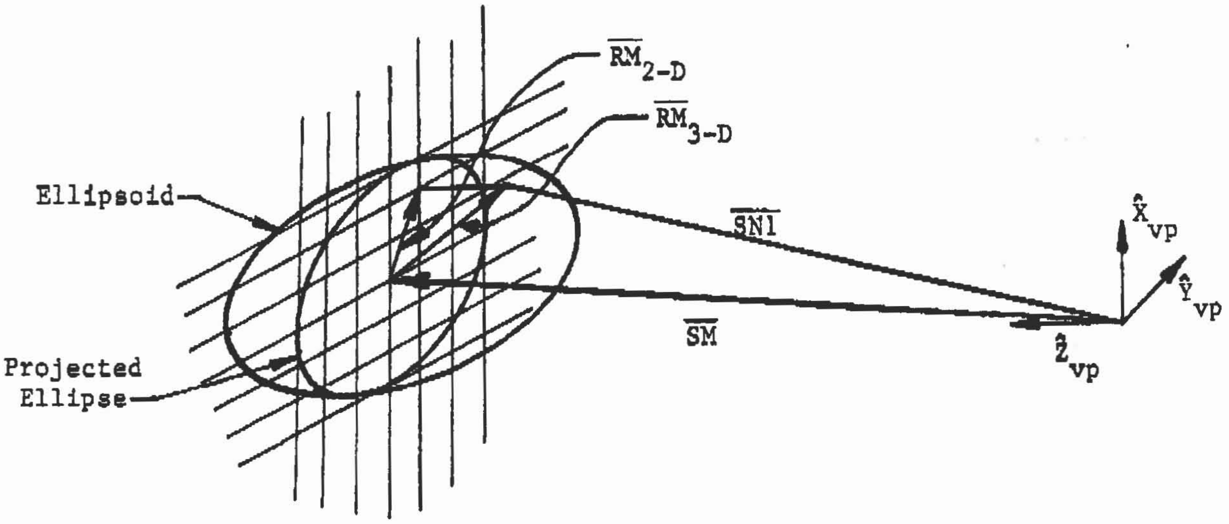

Figure 3 Three Dimensional Location of $\overline{RM}$

For the next two checks, the three dimensional location of $\overrightarrow{RM}$ on the ellipsoid is needed. See Fig. 3. The Z component of the three dimensional vector, $\overrightarrow{RM}$, is found by solving the ellipsoid equation,

$$(RM_X,\ RM_Y,\ RM_Z) \begin{bmatrix} AM_{11} & AM_{12} & AM_{13} \\ AM_{12} & AM_{22} & AM_{23} \\ AM_{13} & AM_{23} & AM_{33} \end{bmatrix} \begin{bmatrix} RM_X \\ RM_Y \\ RM_Z \end{bmatrix} = 1$$

Expanding and solving for RM_Z

$$RM_z = \frac{-TM_2 \pm \sqrt{TM_2^2 - 4TM_1 \quad TM_3}}{2TM_1}$$

where

$$TM_1 = AM_{33}$$

$$TM_2 = 2\ (RM_X\ AM_{13} + RM_Y\ AM_{23})$$

$$TM_3 = RM_x^2\ AM_{11} + RM_Y^2\ AM_{22} + 2RM_X\ RM_Y\ AM_{12} - 1$$

and $\overrightarrow{SN1}$ is the vector from the viewpoint to $\overrightarrow{RM}$ on the ellipsoid's surface and is given by

$$\overrightarrow{SN1} = \overrightarrow{SM} + \overrightarrow{RM}$$

with all vectors expressed in the viewpoint coordinate system.

<u>2.3.3.2 Check if $\overrightarrow{RM}$ Is Penetrating Wind Plane</u>. From Fig. 4 define

$\hat{PL}$ - normal unit vector to wind plane (in the segment coordinates to which plane is attached)

PL_4 - nearest distance from plane segment origin to plane

$\overline{XMM}$ - vector from plane segment origin to tip of $\overline{RM}$ (in the segment to which plane is attached)

BTS - component of $\overline{XMM}$ along $\hat{PL}$

$BTS = \overline{XMM} \cdot \hat{PL}$

If $BTS > PL_4$ then $\overline{RM}$ is penetrating wind plane.

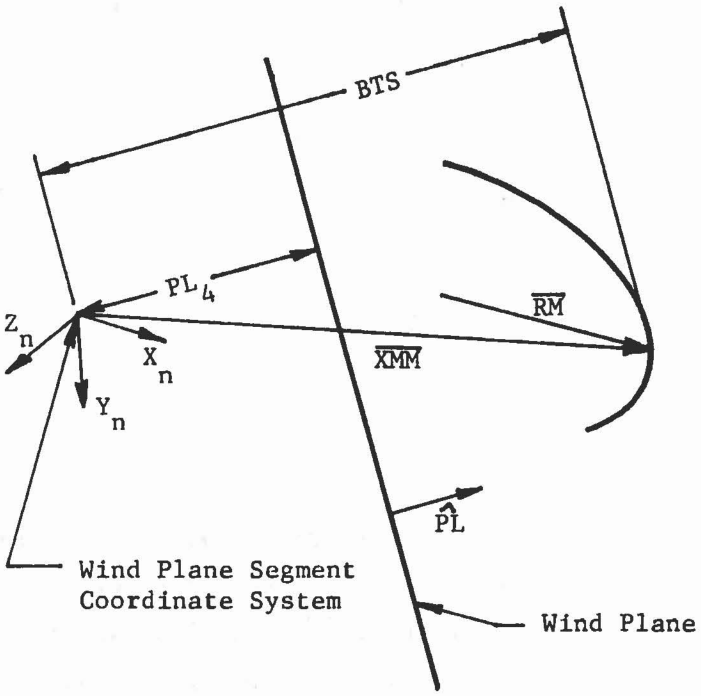

Figure 4 Ellipsoid/Plane Penetration

2.3.3.3 Check that $\overline{RM}$ Is Not Blocked By Any Of The Blocking Segments.

First, the ellipsoid matrices for the segments that may be blocking the wind must be transformed to the viewpoint coordinate system. This is done using

$$\underline{AI_i} = \underline{DVP_i}\ \underline{D_i}^T\ \underline{BD_i}\ \underline{D_i}\ \underline{DVP_i}^T$$

Also $\overline{SI_i}$ is defined as the location of the i-th blocking segment in the viewpoint coordinate system (Fig. 5).

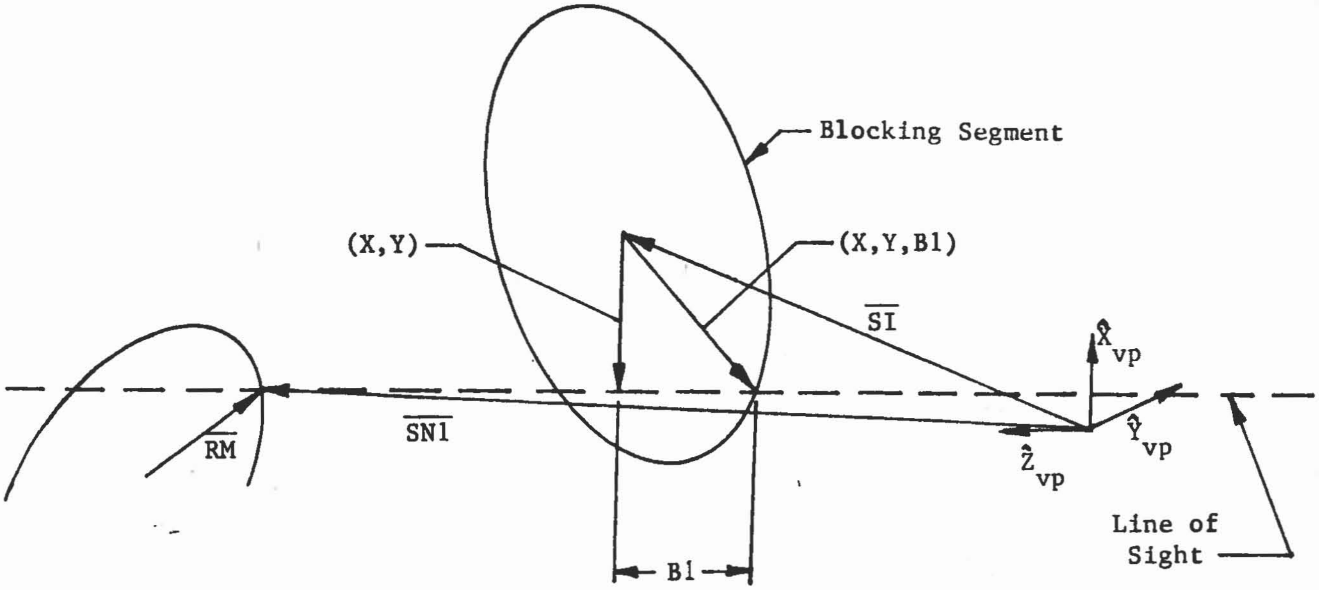

Figure 5 Check For Blocked Wind

A line of sight is defined as a line normal to the viewpoint X-Y plane through the tip of $\overline{RM}$, and is used to determine if the wind hits the blocking segment before reaching $\overline{RM}$ (Fig. 5). The two-dimensional vector, (X, Y), from the center of the blocking ellipsoid to the line of sight is used in defining the point where the line of sight enters the blocking ellipsoid.

$$X = SN1_X - SI_X$$

$$Y = SN1_Y - SI_Y$$

The Z component of the entry point, B1, is calculated in the same manner as RM_Z earlier,

$$(X,\ Y,\ B1) \begin{bmatrix} AI_{11} & AI_{12} & AI_{13} \\ AI_{12} & AI_{22} & AI_{23} \\ AI_{13} & AI_{32} & AI_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ B1 \end{bmatrix} = 1 \qquad (12)$$

If the line of sight does not pass through the blocking segment, B1 in equation 12 is a complex number. The entry point could be beyond $\overline{RM}$, therefore, the distances from the X-Y plane of the viewpoint coordinate system are compared. If $SN1_Z < SI_Z + B1$, then $\overline{RM}$ is not blocked.

Each of the possible blocking segments are checked using this method. If each of these checks are true, then a wind force is applied to the incremental area at $\overline{RM}$. Each corner point is handled the same way and the forces are totaled and added to the U1 and U2 arrays.

2.4. CHANGES TO THE PROGRAM

A new H card is now needed for the wind forces to be output as tabular time histories. The wind force applied to any segment can be output to the tabular time histories in any reference system.

This addition requires changing common block RSAVE to:

```
COMMON/RSAVE/  XSG(3,20,3), DPMI(3,3,30), LPMI(30), NSG(9),
               MSG(20,9), MCG, MCGIN(25,5), KREF(9)
```

The size of NSG, MSG, and KREF are increased to allow for the additional set of tabular time histories.

Besides RSAVE, the other changes require common block WINDFR to be modified to:

```
COMMON/WINDFR/WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),
          MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)
```

The size of MWSEG is increased to include the drag coefficient function number and the number of possible blocking segments.

New variables that are added to the COMMON block are:

WF(3, 30)   Wind force vectors applied to segments (in local reference),

MOWSEG(30, 30)

MOWSEG(2I-1, J)   Segment identification number of I-th segment that can block segment J,

MOWSEG(2I, J)   Contact ellipsoid associated with the MOWSEG (2I-1,J) segment.

In addition to the common block changes, coding changes affected a number of subroutines. Subroutines WINDY and KINPUT contain the majority of these changes and SOLVA and SOLVR are new subroutines. In Volume 3 of this report, the listing of the ATB-IV code has the labels, WINDOP or WINDROT, in column 73, of all the new or changed lines needed for these wind force options.

To use the velocity dependent pressure, drag coefficient, or blocked wind the input deck has to be modified. The input description for ATB-IV is in Volume 2 of this report and describes these modifications. Note: Previous ATB or CVS input decks require a blank card to be inserted for H.8.

3.0 ANGULAR DRIFT CORRECTION

The locked axes of the joints in the ATB Model often drift from their original position because of inherent inaccuracies due to the numerical integration process. The CHAIN subroutine was written to correct for these errors after each integration step, but the drift of the locked axes still occurred, especially during long simulations. The code modifications described here correct this drift and the sudden shifts in the joint azimuth angles.

3.1 TECHNICAL DISCUSSION

The ATB Model has four types of joints, they are:

1) Ball and Socket Joint,
2) Pin or Hinge Joint,
3) Euler Joint, and
4) Null Joint.

The Ball and Socket Joint and the Pin Joint may be locked. The Model will unlock these joints when a specified torque is exceeded. The Euler Joint has three axes which may be locked or unlocked independently thus providing eight states for the joint. The Null Joint is used to provide the option of disjoining sets of segments.

In the ATB Model constraints are imposed on the joints by computing a constraining torque. The basic equations are:

$$\underline{I}_1 \bar{\dot{w}}_1 + \underline{D}_1 \underline{P}\bar{q} = \text{other torques} \tag{13a}$$

$$\underline{I}_2 \bar{\dot{w}}_2 - \underline{D}_2 \underline{P}\bar{q} = \text{other torques} \tag{13b}$$

where $\underline{I}_1$, $\underline{I}_2$ are inertia matrices of the adjoining segments,

$\bar{\dot{w}}_1$, $\bar{\dot{w}}_2$ are the angular accelerations,

$\underline{D}_1$, $\underline{D}_2$ are direction cosine matrices,

$\underline{P}$ is a projection matrix which depends on the type of constraint, and

$\bar{q}$ is the constraint torque.

The constraint equation (which is needed to solve for $\bar{q}$) can be best derived by considering the case of a Pin Joint. In this joint the constraint is that the pin vector in one segment must coincide with the pin vector in the adjoining segment. The pin vector defines the free axis of the joint (the Pin Joint has only one free axis). The equation is:

$$\underline{D}^T{}_1 \, \bar{h}_1 = \underline{D}^T{}_2 \, \bar{h}_2 = \bar{h} \qquad (14)$$

where $\bar{h}_1$, $\bar{h}_2$ are the pin vectors (1x3 matrices) in the respective segments. These vectors are constant in the segments.

$\bar{h}$ Is the instantaneous pin vector in inertial reference,

and

$\underline{D}^T{}_1$, $\underline{D}^T{}_2$ are the transposes (inverses) of the direction cosine matrices.

Differentiating equation (14) yields an equation in the velocities $\bar{w}_1$ and $\bar{w}_2$ thus:

$$\underline{D}^T{}_1 \, (\bar{w}_1 \times \bar{h}_1) = \underline{D}^T{}_2 \, (\bar{w}_2 \times \bar{h}_2) \qquad (15)$$

Differentiating equation (15) yields an equation in the acceleration, thus:

$$\underline{D}^T{}_1 \; (\dot{\bar{w}}_1 \times \bar{h}_1) \;+\; (\bar{w}_1 \times (\bar{w}_1 \times \bar{h}_1))$$
$$= \underline{D}^T{}_2 \; (\dot{\bar{w}}_2 \times \bar{h}_2) \;+\; (\bar{w}_2 \times (\bar{w}_2 \times \bar{h}_2)) \qquad (16)$$

Equation (16) is the desired constraint equation for the accelerations, however it is only of rank 2 and we need an equation of rank 3 to solve for the torque. This is obtained by observing that the constraint torque can have no component on the pin axis, i.e.

$$\bar{h}^T\bar{q} \;=\; 0 \qquad (17)$$

Equations (16) and (17) can be combined into a single matrix equation of rank 3 by crossing equation (16) with $\bar{h}$ and adding the term $\bar{h}\bar{h}^T\bar{q}$. The resulting equation can be put in the form:

$$\underline{P} \; \underline{D}^T{}_1 \; \dot{\bar{w}}_1 \;-\; \underline{P}\,\underline{D}^T{}_2 \; \dot{\bar{w}}_2 \;+\; (\underline{I} - \underline{P})\bar{q}$$
$$= \bar{h}^T{}_2 \, \bar{w}_2 \, \underline{D}^T{}_2 \, (\bar{h}_2 \times \bar{w}_2) - \bar{h}^T{}_1 \, \bar{w}_1 \, \underline{D}^T{}_1 \, (\bar{h}_1 \times \bar{w}_1) \qquad (18)$$

The projection matrix $\underline{P}$ is given by

$\underline{P} = \underline{I} - \bar{h}\;\bar{h}^t$ where $\underline{I}$ is the identity matrix.

Equations (13) and equation (18) are the basic equationsused in the ATB Model to form the system equations which are solved for the constraint forces and torques and for the linear and angular accelerations.

Details on the form of equation (18) for a locked joint and for an Euler Joint are given in Volume 1 of Reference (4). The only differences are the form of the projection matrix $\underline{P}$ and the right hand side of equation (18). In particular for a locked joint $\underline{P}$ is the identity matrix and the right hand side of equation (18) is the zero vector (1x3 matrix).

3.2 CORRECTION OF THE DRIFT PROBLEM

The problem with drift arises since the constraints are imposed on the acceleration (equation (18)) and the values of angular velocity ($\bar{w}$'s) and angular position ($\underline{D}$'s) are obtained by numerical integration. Errors can arise because of errors in the solution of the system equations and errors due to the numerical integration process. Errors of this nature are unavoidable because of the finite precision calculation on a digital computer. Thus we may find that equation (14) and equation (15) are not satisfied to some desired degree of precision at some point in the solution process. To correct this, Subroutine CHAIN was modified.

Consider the following equations:

Let $\bar{u} = \bar{h}_2 \mathrm{~x~} (\underline{D}_2 \ \ \underline{D}^{T}{}_{1} \ \bar{h}_1)$, a vector (1x3 matrix).

If the vector $\bar{u}$ is zero then $\bar{h}_2$ and $\bar{h}_1$ are aligned. If $\bar{u}$ is not zero it is perpendicular to the pin vectors $\bar{h}_1$ and $\bar{h}_2$ and has a magnitude which is the sine of the angle between the pin vectors. This vector is used to define the rotation operator that is applied to segment 2's direction cosine matrix aligning the $\bar{h}_1$ and $\bar{h}_2$ vectors;

$$\underline{D}_2{*} \ = \ [\ c\underline{I} + \bar{u}\bar{u}^{T}/(1 + c) - (\bar{u} \mathrm{~x})]\underline{D}_2 \qquad (19)$$

where c is the cosine of the angle between the pin vectors, and ($\bar{u}$ x) is the matrix analogous to a vector cross product operation.

The velocity vector is also modified;

$$\bar{w}_2{}^* = \underline{D}_2{}^* \underline{D}^T{}_1 \bar{w}_1 + \bar{h}_2 (\bar{h}^T{}_2 \bar{w}_2 - \bar{h}^T{}_1 \bar{w}_1) \qquad (20)$$

Subroutine CHAIN (at the option of the user) modifies the direction cosine matrices $\underline{D}$ and the angular velocities $\bar{w}$ as specified in equations (19) and (20). This insures that the angular position constraint (equation (14)) and the angular velocity constraint (equation 15) are satisfied for pin joints.

3.3 CORRECTIONS TO THE PROBLEM

The ATB model was studied in detail to determine why the above procedure was not functioning properly. Two errors were found, they are:

1. The right hand side of equation (12), which is computed by Subroutine SETUP1, was being computed before the direction cosine matrices were modified.

2. Incorrect $\bar{h}$ vectors were used in Subroutine CHAIN for an Euler Joint in states 4, 5, or 6.

Error 1 was corrected by calling Subroutine CHAIN before calling Subroutine SETUP1 in Subroutine DAUX, and error 2 was corrected by correctly defining the $\bar{h}$ vectors. Also the code to correct for drift was removed from subroutine CHAIN and put into a new subroutine called DRIFT. This required that the dimensions of the HIR array and the CONST array, which are in COMMON/CEULER/, be changed. The new arrays are HIR(3,3,90) and CONST(5,30). This change was made in all the routines that included this COMMON.

Subroutines EJOINT, INITIAL, and UPDATE were modified to store the variables needed for the new drift routine.

Listings of the new DRIFT subroutines and of the changed subroutines are in Volume 3 of this report. New or changed lines are labeled with JDRIFT starting in column 73.

4.0 EDGE EFFECT OPTION

In the past, problems often developed in ATB simulations when an ellipsoid came in contact with a plane near the plane's edge. If part of the ellipsoid contacted the plane edge but the center, of the cross-sectional ellipse, containing the area cut by the plane, did not lie within the plane boundaries, then no force was applied as if no contact had occurred at all. However, as soon as this center moved within the boundaries of the plane, a full contact force was applied. The planes had to be adjusted and modified frequently to avoid instantaneous jumps in force when contact occurred at object corners. These new routines have been developed to solve this edge effect problem. In particular, use of the new edge effect option insures that a contact of a plane with an ellipsoid will not be ignored and that a smaller force will be applied when only part of the contact area is within the plane boundaries. Also an option was added allowing a force to be applied when the ellipsoid has completely penetrated the plane, if the edge effect option is not used. Another alternative for improving contact force calculations is to use a hyperellipsoid to describe the surface. This option is described in Section Seven.

4.1 NEW SUBROUTINES

4.1.1 Subroutine PLELP

Subroutine PLELP computes the point of maximum penetration of an ellipsoid associated with segment m intersecting a plane associated with segment n. Previously the point of maximum penetration was projected onto the plane. If this projection fell outside of the boundaries of the plane, the contact was ignored. A five way option has been added to the routine. The choice is made by the user by inputting an additional integer on the F.1.B - F.1.N cards. This integer is stored in the twenty third location of the TAB array associated with the contact. The options are:

TAB(NT+22) > 0, call new edge effect routine PLEDG, no force is computed for complete penetration,

= 0, use standard finite plane test, no force is computed for complete penetration,

= -1, treat plane as infinite (bypass edge test), no force is computed for complete penetration,

= -2, treat plane as infinite (bypass edge test), a force is computed for complete penetration,

< -2, use standard finite plane test, a force is computed for complete penetration.

<u>Equations used in PLELP.</u>

Let: (Fig. 6)

$\vec{Z}_m$ location of the reference point of segment m, (inertial system)

$\vec{Z}_n$ location of the reference point of segment n, (inertial system)

$\vec{O}_m$ offset of the ellipsoid, (inertial system)

$\overrightarrow{Pl}$ first reference point for the plane, (inertial system)

$\hat{T}$ unit exterior normal of the plane,

$\underline{E}$ ellipsoid matrix.

Then the equations are:

$$\vec{X}_{nc} = \vec{Z}_m + \vec{O}_m - \vec{Z}_n - \overrightarrow{Pl}\ ,$$ vector from $\overrightarrow{Pl}$ to center of ellipsoid.

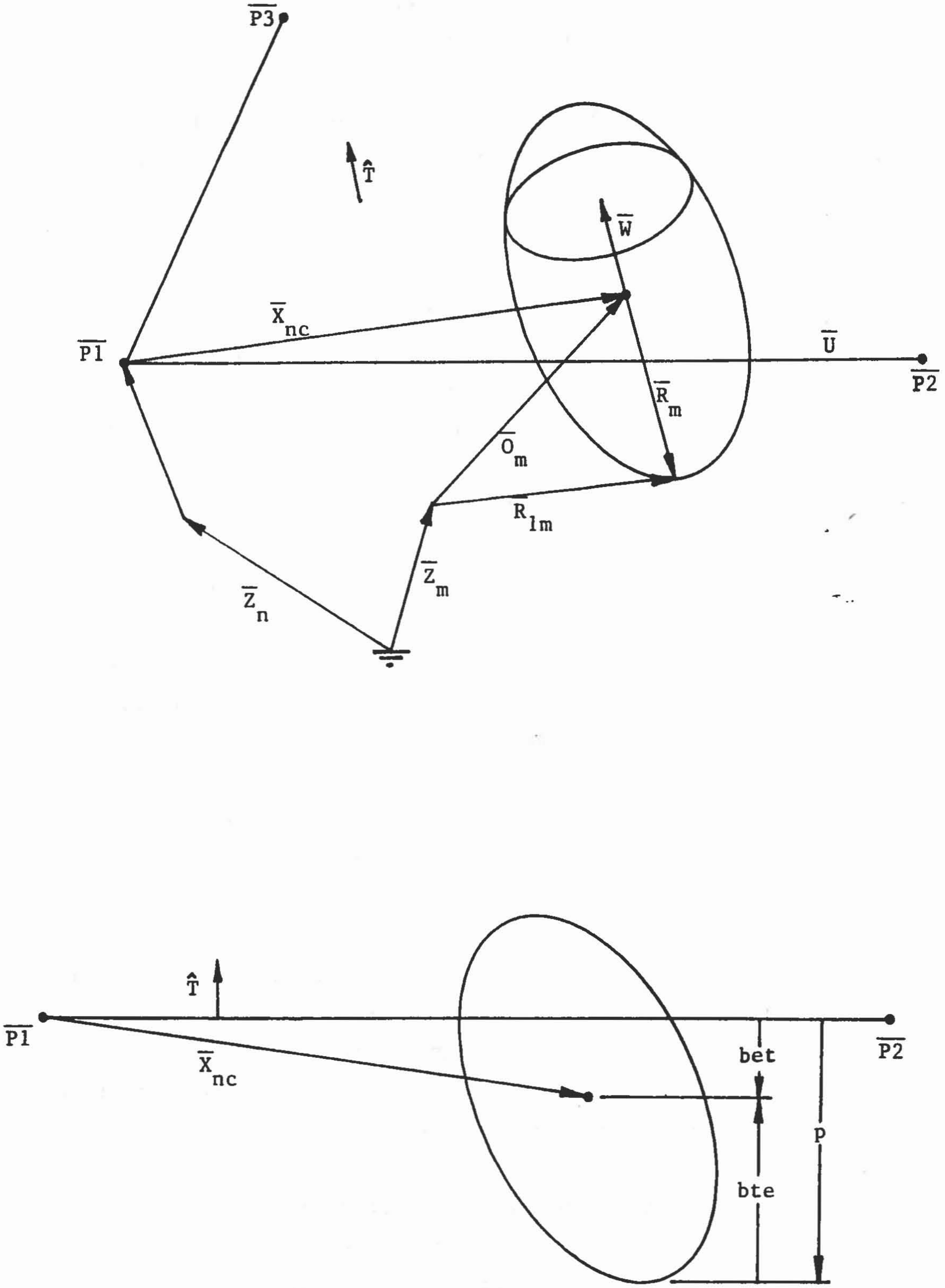

Figure 6 Plane - Ellipsoid Contact

bet = - $\hat{T}^T\bar{X}_{nc}$, distance from plane to center of ellipsoid (positive if center of ellipsoid has penetrated the plane).

bte = -sqrt($\hat{T}^T \underline{E}^{-1} \hat{T}$), the component, normal to the plane, of the vector from the point of maximum penetration to the ellipsoid center.

$\bar{R}_m = \underline{E}^{-1} \hat{T}$ / bte , vector from center of ellipsoid to point of maximum penetration. If the edge routine is used this vector is computed by PLEDG as the location of the centroid of the common area of intersection.

p = bet - bte , penetration. If the edge routine is used, the penetration is computed by PLEDG as the point on the ellipsoid below (along the -normal to the plane) the centroid.

amr = 1 - (bet/bte)^2, parameter used to determine if intersection has occurred (See Subroutine PLEDG).

Note that if amr is negative the ellipsoid doesn't intersect the plane. If amr is zero the ellipsoid is tangent to the plane. The current logic in Subroutine PLELP ignores the contact unless amr is greater than zero. This has the effect of dropping the contact once the ellipsoid has fully penetrated the plane.

$\overline{Rl}_m = \bar{R}_m + \bar{O}_m$, location of the point of maximum penetration relative to the reference point of segment m.

$\overline{R1}_n = \overline{R}_m + \overline{O}_m + \overline{Z}_m - \overline{Z}_n$, location of the point of maximum penetration relative to the reference point of segment n.

4.1.2 Subroutine PLEDG

If the ellipsoid intersects the plane, the figure of intersection will be an ellipse. Subroutine PLEDG is used to determine if this ellipse has any common area with the finite plane.

$\overline{Z} = a\overline{U} + b\overline{V}$ is a point in the plane where

$\overline{U}$, $\overline{V}$ are the vectors defining the boundary of the finite plane, and ($\overline{U} = \overline{P2} - \overline{P1}$, $\overline{V} = \overline{P3} - \overline{P1}$, where $\overline{P1}$, $\overline{P2}$, $\overline{P3}$ are the vectors defining the plane), and

a, b are scalars. Let

$\overline{W} = (bet/bte)\overline{R}_m$, be the vector from the center of the ellipsoid to the center of the cross-sectional ellipse.

d_1, d_2 is location of the center of the ellipse from the reference point $\overline{P1}$ in $\overline{U}$, $\overline{V}$ coordinates.

Then, the equation of the ellipse is

$$(\overline{Z} - \overline{W})^T \underline{E} (\overline{Z} - \overline{W}) = 1.$$

This may be written as

$$a^2e_{11} + 2abe_{12} + b^2e_{22} = \rho\ 2, \text{ where} \qquad (21)$$

$e_{11} = \overline{U}^T\underline{E}\overline{U}$, $e_{12} = \overline{U}^T\underline{E}\overline{V}$, $e_{22} = \overline{V}^T\underline{E}\overline{V}$, and $\rho\ 2 = 1 - \overline{W}^T\underline{E}\overline{W}$

(Note: $\rho 2$ is the same as amr in PLELP).

The extreme values of a and b for the ellipse are:
(Fig. 7)

a_{a2} = sqrt ($e_{22} \rho 2$/delt) , largest value of a,

b_{a2} = $- e_{12}a_{a2}/e_{22}$, value of b at a_{a2},

a_{a1} = $- a_{a2}$, smallest value of a,

b_{a1} = $-b_{a2}$, value of b at a_{a1},

b_{b2} = sqrt ($e_{11} \rho 2$/delt) , largest value of b,

a_{b2} = $-e_{12}b_{b2}/e_{11}$, value of a at b_{b2},

b_{b1} = $-b_{b2}$, smallest value of b,

a_{b1} = $-a_{b2}$, value of a at b_{b1},

where

delt = $e_{11}e_{22} - e_{12}^2$.

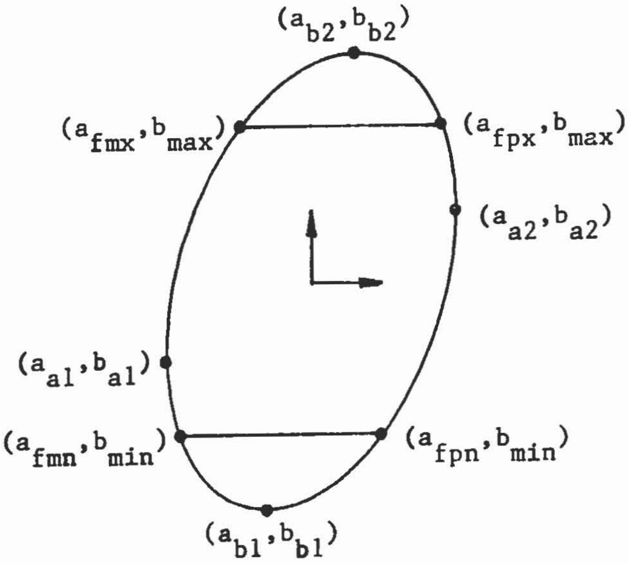

Figure 7 Extreme Values of Ellipse

Computation of the intersection of the ellipse and the finite plane (a parallelogram) is performed as follows:

Define

$a_{min} = max(a_{a1}, -d_1)$, and

$a_{max} = min(a_{a2}, 1 - d_1)$.

If a_{min} is greater or equal to a_{max}, the ellipse has no common area with the finite plane. (It lies entirely to the right of the finite plane or entirely to the left.)

Define

$b_{min} = max(b_{b1}, - d_2)$, and

$b_{max} = min(b_{b2}, 1 - d_2)$.

If b_{min} is greater or equal to b_{max}, the ellipse has no common area with the finite plane. (It lies entirely above the finite plane or entirely below.)

If b_{min} is greater than b_{b1}, the lower boundary of the plane intersects the ellipse. The corresponding values of a may be calculated from the ellipse equation 21 as

$$a_{fmn} = -b_{min}e_{12}/e_{11} - sqrt[\rho 2/e_{11} - delt(b_{min}/e_{11})^2]$$

$$a_{fpn} = -b_{min}e_{12}/e_{11} + sqrt[\rho 2/e_{11} - delt(b_{min}/e_{11})^2]$$

Similarily if b_{max} is less than b_{b2}, the corresponding values of a are

$$a_{fmx} = -b_{max}e_{12}/e_{11} - \mathrm{sqrt}[\ \rho 2/e_{11}-\mathrm{delt}(b_{max}/e_{11}{}^{2}]$$

$$a_{fpx} = -b_{max}e_{12}/e_{11} + \mathrm{sqrt}[\ \rho 2/e_{11}-\mathrm{delt}(b_{max}/e_{11})^{2}]$$

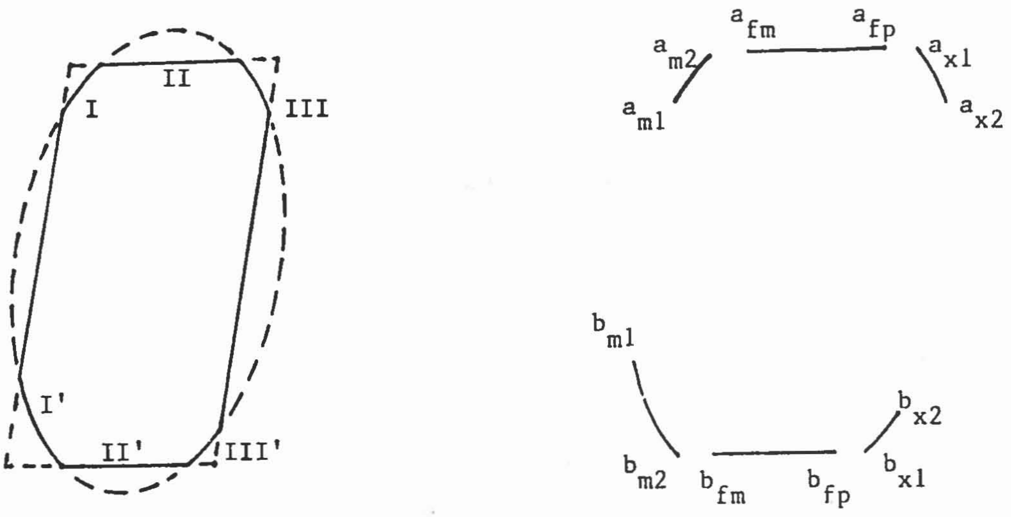

Figure 8 Common Area Boundaries

Peferring to Figure 8, the most general intersection of the plane and the ellipse consiscs of three sections comprising an upper boundary and three sections comprising a lower boundary. The computation of the area and the centroid is done in subroutine PLREA using the abscissa, a, as the independent variable. The abscissa of the sections are determined as follows:

Section

I left upper arc of ellipse:

initially $a_{m1} = a_{m2} = a_{min}$, then

if $b_{max} > b_{a1}$, $a_{m2} = \min(a_{fmx}, a_{max})$

if $b_{max} > b_{a1}$, and $b_{min} > b_{a1}$,
$a_{m1} = \max(a_{fmn}, a_{min})$

II upper straight line boundary:

$a_{fp} = \min(a_{fpx}, a_{max})$

$a_{fm} = \max(a_{fmx}, a_{min})$

III right upper arc of ellipse:

initially $a_{x1} = a_{x2} = a_{max}$, then

if $b_{max} > b_{a2}$, $a_{x1} = \max\ (a_{fpx}\ , a_{min})$

if $b_{max} > b_{a2}$, and $b_{min} > b_{a2}$,
$a_{x2} = \min\ (a_{fpn}, a_{max})$

I' left lower arc of ellipse:

initially $b_{m1} = b_{m2} = a_{min}$, then

if $b_{min} < b_{a1}$, $b_{m2} = \min\ (a_{fmn}, a_{max})$

if $b_{min} < b_{a1}$, and $b_{max} < b_{a1}$,
$b_{m1} = \max(a_{fmx}, a_{min})$

II' lower straight line boundary:

$b_{fp} = \min(a_{fpn}, a_{max})$

$b_{fm} = \max(a_{fmn}, a_{min})$

III' right lower arc of ellipse:

initially $b_{x1} = b_{x2} = a_{max}$, then

if $b_{min} < b_{a2}$, $b_{x1} = \max(a_{fpn}, a_{min})$

if $b_{min} < b_{a2}$, and $b_{max} < b_{a2}$,
$b_{x2} = \min(a_{fpx}, a_{max})$

With the above logic, if the lower abscissa of any section is greater than or equal to the upper limit the section does not exist. If no section of the upper boundary exists then there is no common area and the routine exits.

If it is determined that a common area exists, subroutine PLREA is called to compute the centroid of the common area.

Next, the vector from the center of the ellipsoid to the centroid of the common area is computed as

$\bar{R}_m = a_c\bar{U}b_c\bar{V} + \bar{W}$, where

a_c, b_c are the abscissa and the ordinate of the centroid and

$\bar{W}$ is the vector from the center of the ellipsoid to the center of the ellipse

Finally, the penetration parameter, p, is computed as the point on the ellipsoid below the centroid. This is done by using the equation of the ellipsoid

$(\bar{R}_m - p\hat{T})^T\underline{E}(\bar{R}_m - p\hat{T}) = 1$, where $\hat{T}$ is the normal to the plane.

This equation is a quadratic in the parameter p and may be readily solved with the quadratic formula. The penetration parameter p is the positive solution of this quadratic equation.

4.1.3 Subroutine PLREA

This routine computes the common area and its centroid. Since the equation of the ellipse (equation 21) is a quadratic the integration can be done in closed form. The abscissa is used as the independent variable. For a given value of the abscissa, a, the ordinate, b, is computed from equation 21 as

$b = -ae_{12}/e_{22} + \mathrm{sqrt}[r/e_{22} - d(a/e_{22})^2]$, where (22)
$r = \rho 2$ and $d = \mathrm{delt}$, are already computed by PLEDG.

The area is computed by adding the area contributed by each ellipse or straight line section. For the ellipse sections incremental areas are obtained by integrating equation 22 for each arc:

$$\int bda = h^2[t + \sin(t)\cos(t) - f\sin^2(t)/g]/(2g)$$

For the straight line portions the incremental area is ab.

The abscissa of the centroid is determined by summing the contributions from each section and then dividing by the total area. For the ellipse sections the contributions are determined by integrating equation 22 times a:

$$\int bada = h^3[-\cos^3(t) - f\sin^3(t)/g]/(3g^2)$$

For the straight line portions the contribution is $ba^2/2$.

Similarly the ordinate of the centroid is determined by summing the contributions from each section and dividing by the area. The ellipse contributions are equal to one half the integral of the square of equation 22:

$$1/2 \int b^2 da = h^3[2f\cos^3(t) + 3g\sin(t) + (f^2 - g^2)\sin^3(t)/g]/(6g^2),$$

and the straight line contributions are $ab^2/2$.

Where $\sin(t) = ag/h$,

$f = e_{12}/e_{22}$,

$g = \mathrm{sqrt}(d)/e_{22}$, and

$h = \mathrm{sqrt}(r/e_{22})$.

The routine combines these contributions to compute the centroid and the area. The true area is the area computed by this routine times the magnitude of the cross product of the $\overline{U}$ and $\overline{V}$ vectors. The true area is not computed since it is not currently used.

4.2 MODIFICATION OF OTHER ROUTINES

In the development of the edge effect routine it was convenient to have the vectors $\overline{U}$ and $\overline{V}$, which are the sides of the parallelogram of the finite plane, available. Hence the dimension of the plane array in COMMON/CNTSRF/ was changed from 17 to 24. This change was made in all routines that included this COMMON. The new format of the plane array is given in the table below.

Table 1

Format of PL Array

subscript	Description
1, 2, and 3	unit exterior normal, $\hat{T}$, to plane,
4	$\hat{T}^T\overline{P1}$
5, 6, and 7	$\overline{P1}$
8, 9, and 10	$\hat{U}p$, unit vector perpendicular to side $\overline{P3} - \overline{P1}$
11	$\hat{U}p^T\overline{P1}$
12	$\hat{U}p^T(\overline{P2} -\overline{P1})$
13, 14, and 15	$\hat{V}p$, unit vector perpendicular to side $\overline{P2} - \overline{P1}$
16	$\hat{V}p^T\overline{P1}$
17	$\hat{V}p^T(\overline{P3} - \overline{P1})$
18, 19, and 20	$\overline{P2} - \overline{P1}$, $\overline{U}$ vector
21, 22, and 23	$\overline{P3} - \overline{P1}$, $\overline{V}$ vector
24	not currently used

$\overline{P1}$, $\overline{P2}$ and $\overline{P3}$ are the points defining the plane.

Subroutine ROTATE was modified to rotate the proper components of the new PL array.

Subroutine FINPUT was modified to allow the input of the parameter used to select the edge effect option in subroutine PLELP.

Subroutine EQUILB was modified to use the new PLELP instead of using a shortened version of the subroutine.

In making changes to PLELP, a number of improvements were developed to sections of code that are in both PLELP and SEGSEG. Therefore, SEGSEG was changed to incorporate these improvements.

Listings of the changed subroutines and new PLEDG and PLREA are in Volume 3. New or changed lines have EDGE in column 73.

To use this new option changes to the F.1 cards are required. The changes are described in the input description in Volume 2.

5.0 MULTI-AXIS ANGULAR VEHICLE DISPLACEMENT

ATB-II and subsequent versions of the ATB Model allow the user to input a specified motion for predesignated segments. Thus input may be in the form of positions, velocities or accelerations. However, position information for angular orientation may be specified only for one axis, i.e. yaw, pitch or roll. In order to remove this restriction it was necessary to develop a technique that will allow general angular orientation data to be used.

5.1 MATHEMATICAL DEVELOPMENT

Given the yaw, pitch and roll angles of a body at specified points in time:

Let TM(n) be the time at point n, where n = 1, N
(these time points need not be equally spaced),
A1(n,1) be the yaw angle at point n,
A1(n,2) be the pitch angle at point n,
A1(n,3) be the roll angle at point n, and
N number of data sets.

The angular velocity $\overline{V}$ may be calculated from the quaternion product

$$\overline{V} = 2\, q^{*}\ \dot{q} \qquad (23)$$

and the angular acceleration from,

$$\overline{A} = 2q^{*}\,\ddot{q}D \qquad (24)$$

where $\overline{V}$ is a column matrix (3x1) with components V(i), i = 1,3
$\overline{A}$ is a column matrix (3x1) with components A(i), i = 1,3
q is the quaternion representing the angular orientation,
q^{*} is the conjugate of the quaternion q,
$\dot{q}$ is the time derivative of the quaternion q, and
$\ddot{q}$ is the second time derivative of the quaternion q.

The quaternion q may be represented as a column matrix with components q(i), i = 1,4. It is convenient to consider the first component q(1) as a scalar and the other three as a vector, i.e. a 3x1 matrix hence let

$c \;=\; q(1)$

$\bar{u} \;=\;$ a 3x1 matrix with components q(i), i = 2,4 and let

$e \;=\; \dot{q}(1)$

$\bar{v} \;=\;$ a 3x1 matrix with components $\dot{q}(i)$, i = 2,4

$\dot{e} \;=\; \ddot{q}(1)$

$\dot{\bar{v}} \;=\;$ a 3x1 matrix with components $\ddot{q}(i)$, i = 2,4

The quaternion q may be determined from the direction cosine matrix representing the angular orientation of the body. The direction cosine matrix is obtained from the yaw, pitch and roll angles.

$$\underline{D} = (c^2 - \bar{u}^T\bar{u})I \;+\; 2\,\bar{u}\,\bar{u}^T - 2c\,(\bar{u}\ x)$$

where $\underline{D}$ is the direction cosine matrix (3x3),

$\bar{u}^T$ is the transpose of the matrix $\bar{u}$,

$(\bar{u}\ x)$ is the matrix representing the cross product operation

and,

$\underline{I}$ is the identity matrix.

From the above equation one has

$$\underline{D}^T - \underline{D} = 4\,c\,(\,\bar{u}\ x) \text{ and}$$

$$\mathrm{trace}(\underline{D}) = 3\,c^2 - \bar{u}^T\bar{u} \;=\; 4\,c^2 - 1 \text{ since } c^2 + \bar{u}^T\bar{u} = 1.$$

Therefore
$$\begin{aligned}
Q(1) &= c &&= 0.5\ \mathrm{SQRT}(D(1,1) + D(2,2) + D(3,3) + 1),\\
Q(2) &= u(1) &&= (D(2,3) - D(3,2))/(4\ c),\\
Q(3) &= u(2) &&= (D(3,1) - D(1,3))/(4\ c),\\
Q(4) &= u(3) &&= (D(1,2) - D(2,1))/(4\ c).
\end{aligned}$$

It should be noted that the magnitude of the u(i) may be computed from the formula u(i) = SQRT((D(i,i) + 1 - c^2)/2). This may be used in the special cases where c = 0. The signs of the u(i) may be resolved by examining the signs of the off diagonal terms of $\underline{D}^T + \underline{D}$.

Assuming that these angles are continuous in time, c and $\bar{u}$ can be fitted with the Spline Routine that is already part of the ATB Model. The Spline Subroutine produces a set of functions SP(i,j,k) for a cubic spline which preserves the given values of the quaternion:

$$\begin{array}{lcll} SP(1,j,k) & = & TM(j) & ,j = 1,N,\ k = 1,4 \\ SP(2,j,k) & = & q(j,k) & ,j = 1,N,\ k = 1,4 \\ SP(3,j,k) & = & q1(j,k) & ,j = 1,N,\ k = 1,4 \\ SP(4,j,k) & = & q2(j,k) & ,j = 1,N,\ k = 1,4 \\ SP(5,j,k) & = & q3(j,k) & ,j = 1,N,\ k = 1,4 \end{array}$$

where q1, q2 and q3 are the linear, quadratic and cubic terms determined by the spline routine. These were determined such that the first and second derivatives of the piecewise cubics are continuous at the specified time points and the sum of the squares of the changes in the third derivative at these points is minimized.

Using these spline functions as interpolating functions, values of the quaternion terms may be determined at intermediate time points by the formula

$$\begin{aligned} B(t,k) \ = \ & SP(2,m,k) + X\ (SP(3,m,k) + X\ (SP(4,m,k) + \\ & X\ SP(5,m,k))) \end{aligned}$$

where B(t,k) is the interpolated value of quaternion term k at time t,

X = t - SP(1,m,k) and m is selected such that,
X is positive or zero and t - SP(1,m,k+1) is negative.

The values of the time derivatives of q are estimated using the derivative of the spline interpolating formula at each time point,

$$\dot{q}(k) = SP(3,k,3) + X\ (2.0SP(4,k,L) + 3.0\ X\ SP(5,k,L)$$

$$\ddot{q}(k) = 2.0\ SP(4,k,L) + 6.0\ X\ SP(5,k,L)$$

The values of q, $\dot{q}$, and $\ddot{q}$ are then used in equations 23 and 24 to calculate the angular velocity and acceleration at the specified time points;

$$\bar{V} = 2(c\bar{v} - e\bar{u} + \bar{v}x\bar{u})$$

In detail:

$$\begin{aligned}
V(1) &= 2\ (\ c\ v(1) - e\ u(1) + v(2)\ u(3) - v(3)\ u(2)\ ) \\
V(2) &= 2\ (\ c\ v(2) - e\ u(2) + v(3)\ u(1) - v(1)\ u(3)\ ) \\
V(3) &= 2\ (\ c\ v(3) - e\ u(3) + v(1)\ u(2) - v(2)\ u(1)\ )
\end{aligned}$$

$$\bar{A} = 2(c\dot{\bar{v}} - \dot{e}\bar{u} + \dot{\bar{v}}\ x\ \bar{u})$$

In detail

$$\begin{aligned}
A(1) &= 2(\ c\ \dot{v}(1) - \dot{e}\ u(1) + \dot{v}(2)\ u(3) - \dot{v}(3)\ u(2)) \\
A(2) &= 2(\ c\ \dot{v}(2) - \dot{e}\ u(2) + \dot{v}(3)\ u(1) - \dot{v}(1)\ u(3)) \\
A(3) &= 2(\ c\ \dot{v}(3) - \dot{e}\ u(3) + \dot{v}(1)\ u(2) - \dot{v}(2)\ u(1))
\end{aligned}$$

The angular acceleration is saved and used to prescribe the vehicle motion as it is for the other cases when the angular velocity or acceleration is input.

5.2 CHANGES TO THE PROGRAM

The above calculations are done in subroutine, VINPUT. A new subroutine, QUAT has been added to calculate a quaternion from the yaw, pitch, and roll angles. Both VINPUT and QUAT are listed in Volume 3. The changed lines in VINPUT are labelled with JTF984 in column 73.

6.0 SLIP JOINT

The ATB Model insures that joints do not pull apart by imposing a constraint force. A new feature that has been added to the model is termed a "slip joint". The slip joint allows one segment of a particular tree structure to move linearly along a prescribed axis with respect to the adjoining segment.

6.1 EQUATIONS

First, we look at the equations for a two segment model.

The linear equations of motion are:

$$\begin{aligned} m_1 \ddot{\bar{x}}_1 + \bar{f} &= \bar{u}_{11} \\ m_2 \ddot{\bar{x}}_2 - \bar{f} &= \bar{u}_{12} \end{aligned} \qquad (25)$$

subject to the constraint equation:

$$\bar{x}_1 + \underline{D}_1{}^T \bar{r}_1 = \bar{x}_2 + \underline{D}_2{}^T \bar{r}_2 \qquad (26)$$

where

m_1, m_2 are the masses of segment 1 and segment 2,

$\bar{x}_1$, $\bar{x}_2$ are the locations of the centers of gravity (inertial system),

$\ddot{\bar{x}}_1$, $\ddot{\bar{x}}_2$ are the linear accelerations (inertial system),

$\bar{r}_1$, $\bar{r}_2$ are the locations of the joint connecting the two segments (local system),

D_1^T, D_2^T are the transposes of the direction cosine matrices,

$\bar{U}_{11}$, $\bar{U}_{22}$ are the external forces acting on the segments (inertial system), and

$\bar{f}$ is the constraint force at the joint (inertial system).

For the standard joint, $\bar{r}_1$ and $\bar{r}_2$ are fixed in the local reference systems of the respective segments, i.e. the joint does not pull apart. For the slip joint, equation 26 is replaced by the equation

$$\bar{x}_1 + D_1^T (\bar{r}_1 + \bar{q}) = \bar{x}_2 + D_2^T \bar{r}_2 \qquad (27)$$

where $\bar{q}$ is displacement of the joint relative to the fixed position $\bar{r}_1$. The vector $\bar{q}$ is constrained to lie along a vector fixed in segment 1.

Thus, for the displacement of the joint

$$\bar{q} - a \bar{h} \qquad (28)$$

where a is a scalar (initialized to 0) and
$\bar{h}$ is a vector fixed in segment 1.
($\bar{h}$ is selected as the z axis of the local joint reference system in segment 1).

In order to solve the system equations, equation 27 must be doubly differentiated to obtain an equation for the accelerations. Performing this differentiation produces a term involving the acceleration of the scalar a. Since this is not known, it must be eliminated. This is done by eliminating the component of the differentiated equation 27 that is parallel to the vector $\bar{h}$ resulting in a vector equation of rank 2. To produce a complete set of equations an equation which states that the constraint force, $\bar{f}$, have no component along the free axis, $\bar{h}$. is added to the system equations. This process is similar to the process used for joint constraints as described in Volume 1 of reference 4.

6.2 IMPLEMENTATION

To define a joint as a slip joint the user is asked to provide additional information on the first card of the two card set B.3.A1 - B.3J1. On this card the user specifies:

JOINT(J) as before,
JS(J) as before,
JNT(J) as before,
IPIN(J) as before plus these new options;

= 5 slip joint with complete angular freedom
(same as IPIN = 2 for angular motion),

= 6 slip joint with pin as y-axis of joint
(same as IPIN - 1 for angular motion),
(program will use the flexural spring characteristics),

= 7 slip joint with pin as z-axis of the joint in segment JNT(J),
(program will use the torsional spring characteristics),

Negative numbers may be used to indicate that the joint is initially locked, however, if a negative number is used to indicate an initially locked slip joint, ISLIP cannot be 0 (see ISLIP below).

SR(I,2*J-1) as before,
SR(I,2*J) as before,

ISLIP new parameter,

= 1 joint is a slip joint,

= 0 standard joint or unlocked slip joint,

=-1 joint is a slip joint which may be locked or unlocked for angular motion (depending on + or - IPIN), but is locked for linear motion.

C1 new parameter, a negative number for the value of unlocking force for tension,

C2 new parameter, a positive number for the value of unlocking force for compression.

If C1 and C2 are both zero the joint will not unlock for slippage.

The slip joints allow the J + 1 joint coordinate system to move along the z-axis of the JNT(J) joint coordinate system. Tension is movement along the positive z-axis and compression along the negative z-axis.

When the slip joint is free to slip, spring and damper forces may be introduced on the slip axis by using the spring damper option in the model. The spring damper option is specified on the D.8 cards. The coordinates of the attachment points should be selected as the location of the joint in the respective segments.

6.3 CHANGES TO PROGRAM

The dimension of parameter SR in COMMON/DESCRP/ has been changed to SR(4,60). For joint J the value of 'a' (see equation 28 above) is stored in SR(4,2*J-1) and the value of the time derivative of 'a' is stored in SR (4,2*J). These values are initialized to zero by Subroutine BINPUT and are computed by Subroutine CHAIN. The change was made in all routines using this COMMON block.

A new matrix All has been added to COMMON/CMATRX/. The dimensions are All(3,3,30). Refer to section 4.7 of reference 4 to see how this matrix is used in the system equations (the matrix Bll is the transpose of All). Matrix All is computed in Subroutine SETUPl. This change to the COMMON block was made in all routines which use this COMMON block.

Subroutine BINPUT was modified to input the parameters ISLIP, Cl, and C2 (see the above section on implementation). The value of ISLIP is stored in the IEULER array and values of Cl and C2 are stored in the CONST array. These arrays are in COMMON/CEULER/.

Subroutine CHAIN was modified to compute the value of 'a' and its derivative for a slip joint. The values of SEGLP and SEGLV for segment J+l (joint J) are adjusted to insure that the slip is on the prescribed axis (axis h equation 28 above).

The DAUX subroutines were modified to introduce the matrix All into the system equations and account for values of IPIN of 5, 6, or 7.

Subroutines DHHPIN and DRIFT were modified to account for the new values of IPIN.

Subroutine PDAUX was modified to allow for the integration of the linear position and velocity of segment J+l if joint J is a slip joint and is free to slide.

Subroutine ROTATE was modified to accommodate the slip joint.

Subroutines RSTART and SEARCH were updated to account for the change in the dimension of SR and the new matrix All.

Subroutine SETUP1 was modified to compute the A11 matrix and the V1 array for the joints.

Subroutine SETUP2 was modified to account for the new values of IPIN.

Subroutine UPDATE was modified to unlock the linear motion of a slip joint based on the parameters C1 and C2 described above.

Subroutine VISPR was modified to accommodate the new values of IPIN.

7.0 HYPERELLIPSOID OPTION

In order to improve the modeling of corners and other geometries the option to use hyperellipsoids as contact surfaces rather than standard ellipsoids was added. This option was originally developed for General Motors Corporation and has been incorporated into the ATB model with their permission.

A hyperellipsoid is defined as the surface generated by the functional:

$$F(\bar{p}) = |x/a|^m + |y/b|^m + |z/c|^m \qquad (29)$$

where $\bar{p}$ is the vector from the center of the hyperellipsoid to a point on the boundary,

x, y, z are the components of $\bar{p}$,

a, b, c are the semi-axes' lengths, and

m is the power of the hyperellipsoid, an even integer.

As with an ellipsoid if

$F(\bar{p}) = 1$ the point $\bar{p}$ is on the hyperellipsoid surface, if

$F(\bar{p}) < 1$ the point $\bar{p}$ is an interior point, and if

$F(\bar{p}) > 1$ the point $\bar{p}$ is an exterior point.

If $m = 2$ the surface is an ellipsoid. For larger values of m the figure "squares off" at the corners. As m approaches infinity the figure approaches a rectangular parallelpiped with the same dimensions as the hyperellipsoid. This makes the hyperellipsoid very useful for describing contact surfaces.

To compare the hyperellipsoid shape to a parallel piped let $x = ra$, $y = rb$, and $z = rc$ in the functional, then the value for r for which this point is on the hyperellipsoid surface is given in Table 2 for various powers of m.

x = a, y = b, and z = c is a vector to the corner of the parallel piped, therefore r is the ratio between the components of the vector on the hyperellipsoid surface in the direction of the parallel piped corner and the components of the vector to the corner.

Note that for m = 128 the ellipsoid point is within 1% of the corner of the rectangular parallelpiped.

TABLE 2

Corners of a Hyperellipsoid

m	r
2	0.57735
4	0.75984
8	0.87169
16	0.93364
32	0.96625
64	0.98298
128	0.99145
256	0.99572
512	0.99786
large	1-log(3)/m = 1 - 1.0986/m

7.1 CONTACT WITH A PLANE

Subroutine PLELP computes the contact with a plane and has been modified to allow the use of hyperellipsoids. If the surface is an ellipsoid represented by the old format the original method of calculating the point of maximum penetration is used. If the surface is a hyperellipsoid the surface point whose normal is perpendicular to the plane is the point of maximum penetration and can be found by taking the gradient of the functional in equation 29;

$$\nabla F = \frac{m}{a}\left(\frac{x}{a}\right)^{m-1} \hat{i} + \frac{m}{b}\left(\frac{y}{b}\right)^{m-1} \hat{j} + \frac{m}{c}\left(\frac{z}{c}\right)^{m-1} \hat{k} = - \alpha \overline{T} \qquad (30)$$

where α is a positive scalar and
$\overline{T}$ is the outward normal to the plane.

The coordinates x, y and z of the point of maximum penetration are readily obtained from equation 30 as functions of α. α can then be computed by substituting these coordinates into equation 29. This computation is done by the double precision function HYPEN. With the value of α, the point of maximum penetration, $\overline{XH}$ is computed and a scale factor, FM, is determined. This scale factor times the vector $\overline{XH}$ will produce the vector from the center of the hyperellipsoid to the plane. If the surface is an ellipsoid, this vector will locate the center of the intersection ellipse in the plane. The quantity AMR = $1-FM^m$ is then evaluated and if it is less than or equal to zero there is no contact of the surface with the plane and no further computations are done.

If there is contact, the contact is checked to determine if it is within the boundaries of the plane and the forces are applied as described in Section 4 of this report and Vol 1 of Reference 4.

It should be noted that the roll-slide option can not be used with hyperellipsoids.

7.2 CONTACT ANOTHER HYPERELLIPSOID

Modifications were made to subroutine SEGSEG to handle the contact of two hyperellipsoids. Also two new subroutines HYEST and HYNTR were written to replace subroutine INTERS to calculate the penetration parameter and the point of force application.

To determine if penetration exists and the point of force application,three conditions must be met. First the point of force application must be the same in both segment coordinate systems:

$$\bar{q} = \underline{T}\,(\bar{p} - \bar{d}). \qquad (31)$$

Where

- $\bar{p}$ is the point in the first segment's principal coordinate system,
- $\bar{q}$ is the point in the second segment's principal coordinate system,
- $\bar{d}$ is the vector between the centers of the two surfaces in the first segment's principal coordinate system,
- $\underline{T}$ is the transformation matrix from the first to the second segment's principal coordinate system.

The normal to each surface passing through this point must be parallel and opposite in sign:

$$\nabla F(\bar{p}) = -c \nabla G(\bar{q}) \qquad (32)$$

Where

- $F(\bar{p})$ is the functional of the first hyperellipsoid at point $\bar{p}$,
- $G(\bar{q})$ is the functional of the second hyperellipsoid at point $\bar{q}$,
- ∇ is the vector gradient, and
- c is a positive scalar.

Finally, the point is chosen to be within each hyperellipsoid a distance proportional to the hyperellipsoid size:

$$F(\bar{p}) = G(\bar{q}) \qquad (33)$$

If the value of $F(\bar{p})$ (and $G(\bar{q})$) is less than 1 the figures intersect, if the value is greater than 1 no intersection occurs and if the value is 1 the figures just touch at the point $\bar{p}$.

In the original algorithm, since the figures are ellipsoids, equations (29) and (30) may be combined to form a matrix equation which can be solved for p as a function of the scalar c. A Newton-Raphson iterative method is then used to determine the value of c that allows all three equations to be satisfied (Subroutine INTERS).

In the hyperellipsoid case, equation 30 is no longer a matrix equation and so a different approach was developed. To obtain a first approximation the hyperellipsoids are treated as rectangular parallelpiped 'boxes' whose half-widths are the same as the semi-axes of the hyperellipsoids. Subroutine HYEST determines whether or not these 'boxes' intersect.

If the 'boxes' intersect, subroutine HYNTR is called to refine the estimate.

7.3 NEW SUBROUTINES

A number of new subroutines were added to the ATB model in support of the hyperellipsoid option. A description of each of these routines follows. It should be noted that in several of these routines to obtain approximations of desired quantities reference is made to a 'box'. This 'box' is the rectangular parallelpiped that is centered at the center of a hyperellipsoid. The edges are parallel to the principal axes of the hyperellipsoid and the half-widths are the same as the semi-axes of the hyperellipsoid. For large values of the power (the exponent) the hyperellipsoid almost 'fills' the box.

In order to store the variables required to define a hyperellipsoid the BD array containing the ellipsoid parameter was reformatted. The formats now used are listed in Table 3.

TABLE 3

Format of BD Array

Subscript	old format ellipsoids only	new format hyperellipsoids
1	a	-1 power of x
2	b	a
3	c	b
4	O(1)	c
5	O(2)	O(1)
6	O(3)	O(2)
7	$D^TED(1,1)$	O(3)
8	$D^TED(2,1)$	D(1,1)
9	$D^TED(3,1)$	D(2,1)
10	$D^TED(1,2)$	D(3,1)
11	$D^TED(2,2)$	D(1,2)
12	$D^TED(3,2)$	D(2,2)
13	$D^TED(1,3)$	D(3,2)
14	$D^TED(2,3)$	D(1,3)
15	$D^TED(3,3)$	D(2,3)
16	$D^TFD(1,1)$	D(3,3)
17	$D^TFD(2,1)$	$1/a^2$
18	$D^TFD(3,1)$	$1/b^2$
19	$D^TFD(1,2)$	$1/c^2$
20	$D^TFD(2,2)$	l power of x
21	$D^TFD(3,2)$	m power of y
22	$D^TFD(1,3)$	n power of z
23	$D^TFD(2,3)$	0 if equal powers
24	$D^TFD(3,3)$	not-used

where: l, m, n are the powers of the hyperellipsoid,
a, b, c are the semi axes of the (hyper)ellipsoid,
$\bar{O}$ is the offset of the ellipsoid from the c.g. of the segment,
$\underline{D}$ is the direction cosine defining the orientation of the (hyper)ellipsoid with respect to the segment principal axes,
$\underline{D}^T\underline{E}\underline{D}$ is the ellipsoid matrix, and
$\underline{D}^T\underline{F}\underline{D}$ is the inverse of the ellipsoid matrix.

7.3.1 Subroutine HYABF (B, Z, A, F)

This routine computes the hyperellipsoid functional, F, and it's derivatives. It is called by subroutines HYEST and HYNTR.

Inputs:

B BD array for hyperellipsoid containing m, a, b, c

Z array containing x, y, z

Equations:

$$F = (x/a)^m + (y/b)^m + (z/c)^m$$

$$(\nabla F)/m = \begin{pmatrix} y^{m-1}/a^m \\ y^{m-1}/b^m \\ z^{m-1}/c^m \end{pmatrix}$$

The diagonal hyperellipsoid matrix is

$$\begin{bmatrix} (x/a)^{m-2} & 0 & 0 \\ 0 & (y/b)^{m-2} & 0 \\ 0 & 0 & (z/c)^{m-2} \end{bmatrix}$$

Outputs:

A 3x3 matrix containing the diagonal elements of the hyperellipsoid matrix in the first column, the components of F in the third column.

F the value of the functional

7.3.2 Subroutine HYBND(M,Z,IV,U,C,X)

This routine computes a point on the polygon, determined from the intersection of a plane with a box, that is furthest from an interior point of the polygon in a specified direction. It is called by subroutine PLEDG.

Inputs:

M number of points in array $\bar{Z}$,

$\bar{Z}$ array determined by subroutine HYBOX,

IV pointer array determined by HYBOX,

$\bar{U}$ vector direction of interest,

$C \begin{cases} +1, \text{ use direction of } \bar{U}, \\ -1, \text{ use } -\bar{U} \text{ direction,} \end{cases}$

Output:

$\bar{X}$ point on box in direction of $C{*}\bar{U}$

Equations:

The distance of point j from the origin in the direction $C\bar{U}$ is given by

$$d = C\bar{U}^T\overline{Z(*,j)}$$

Procedure:

The points are examined and the one yielding the maximum d is stored in $\bar{X}$. If two points give the same distance $\bar{X}$ is their average value (the mid-point).

7.3.3 Subroutine HYBOX(E,T,P,N,Z,IV)

This routine computes the intersection of a plane with the edges of a rectangular box. It is called by subroutine PLEDG.

Inputs:

$\bar{E}$ array containing a, b, c, the half-widths of the box,

$\bar{T}$ the vector normal to the plane,

$\bar{P}$ the coordinates of a point on the plane.

Outputs:

N the number of paired solutions,

IV pointers to on ordered set of solutions,

ZC the coordinates of the points of intersection.

Equations:

Let the origin of coordinates be the center of the box and let the vectors $\overline{E}(i)$, i = 1, 2, 3, be parallel to the edges of the box and of length equal to the respective half-widths.

Let $\overline{Z}$ be the vector

$\overline{Z} = u\ \overline{E}(1) + v\ \overline{E}(2) + w\ \overline{E}(3)$, where u, v, w are scalars.

$\overline{Z}$ is a point in the box if $-1 < u, v, w < 1$

If $\overline{T}^T\overline{Z} = \overline{T}^T\overline{P}$ the point is in the plane.

Procedure:

The box has 6 surfaces, each of these is selected in turn and the functional $\overline{T}^T\overline{Z} - \overline{T}^T\overline{P}$ is evaluated at the four corners of the surface.

If the functional changes sign between any adjacent corners the plane intersects the edge of the box between these corners.

The intersection point is computed and stored in the array ZC.

For each surface, points are obtained in pairs, lines of intersection of zero length are ignored. The maximum number of pairs will be 12 and the minimum will be 6 if there is a true intersection of the plane with the box.

If at least 6 pairs were found the location in the array ZC of a unique path over the surface of the box is determined and stored in the array IV in such a fashion that the sequence

j = IV(k) , k = 1, 3, 5,..., N-1 will determine the path going through the points ZC(*,j).

7.3.4 Subroutine HYDAD(D,A,DAD)

This routine computes the matrix $\underline{D}^T\underline{AD}$ where $\underline{D}$ is a direction cosine matrix and $\underline{A}$ is a diagonal matrix. It is called by subroutine HYNTR.

Inputs:

$\underline{D}$ 3x3 direction cosine matrix,

$\underline{A}$ array containing diagonal elements of $\underline{A}$ (see HYABF)

Output:

$\underline{DAD}$ the 3x3 product matrix, $\underline{D}^T\underline{AD}$.

Procedure:

The computation is a straight-forward matrix multiplication.

7.3.5 Subroutine HYEST(BM,BN,TAB)

This routine is called by subroutine SEGSEG to make a preliminary estimation of intersection of two hyperellipsoids if an estimate does not exist. It is called by subroutine SEGSEG.

Input:

BM, BN BD arrays containing the data defining the hyperellipsoids m and n, The following inputs are in COMMON/TEMPVS/

$\vec{R}$ the vector from the center of hyperellipsoid m to the center of hyperellipsoid n.

<u>D12</u> the direction cosine matrix which transforms for the segment reference of n to the segment reference of m.

Output:

TAB array used as a memory, contains the same information as the V array described below if there is an intersection.

The following outputs are in COMMON/TEMPVS/

V(i) array containing the following;

i = 1 value of α (ALP), ratio of magnitudes of gradients at the intersection points,

2 value of β (BE), the expansion factor,

3,4,5 point on hyperellipsoid m,

5,6,7 point on hyperellipsoid n.

Equations:

Let $\vec{Z}$ be a vector on m and $\vec{U}$ be a vector on n where

$\vec{Z} = \vec{U} + \beta \vec{R}$; β is an expansion factor,

$\alpha = |\nabla F| / |\nabla G|$, where F and G are the hyperellipsoid functionals for $\vec{Z}$ and $\vec{U}$ respectively,

$\beta = |\vec{Z} - \vec{U}| / |\vec{R}|$.

Procedure:

The hyperellipsoids are treated as boxes and subroutine HYLPX is called to find the largest value of β , $\vec{Z}$ and $\vec{U}$ that satisfy the equation $\vec{Z} = \vec{U} + \beta \vec{R}$.

If the value of β is less than 1 there is no intersection and the routine exits storing the value of β in the TAB array. The routine also exits for a value of β equal to 1 since there can be no penetration.

If the value of β is greater than 1 there is an intersection of the boxes. In this case the points $\vec{Z}$ and $\vec{U}$ are scaled to lie on the respective hyperellipsoids, the value of α is estimated and the value of β for the scaled points is estimated. The results are stored in the TAB array.

7.3.6 Double Precision Function HYFCN(C,Z,A,P)

This function is used by subroutines HYABF, HYLIM, and HYVAL to evaluate the term HYFCN = $C(Z /A)^P$ in such a fashion as to prevent underflows. The value of A is always greater than zero and P is non-negative.

Equations:

If P = 0, HYFCN = C and if Z = 0, HYFCN = 0.

For P >0 and Z $\neq$ 0, the value of a parameter q is determined;

$$q = P(\ln|Z| - \ln A)$$

If q $\leq$ -88.5, HYFCN = 0. Otherwise HYFCN = C exp(q)

Procedure:

The equations are evaluated as above. The value of -88.5 should be adjusted to represent the smallest value which will not produce an underflow on the computer being used. (exp(-88.5) = 3.6*10$^{(-39)}$)

7.3.7 Subroutine HYLIM(A,U,B,V,C,W,Z,BD)

This routine is used to calculate the boundaries of the figure formed by the intersection of a hyperellipsoid and a plane, i.e., the point on the figure whose abscissa is a minimum or a maximum. It is called by subroutine PLEDG.

Input:

$\bar{U}$	vector defining horizontal axis (abscissa)
$\bar{V}$	vector defining vertical axis (ordinate)
C	scalar multiplier of $\bar{Z}$
$\bar{W}$	vector such that C*$\bar{W}$ is in the plane
$\bar{Z}$	estimate of desired point
BD	array containing parameters of hyperellipsoid

Output:

A scalar multiplier of $\bar{U}$
B scalar multiplier of $\bar{V}$
$\bar{Z}$ desired point

Equations:

$\bar{Z} = A\bar{U} + B\bar{V} + C\bar{W}$; origin is center of hyperellipsoid,

$\bar{T} = \bar{U} x \bar{V}$; vector cross product, a vector perpendicular to plane.

Constraints equations;

$\bar{Z}^T \underline{E} \bar{Z} = 1$; functional equation of hyperellipsóid,

$\bar{V}^T \underline{E} \bar{Z} = 0$; boundary constraint, normal perpendicular to ordinate,

$\bar{T}^T \bar{Z} = C\bar{T}^T \bar{W}$; $\bar{Z}$ must lie in plane,

Procedure:

A first estimate was obtained before calling this routine by a call to HYBND. This estimate is the value of $\bar{Z}$ on entry.

First order perturbation equations are used to refine the value of $\bar{Z}$. These equations are:

$$\bar{Z}^T \underline{E} \bar{D} = (1 - \bar{Z}^T \underline{E} \bar{Z})/m$$

$$\bar{V}^T \underline{E} \bar{D} = -\bar{V}^T \underline{E} \bar{Z}/(m-1)$$

$$\bar{T}^T \bar{D} = C\bar{T}^T \bar{W} - \bar{T}^T \bar{D}$$

where $\bar{D}$ is the perturbation of $\bar{Z}$ and m is the power of the hyperellipsoid.

The equations are solved for $\bar{D}$ and $\bar{Z}$ is updated as $\bar{Z} = \bar{Z} + \bar{D}$.

The process is iterated until the perturbations are small when compared to $\bar{Z}$.

When convergence is obtained the values of A and B are computed.

7.3.8 Subroutine HYLPR(J1,J2,ID,C,S,E,T)

This routine is a simplex method for solving a linear programming problem. It is called by subroutine HYLPX.

Input:

J1 index of first column to search,
J2 index of last column to search,
ID pointer array to identify columns,
$\bar{C}$ cost vector,
$\underline{S}$ constraint array,
E temporary storage for pivot column,

Output:

$\bar{T}$ vector indicating final costs of each column,
$\underline{S}$ right hand column contains solutions obtained,
ID pointer to identify columns.

Equations:

$\underline{S}$ is a matrix whose rows are the constraint equations and whose columns are the coefficients of a particular variable in these constraint equations. The last column of $\underline{S}$ is the constant term in the equations.

$$T(j) = \overline{C1}^T\overline{S(*,j)} - C(k), \; j = J1, J2, \; k = ID(j)$$

where $\overline{C1}$ is the cost vector of the current solution.

Procedure:

The simplex algorithm is used. The values of T(j) are computed if any T(j) is positive, variable j is entered into the solution, replacing the variable whose elimination will reduce the cost. Pointers to the current solution variables are kept in the ID array which is updated.

The process is iterated until all T(j) are non-positive.

The J1 = J2 the variable identified with column J1 is forced into the solution and no iteration is performed.

7.3.9 Subroutine HYLPX(BM,BN)

This program is called by subroutine HYEST to solve for the estimate of the points of intersection of two hyperellipsoids.

Input:

BM,BN BD arrays containing the parameters of hyperellipsoids m and n.

The following inputs are in COMMON/TEMPVS/

$\bar{R}$ the vector from the center of m to the center of n.

$\underline{D}$ the direction cosine matrix which transforms from the n's reference system to m's reference system.

Output: (in COMMON/TEMPVS/)

V(i) an array containing the following;

i = 1, 2, 3	point on the box enclosing hyperellipsoid m,
i = 4, 5, 6	point on the box enclosing hyperellipsoid n,
i = 7	the expansion factor.

Constraint Equations:

$\bar{Z} - \bar{V} - \beta \bar{R} = 0$

$|Z(i)| \leq$ a(i), where a(i) are the semi-axes of m, i = 1,3,

$|V(i)| \leq$ b(i), where b(i) are the semi-axes of n, i = 1,3,

Procedure:

The array S representing the constraint equations is computed. The value of β is assigned a cost of -1, the values of $\bar{Z}$ and $\bar{V}$ and their associated slack vectors are assigned costs of 0. Subroutine HYLPR is called to solve for the values of $\bar{Z}$ and $\bar{V}$ which produce the maximum value of β.

If after the initial call to HYLPR the associated cost vector indicates that there is more than one solution HYLPR is recalled to find all solutions and the results are averaged.

See the descriptions of subroutines HYEST and HYLPR for more details.

7.3.10 Subroutine HYNTR(BM,BN,TAB)

This subroutine is called by subroutine SEGSEG to determine the points on intersecting hyperellipsoids that are used to determine the penetration (if any) of these figures.

Inputs:

BM, BN   BD arrays containing the parameters of the figures m and n,

TAB   array containing the current estimates of the desired points,

The following inputs are in COMMON/TEMPVS/

$\bar{R}$ the vector from the center of figure m to the center of figure n.

$\underline{D12}$ the direction cosine matrix which transforms a vector from the segment reference system associated with n to that of m,

Output:

TAB(i)   an array containing:

i = 1   the value of α ,

i = 2   the value of β ,

i = 3, 4, 5   the value of $\bar{Z}$, the point on m,

$i = 6, 7, 8$ the value of $\overline{V}$, the point on n.

This TAB array is offset such that $i = 1$ corresponds to a value of $i = 23$ in the TAB array used by subroutine SEGSEG.

Equations:

$\overline{Z} = \overline{V} + \beta\overline{R}$;   relation between points,

$\nabla F = -\alpha\nabla G$,   where F and G are the hyperellipsoid functionals for the m and n hyperellipsoids respectively.

$F = 1, \; G = 1$

Let $F = \overline{Z}^T\underline{A}\overline{Z}$, $G = \overline{V}^T\underline{B}\overline{V}$, $\nabla F = \underline{A}\overline{Z}$, $\nabla G = \underline{B}\overline{V}$, and let $d\overline{Z}$, $d\overline{V}$, $d\alpha$ and $d\beta$ be perturbations of $\overline{Z}$, $\overline{V}$, α and β respectively. The linearized perturbation equations are:

$$d\overline{Z} - d\overline{V} - d\beta\,\overline{R} = -\overline{Z} + \overline{V} + \beta\overline{R}$$

$$d\underline{A}\overline{Z} + \alpha\, d\underline{B}\overline{V} + d\alpha\,\underline{B}\overline{V} = -\underline{A}\overline{Z} - \alpha\underline{B}\overline{V}$$

$$\overline{V}^T\underline{B}\,d\overline{V} = 1 - \overline{V}^T\underline{B}\overline{V}$$

$$\overline{Z}^T\underline{A}\,d\overline{Z} = 1 - \overline{Z}^T\underline{A}\overline{Z}$$

Procedure:

The values stored in the TAB array on entry are used as first guesses to the variables α, β, $\overline{Z}$ and $\overline{V}$.

The perturbation equations are solved and the values updated. The procedure is iterated until the perturbations of $\overline{Z}$ are small compared to the value of $\overline{Z}$.

When convergence is determined the TAB array is updated.

7.3.11 Subroutine HYREA

This subroutine is called by subroutine PLEDG to compute an approximate area and centroid for the figure formed by the intersection of a hyperellipsoid and a plane.

All inputs and outputs are in COMMON/TEMPVS/.

Input:

AM1, AM2, AFM, AFP, AX1, AX2, BM1, BM2, BFM, BFP, BX1, BX2, AMIN, AMAX, BMIN, BMAX   coordinates of the boundaries of the figure

Output:

AREA   proportional to the area, the true area is this number times the magnitude of the cross product of the vectors used to define the abscissa and the ordinate of the coordinate system used in PLEDG.

AB   the location of the centroid in the abscissa coordinate,

BB   the location of the centroid in the ordinate coordinate,

Equations:

Consider the area below the straight line segment for the point (x1,y1) to the point (x2,y2). Then

$dx = x2 - x1$

$ar = dx(y2 + y1)$, twice the increment of area,

$ax = ar(x2 + x1) + dx(x2y2 + x1y1)$, six times the increment of the abscissa of the centroid, ay = ar(y2 + y1) - dxy2y1, six times the increment of the ordinate of the centroid,

AREA = sum of the ar divided by two,

AB = sum of the ax divided by six times the area,

BB = sum of the ay divided by six times the area,

Procedure:

The sections of the general shape are shown in Figure 9.

Tests are made for the existence of the sections and the formulae are used to compute the area and the centroid.

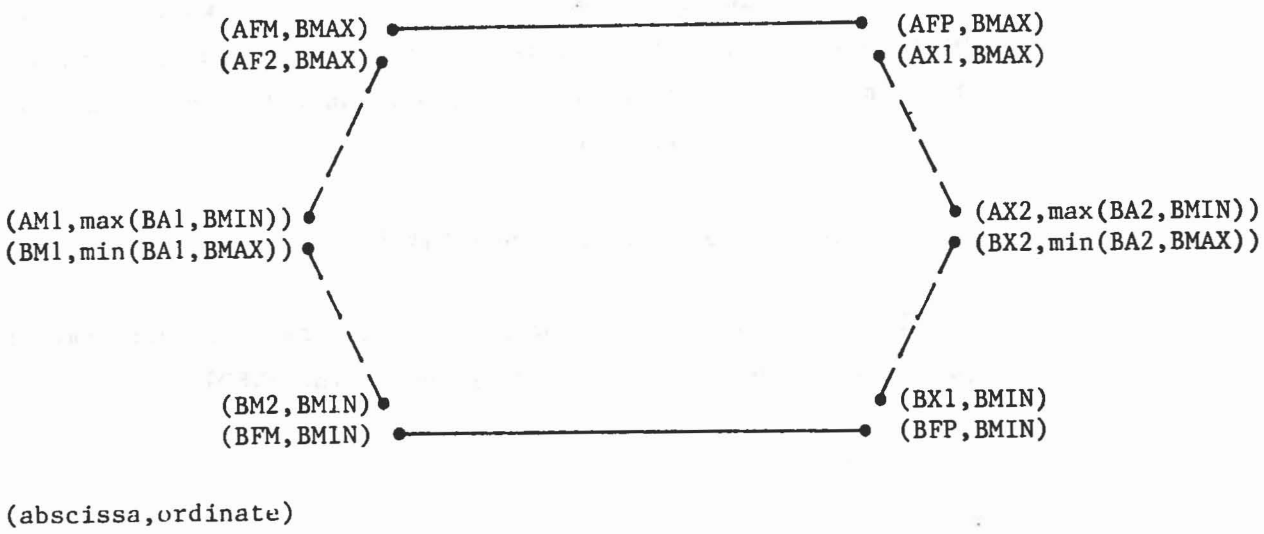

Figure 9 Hyperellipsoid Common Area Boundaries

7.3.12 Subroutine HYSOL(A,N,ND)

This routine is a modified Gauss Elimination process for solving a set of simultaneous equations. It is called by subroutine HYNTR.

Input:

$\underline{A}$ array containing the simultaneous equations,

N the number of equations,

ND the first subscript of the array $\underline{A}$.

Output:

$\underline{A}$ the reduced (solved) equations. The solutions are in column N+1 of the array.

Procedure:

Gauss - Elimination is used with the pivot always on the diagonal. No pivoting is done for a zero diagonal. This modification was necessary because the matrix $\underline{A}$ may have an all zero column (with a corresponding all zero row).

7.3.13 Subroutine HYVAL(A,U,R,BD,L)

This routine computes the point on a hyperellipsoid that lies on a particular line. It is called by subroutine PLEDG.

Input:

$\bar{U}$ vector defining line of interest,

$\bar{R}$ vector locating end point of line, BD array containing hyperellipsoid parameters,

L = 1 indicates point along $-\bar{U}$ desired,
= 2 indicates point along $+\bar{U}$ desired,

Output:

A parameter defining point on $\bar{U}$.

Equations:

$\bar{Z} = A\bar{U} + \bar{R}$, point on line $\bar{U}$,

$\bar{Z}^T\underline{B}\bar{Z} = 1$, constraint that point lies on hyperellipsoid, (hyperellipsoid functional)

Perturbation equations:

Let $f(A) = \bar{Z}^T\underline{B}\bar{Z} - 1$, and let e be a perturbation of A. Expand up to the second derivative with respect to A to get

$f(A+e) = f(A) + ef' + e^2f''/2 = 0,$

(f' is the first derivative of f, and f'' is the second at A),

$e = -f(A)/[f'/2 + \mathrm{sign}(f')[(f'/2)^2 - f(A)f''/2]^{1/2}]$

Procedure:

Subroutine HYVBX is called to determine the point on the box surrounding the hyperellipsoid in the direction specified by L.

The quadratic perturbation equations are solved and the process iterated until the functional equation satisfies a prescribed test.

7.3.14 Subroutine HYVBX(Q,S,B,M,RM)

This routine is called by subroutine HYVAL to determine the points of intersection of a vector with a box.

Input:

$\bar{Q}$ vector which intersects box,

$\bar{S}$ fixed vector from center of box to Q

B array containing dimensions of box,

Output:

M index identifying solutions, M = 2 if there is an intersection, M = 0 for none.

RM array containing the two points of intersection if they exist.

Equations:

$\bar{Z} = r\bar{Q} + \bar{S}$, general point on $\bar{Q}$,

Procedure:

If $\bar{Z}$ is a point on a face of the box some component of $\bar{Z}$ must equal the dimension of the box. Let

B(i) be + or - the half widths of the box,

Z(i), Q(i) and S(i) be the corresponding components of the vectors $\bar{Z}$, $\bar{Q}$, and $\bar{S}$ respectively.

If $|(B(i) - S(i))Q(i) + S(j)Q(i)| < |B(j)Q(i)|$ then

the point determined by $r = (B(i) - S(i))/Q(i)$ will be a desired point. The inequality is tested for all combinations of the indices and the unique solutions for r are saved in the array RM. Before exiting the RM's are ordered such that RM(1) is the smallest algebraic solution and RM(2) is the largest.

7.3.15 Function HYPEN(BD,E,V)

This routine is called by subroutine PLELP to compute the value of ALP which is used in the computation of the intersection of a hyperellipsoid with a plane. The powers of the hyperellipsoid functional may be different.

Input:

BD array containing hyperellipsoid information,

E exponents used in computation,

V other constants used in computation.

Output:

HYPEN the value of ALP

Equations:

$F = |x/a|^k + |y/b|^l + |z/c|^m - 1$, the hyperellipsoid functional. Note: for this derivation we assume k, l and m are even integers.

The gradient of F is

$(k/a)(x/a)^{k-1}$, $(l/b)(y/b)^{l-1}$, $(m/c)(z/c)^{m-1}$.

At the point of maximum penetration the gradient must be parallel to the plane vector (t_1,t_2,t_3), i.e.

$$|x/a|^{k-1} = ALP|t_1|a/k,$$

$$|y/b|^{l-1} = ALP|t_2|b/l,$$

$$|z/c|^{m-1} = ALP|t_3|c/m.$$

Let: $v_1 = [|t_1|a/k]^{e1}$, $e_1 = k/(k-1)$,

$v_2 = [|t_2|b/l]^{e2}$, $e_2 = l/(l-l)$,

$v_3 = [|t_3|c/m]^{e3}$, $e_3 = m/(m-1)$.

where $V = (v_1,v_2,v_3)$, and $E = (e_1,e_2,e_3)$ are computed by

PLELP before the call to HYPEN.

The parameter ALP must be determined such that the functional equation of the hyperellipsoid is satisfied. This equation may be written as:

$$F(ALP) = v_1ALP^{e_1} + v_2ALP^{e_2} + v_3ALP^{e_3} - 1.$$

Procedure:

The exponent, e_m, associated with the maximum value of v's is determined. The values of ALP is first estimated as:

$$ALP = 1/(v_1 + v_2 + v_3)\ (1/e_m).$$

The value of F(ALP) is computed. If $|F(ALP)|$ is less than 10^{-8} the functional is assumed to be satisfied and the routine exits. If the F(ALP) is greater or equal to 10^{-8} a stepping procedure is used to modify ALP until convergence is obtained. Note: if the exponents are all equal, the first estimate of ALP should satisfy the convergence test.