

AFTT/GAE/ENY/91D-22

AD-A244 045



1

DTIC
ELECTE
JAN 8 1992
S B D

APPLICATION OF FINITE ELEMENT MODELING
AND ANALYSIS TO THE DESIGN OF
POSITIVE PRESSURE OXYGEN MASKS

THESIS

Bruce H. Bitterman, Captain, USAF

AFTT/GAE/ENY/91D-22

92-00056

Approved for public release; distribution unlimited

92 1 2 051

PII Redacted

AFIT/GA/ENG/91D-22

APPLICATION OF FINITE
ELEMENT MODELING AND ANALYSIS TO
THE DESIGN OF POSITIVE PRESSURE OXYGEN MASKS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautical Engineering

Bruce H. Bitterman, B.S.,M.A.

Captain, USAF

December 1991

Approved for public release; distribution unlimited

Preface

The purpose of this study was to develop and apply tools and techniques for finite element modeling and analysis, starting with three dimensional (surface) digitized coordinate data. This was accomplished and applied to the MBU-20/P pilot's oxygen mask. Many other applications and extensions are possible. The tools developed will be of immediate use to the Armstrong Medical Research Laboratory (AAMRL), Human Engineering Division.

I'd like to thank Ms. Jenny Whitestone of AAMRL for her assistance in providing the digitized coordinate data of the mask, and face. Thanks are due to my advisor, Major Dave Robinson (PhD) for providing feedback and guidance scoping this effort. Discussions with Dr. Palazotto were informative regarding loads and boundary conditions. I'd like to acknowledge my wife, Nessa, for her support throughout this effort.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	

Bruce Howard Bitterman



Table of Contents

Page

Preface.	ii
List of Figures	v
Abstract.	vii
1.0 Introduction	1
1.1 Problem Statement	4
1.2 Positive Pressure Breathing Masks	5
2.0 Scope	8
3.0 Computer Program Development	10
3.1 Input Data	11
3.2 Desired Output Data	13
3.3 Connectivity	14
3.4 Parametric Representation of the Data	18
3.5 Model Reduction	26
3.6 Selective Digitized Point Data Reduction	27
3.7 Selective Feature Extraction	28
4.0 Finite Element Modeling	30
4.1 Model Synthesis	30
4.2 Physical Properties	39
4.3 Material Properties	41
5.0 Finite Element Analysis.	43
5.1 Loads Cases	43
5.2 Boundary Conditions	45
6.0 Results / Conclusions	47
7.0 Recommendations	57
Appendix A: PATRAN vs I-DEAS.	60
Appendix B: Derivation of Algorithms for Bicubic Surfaces (PATRAN Neutral File Format)	63
Appendix C: Scan Limit Analysis Listing	70
Appendix D: Sample Intermediate Data File	72
Appendix E: SCANCAD Users Guide	74
Appendix F: SCANCAD Program Listing	79
Appendix G: Frame Formatted Output from I-DEAS.	93

Appendix H:	DISPLZ Program	95
Appendix I:	GAPZ Program	101
Appendix J:	RELABL Program.	106
Appendix K:	SCANCAD Variables Dictionary.	111
Appendix L:	List of Equipment	116
Appendix M:	Discussion of Gap Elements	118
Appendix N:	NEWSHAPE Program	120
Appendix O:	REAC Program	123
Bibliography	126
Vita	127

List of Figures

Figure	Page
1. MBU-20/P Pilot's Oxygen Mask and FEM Representation	3
2. Parts Breakdown of MBU-20/P	5
3. MBU-12/P Mask	6
4. Cyberware Digitizer	9
5. Parameterization of Nodes	20
6. Element Connectivities	21
7. Parametric Representation of Mask	23
8. Parameterized Head Scan	24
9. Mask Seal Region	25
10. Non-Uniform Parameterization of Head Scan.	28
11. Top Down View of Mask Nose Bridge Region	32
12. Scan and Connectivity of Seal Region	34
13. Superposition of Different Mesh Densities.	36
14. Aligning the Plane of Symmetry	37
15. Synthesized Mask and Face Model.	38
16. Gap Elements Connect Mask and Face	40
17. Net Pressure Forces	45
18. Large 'Rigid Body' Displacement of Mask.	46
19. Reaction Forces on Face.	49
20. Enforced Conforming Displacement of Seal	50
21. Stress Contours on Seal Region	51
22. Refined Mesh on Seal Region	52
23. Reaction Forces on Face with No Internal Pressure .	55
24. Reaction Forces Due to PPB Pressure	56

25. Nodes on Mask Seal and Face 97
26. Conforming Displacement of Seal 98
27. Boundary Conditions on Seal 100a
28. Gap Elements Created by GAPZ. 102
29. Change of Starting Longitude 107

Abstract

↓

This study investigated tools and techniques for performing finite element analysis of the MBU-20/P pilot's oxygen mask. The problem which was investigated was the fit of the mask to the face at Positive Pressure Breathing (PPB) pressures. Deformed geometry of the mask and reaction forces on the face were calculated to provide a measure of the fit. The MBU-20/P mask was first digitized by 3-D digitizer. A computer program was written to translate the digitized coordinate data to a finite element format for two commercial packages. The packages targeted were SDRC I-DEAS and PDA PATRAN. Other supplementary programs were written to perform some functions which would have been difficult to accomplish within the commercial codes. The tools and techniques developed are applicable to a wide class of problems, beyond the specific application to the oxygen mask analysis. They provide a capability to rapidly develop finite element models from existing prototypes for redesign, detailed analysis, or reverse engineering.

1.0 Introduction

This work was accomplished at the Air Force Institute of Technology (AFIT). The goal of this research/development effort was to provide tools and procedures for rapid prototype development of environmental protection equipment for Air Force personnel. These tools and techniques augment the standard finite element analysis programs currently available. The tools developed are suitable to a wide class of problems, beyond the specific application dealt with in this effort. They are applicable to reverse engineering and redesign. Developing models which can be analyzed and which realistically duplicate a real life problem is the first step to coming up with possible solutions. Modifications, as potential solutions, can be 'tested' and refined via computer before manufacturing. Trade studies can be effected to optimize certain performance characteristics. As with most tools, a good understanding of the assumptions that went into their development is essential to their proper application.

There were three distinct but related parts to this effort:

- a) Computer Interface Program Development
- b) Finite Element Modelling
- c) Finite Element Analysis

The main computer program which was developed is called SCANCAD. It takes in digitized coordinate data and outputs

finite element model (FEM) files for input to two popular commercial programs, PDA PATRAN and SDRC I-DEAS. SCANCAD is written in standard FORTRAN 77. Using SCANCAD and SDRC I-DEAS, a finite element model of the MBU-20/P pilot's oxygen mask (Figure 1) was developed.

Additional short programs were written as needed to perform particular functions and will be discussed later. Linear static analyses of the MBU-20/P mask connected to a face by non-linear gap elements were performed within the I-DEAS program. Approximately 2400 elements were used in the model analyzed. The modeling and analysis tools and procedures developed could serve as a starting point for trade studies leading to possible design changes of the mask. Additionally, this work provides a framework which could be expanded upon for enhanced capabilities (see Appendix A). The results of this effort will meet an Air Force need at the Armstrong Aerospace Medical Research Laboratory (AAMRL), Wright Patterson AFB, Ohio.

The SCANCAD program developed will automatically turn a digitized coordinate data file into a finite element model. There is no guarantee that the model will produce accurate results. Various factors can affect the accuracy of the finite element analysis. These include: type of elements being used, appropriateness of aspect ratios, mesh densities in the area of large gradients, material properties, loading and boundary conditions. The accuracy limits of the solution

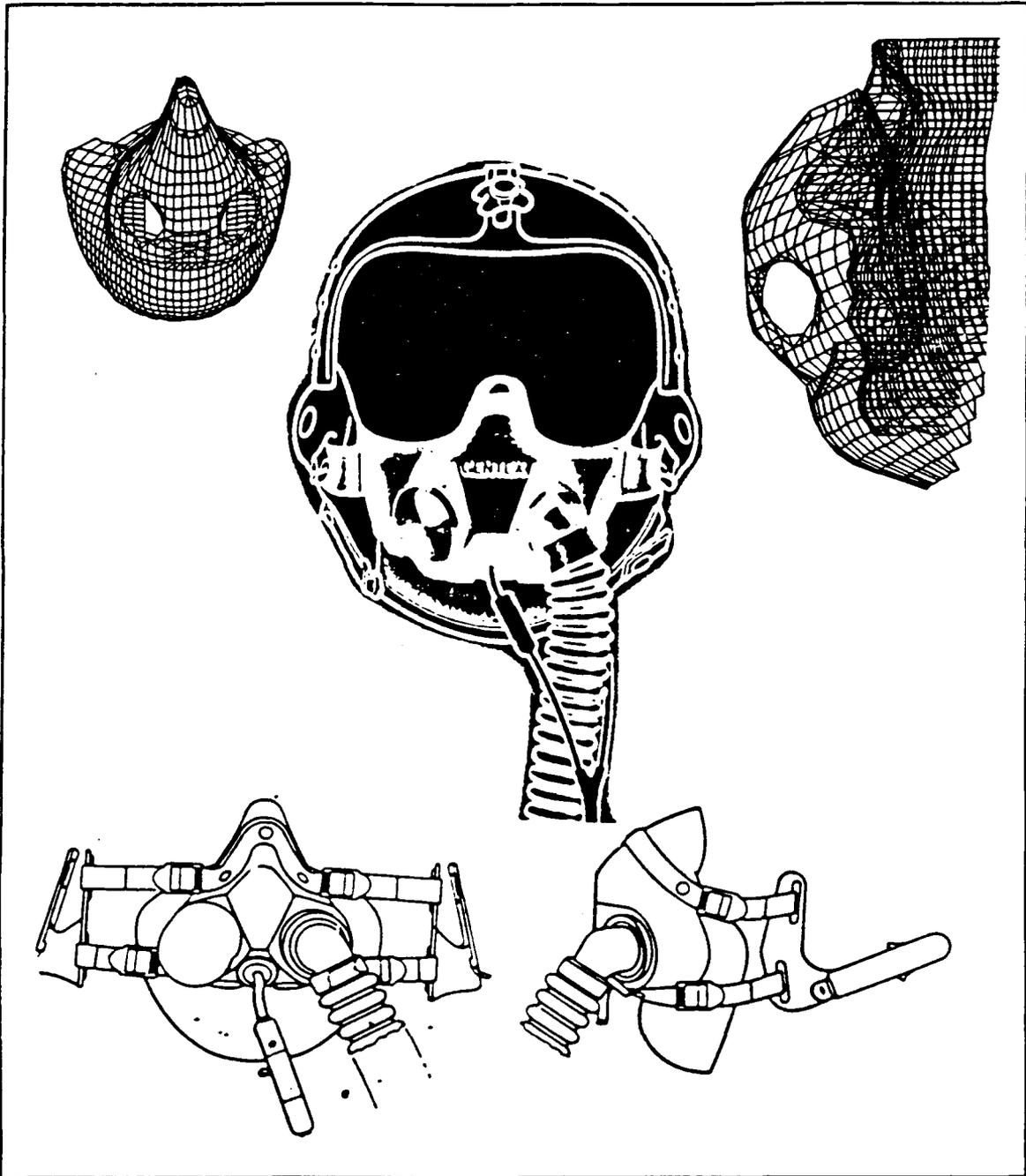


Figure 1. MBU-20/P Pilot's Oxygen Mask and FEM Representation

method (linear vs. nonlinear) should be considered when choosing the solution methodology. Finally, the analysis results must appear reasonable (duplicate known behavior under prescribed conditions) or the model or analysis methodology is

suspect. All of these factors were considered in the application of FEM to analysis of the MBU-20/P oxygen mask.

1.1 Problem Statement

The AAMRL is involved in studying human engineering aspects of equipment for Air Force pilots. One item of specific interest positive pressure oxygen masks: the standard mask sizes do not form a good seal on all faces, especially at high levels of oxygen pressure (up to 80 mm/Hg above ambient (2,18)) during high-G maneuvers:

".. a comfortable seal was difficult to achieve. When the mask was initially adjusted for comfort at 1 G, it would leak under the PPB [Positive Pressure Breathing] pressures generated at 9 G's." (1,31).

This problem is currently addressed by in-house custom fabrication of some pilots' oxygen masks by The USAF Medical Center/SGT, Wright Patterson AFB, OH 45433. Currently, approximately 300 MBU-12/P masks are custom manufactured per year. The material quality of the custom manufactured masks (dipped latex) is inferior to those produced by GENTEX corporation, the manufacturer of the MBU-20 (standard sizes only). The commercially produced masks are made of silicone based latex which is more durable. The Air Force's current fabrication procedures are outdated and time consuming, and not suited to increased production rates required during rapid deployments for contingencies such as Desert Shield /Desert Storm (9). The latest mask, the MBU-20 is not amenable to custom manufacturing, either by the Air Force or a contractor.

Some background on the MBU-20 positive pressure breathing mask will be useful for understanding how it was modelled and analyzed.

1.2 Positive Pressure Breathing Masks

The MBU-20/P mask is also known as the High Altitude-Low Profile Positive Pressure Breathing Mask (HA/LP-PPB) or the Tactical Life Support System (TLSS) mask. Positive pressure refers to the high pressure of oxygen inside the mask which augments the pilot's ability to sustain high positive G loads. An exploded view of the mask is shown in Figure 2. (9,308)

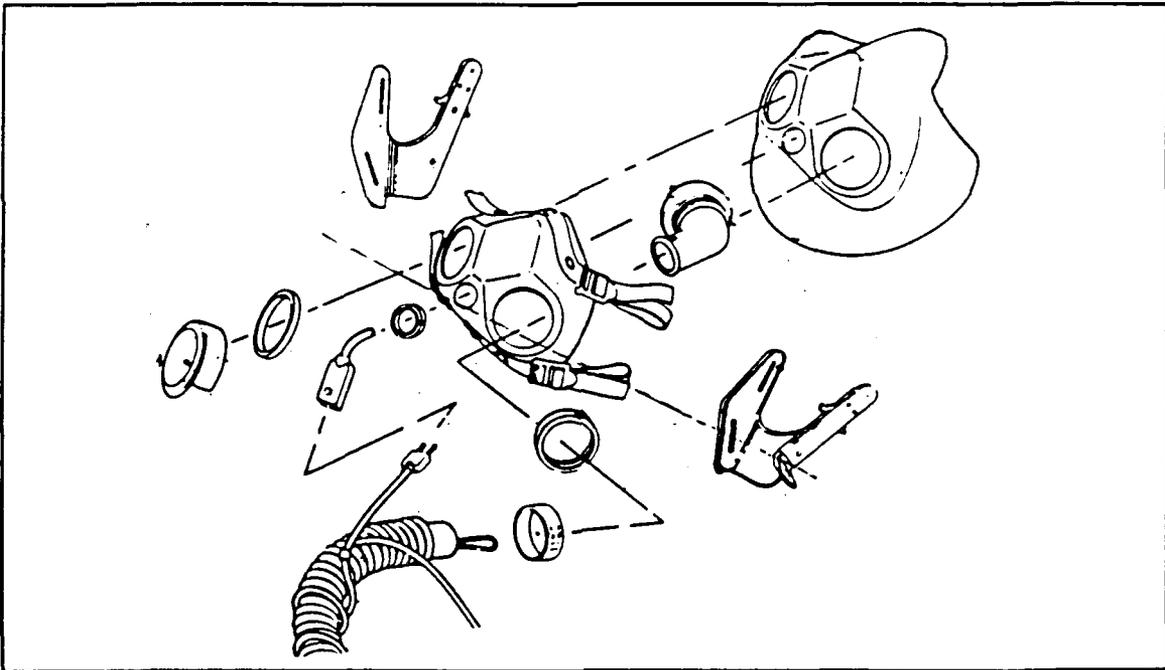


Figure 2. Parts Breakdown of MBU-20/P

It consists of a soft rubber (silicone latex) liner, and a 'rigid' fiberglass/epoxy shell (which covers only part of the liner). Retention straps are riveted to the hard shell and

special adapters (bayonets) mate with the helmet. The rubber liner consists of several distinct areas:

- a) The pressurized chamber (with inhalation and exhalation ports) whose proper seal to the face is critical.
- b) The chin/cheek flap area which fits rather loosely to the face and acts as a guide to proper positioning on the face.

More detailed design rationale is given in reference 1. The PPB mask was designed to be a replacement for the MBU-12/P mask (shown in Figure 3).

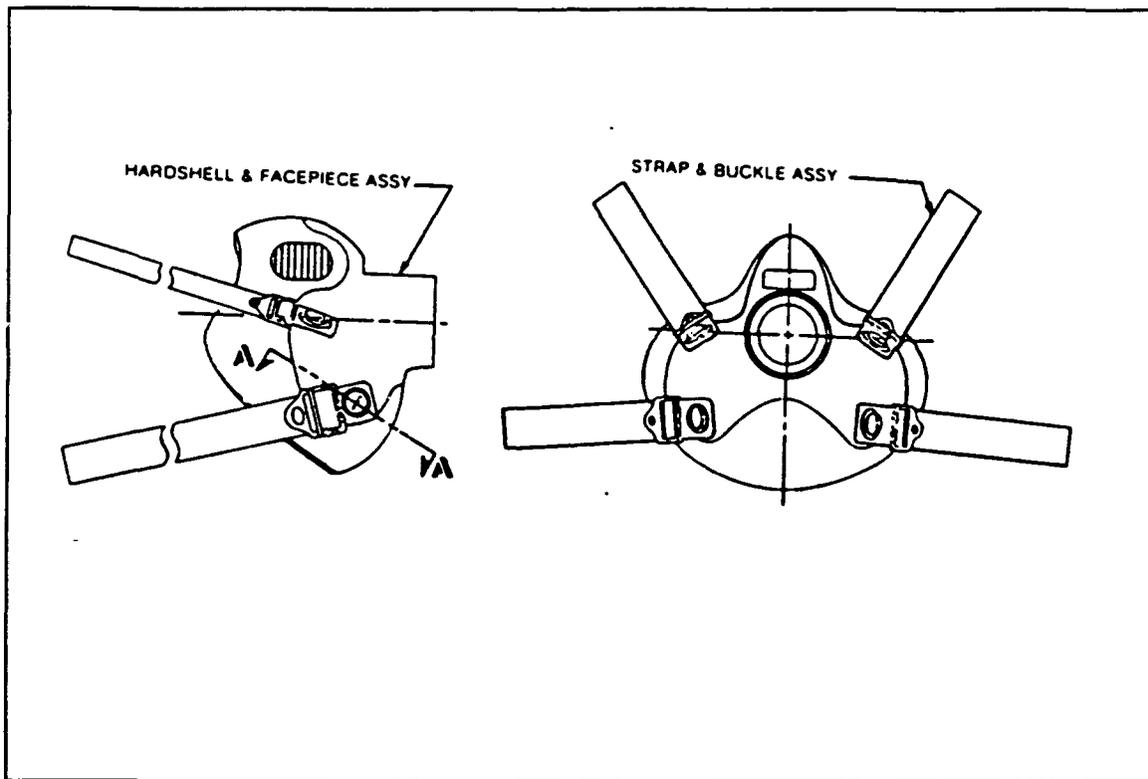


Figure 3. MBU-12/P Mask

Major differences are that the MBU-12/P has only a single intake/outlet and the chin/cheek flap forms a larger region which surrounds the pressurized chamber and this larger region

must seal to the face. Also, the hard shell area is physically attached to the soft rubber areas. A field evaluation of the MBU-20/P masks included the following comments:

"...it became apparent that the TLSS mask fit would be more critical than for current oxygen masks due to PPB...the mask seal was less comfortable...reported discomfort at the bridge of the nose...." (1,31)

The design requirements of the MBU-20/P used the MBU-12/P specifications (MIL-M-87163A USAF) as a starting point. The MBU-20/P mask is also a component in a larger system for providing protection from G-induced loss of consciousness (GLOC), including pressurized jerkins and may include provisions for chemical/nuclear/biological (CBN) protection which brings it under the umbrella of the Tactical Life Support System (TLSS). It is sometimes therefore referred to in the literature as the TLSS mask.

Many mask models, including the MBU-5/P, MBU-10/P (1977), and MBU-12/P (1982/83), preceded the development of the MBU-20/P. Inevitably, the redesign process will continue as such equipment is optimized to meet current or projected requirements.

The subject of this study is development of a capability to rapidly develop the models and perform the analyses necessary for this redesign process to occur in a timely and cost effective manner.

2.0 Scope

The bulk of this project involved development of the computer interface program (see Appendix F for SCANCAD program listing) to take an intermediate data file of three dimensional digitized coordinate data and put it into a finite element model format for modification and/or analysis by a commercial finite element analysis program. The term 'intermediate data file' is used to define the ASCII file which is a product of software at AAMRL which translates the binary output file from the digitizer to readable text format. The ASCII file format facilitates transfer among many different types of computers using the file transfer protocol (FTP) utility. A sample intermediate data file is shown in Appendix D.

The MBU-20 oxygen mask was digitized by a Cyberware 3D (color) digitizer. The digitizer determines the location (coordinates) in three dimensions of various points on the surface(s) of the object being scanned. The author's helmet clad head was also digitized by the scanner. More details about the capabilities of the scanner will be discussed below as they pertain to the form of the input data to the SCANCAD program. The scanner setup is shown in Figure 4 below.

After the items of interest were scanned by the Cyberware digitizer, the point coordinate data was converted to ASCII file format via the AAMRL in-house computer program. This

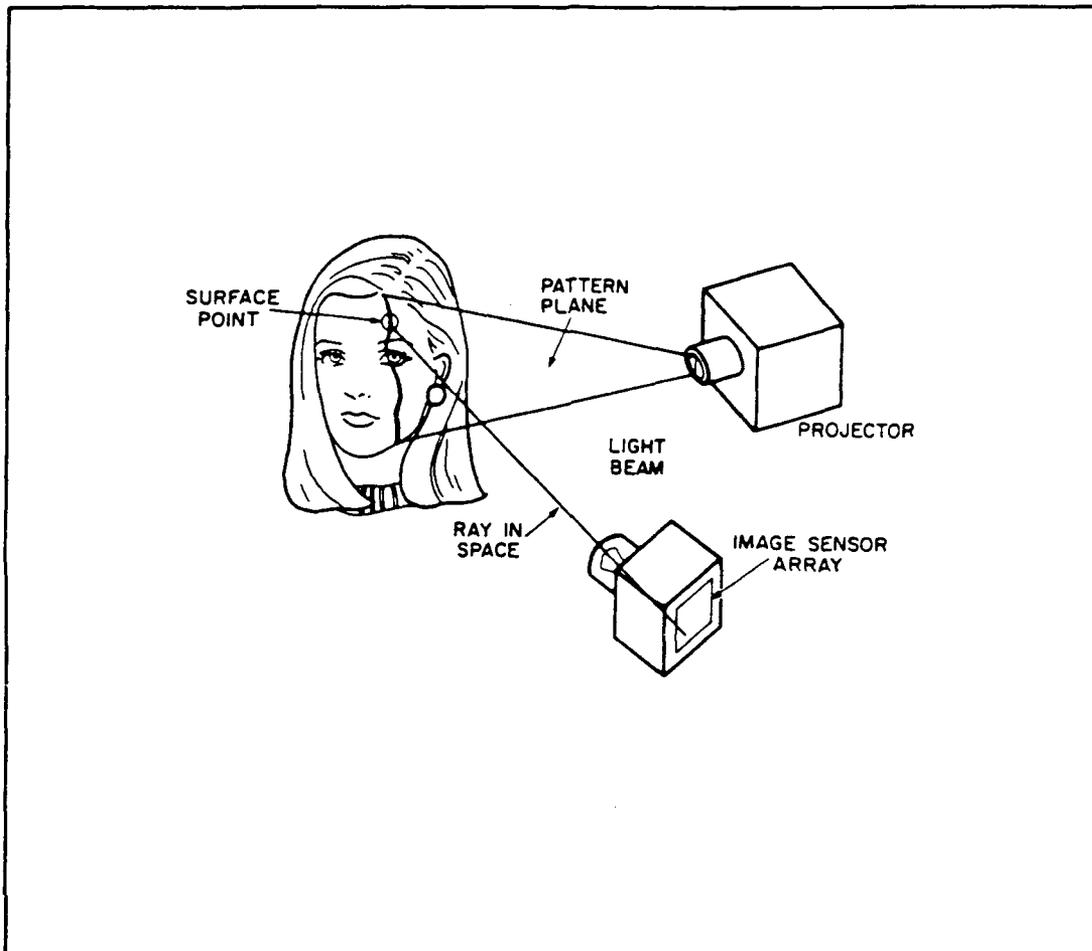


Figure 4. Cyberware Digitizer

intermediate data file (in the format discussed below) served as the input to the main computer program developed in this effort, SCANCAD. Using SCANCAD, this data was converted to finite element model format and input to commercial FEM programs (SDRC I-DEAS or PATRAN). Appropriate boundary conditions and structural loads were applied manually to simulate the behavior exhibited in real life.

Deformed geometry and stresses/reaction forces were assessed via the FEM analysis modules in I-DEAS. Various

features of the FEM programs were experimented with and alternate approaches considered to see which would work best for this particular problem. Changes were made to SCANCAD and supplemental programs to effectively deal with modeling challenges as they arose.

Realistic modeling of the face was outside the scope of this effort. The face was modelled as completely rigid. The tools developed as a result of this effort can aid the development of a flexible face model.

A list of steps involved in analyzing the mask is shown below.

1. Scan Mask and Face with Cyberware 3 D Digitizer
2. Convert data to ASCII for input to SCANCAD
3. Run SCANCAD program to output Finite Elements Input
4. Synthesize component models
5. Run Finite Element Analysis

There are many details and subtasks still to be addressed for steps 3,4 and 5. The specifics will be provided throughout this manuscript.

Since the SCANCAD program is the major tool developed and utilized in this effort, a detailed discussion of the development methodology, rationale, and capabilities pertinent to the analysis of the PPB mask follows.

3.0 Computer Program Development

The purpose of the computer interface program (SCANCAD) is to convert (with or without data reduction) the digitized coordinate data to a finite element format for a

commercial FEM program. Two commercial packages were targeted, SDRC I-DEAS and PDA PATRAN. Since SDRC I-DEAS is available at AFIT, the bulk of the development work was targeted toward it. At the request of AAMRL, interfacing to PATRAN was also included. PATRAN was accessed on ASD computers. Both packages offer similar features and similar modeling logic. A brief discussion of similarities and differences relative to this work may be useful for those considering one package or the other (see Appendix A). SDRC I-DEAS was run on several platforms, including a Tektronix terminal connected to a DEC VAX 8500 (VMS), and on a SUN Sparc Station under OPENWINDOWS and SUNVIEW. The preferred method was under SUNVIEW because it allows for dynamic viewing - real time on screen rotations, translations, zoom, and z-clipping. On all machines, disk space limitations required the model size be kept to a minimum.

The SCANCAD program was developed in standard FORTRAN 77 on the VAX computer, but is portable to other machines. The first place to begin in the development of any piece of software, is to analyze the expected inputs and desired outputs.

3.1 Input Data

This section analyzes the form and possible formats of the input data (for SCANCAD). This is dependent on the capabilities of the digitizer and its associated software.

The maximum size of object which may be digitized by this device is 400 mm high (15.7") with a radius of up to 200 mm (7.9"). The Cyberware digitizer moves around the object in a circle. The 360 degrees of a circle are discretized into 512 stations referred to as longitudes. The first longitude is arbitrarily located wherever the scanner starts and the 512th longitude is $360/512$ degrees away since the first and last longitude are adjacent, completing the full circle. The scanner pauses at each longitude station and records the coordinate data at regular vertical increments, referred to as latitudes. There are 256 latitudes for each longitude. The maximum resolution is therefore 512 by 256 for a total 131,072. The 3-D coordinate data are recorded as longitude station (1 through 512), latitude station (1 through 256), and radial distance (in millimeters) from the center of rotation of the scanner. This data can be output and/or converted to Cartesian coordinates. Typically, not all longitude and latitude locations have a non-zero radial distance associated with them. This is due to the object not filling the entire viewing area, holes or cutouts in the objects, or poor local surface reflectivity (distance is measured by reflection of a low power laser beam from the object). Also, the scanner results are sensitive to the object being off center. Shadow regions (due to complex curvatures) may be obscured from the field of view of the scanner. Data from a single scan will typically contain

missing data for certain areas of a highly convoluted surface such as the oxygen mask. It may therefore be necessary to scan an object more than once, at different orientations, to get a full object representation (this creates its own set of problems and possible solutions as discussed further in the section entitled model synthesis). Data point coordinates from two scans (with the mask suspended at different orientations) were provided by AAMRL as well as a scan of the author's helmeted head/face.

3.2 Desired Output Data

The 'final' goal is to create the nodes and elements comprising the three dimensional finite element model in a format readable by the commercial FEM codes. There are several approaches which may be taken to arrive at this final result. Appendix B discusses the way finite element models are stored by the programs PATRAN and I-DEAS. This is either in the form of large binary files which contain information about display parameters and defaults or in the form of compact ASCII files containing only a subset of all the possible model/program settings. In the compact ASCII files (PATRAN neutral file or I-DEAS universal file), every data entity has its own identifying 'type' label and its unique record format. This information can be found in the documentation that comes with the programs. The desired outputs are the nodes and element connectivities. Several

alternatives exist for developing with the element connectivities forming the three dimensional finite element model.

3.3 Connectivity

The input data represents isolated points in three dimensional space, but not surfaces or elements. The points must be connected in a meaningful way to yield surfaces or elements. Before deciding on a connectivity algorithm, a decision had to be made which of two broad approaches should be taken:

- a) define the surface geometry, then let users map nodes and elements to the geometry utilizing the FEM program's capabilities.
- b) define the finite element model directly (as nodes connected by plates) without associated geometry.

Approach (a) initially looked attractive because it would allow various mesh densities to be mapped to the surfaces. The mesh density could be reduced from that available from the scanner (model reduction) or the mesh density could be increased (model refinement) for greater accuracy. In developing a finite element model from scratch, the normal sequence of activities follows approach (a). Usually, a small number of easily understood and defined surfaces will be used to model an entire component. Typically, this will take the form of a surface generated from primitives (sub-surface of a cube, sphere, cone, etc.), a surface of revolution formed by revolving a curve around

an arbitrarily defined axis, or a small number of ruled surfaces (surfaces defined and bounded by two curves in space). An initial approach was considered to convert the digitized points to GRID points (PATRAN Phase I geometry) or POINTS (I-DEAS geometry). Every 16 geometric points could then be connected to form bicubic surfaces. The surfaces could then be mapped with various densities of node meshes (PATRAN Phase II FEM connectivity / I-DEAS mapped mesh of nodes). The node mesh/surfaces could then be mapped with elements from the element libraries in the computer aided design (CAD) program (PATRAN or I-DEAS). Universal and neutral file formats exist for surfaces. In PATRAN, this would require considerable computational and storage overhead (i.e. computation and storage of the 48 coefficients for each bicubic PATRAN patch involving partial derivatives in each parametric direction. A bicubic interpolation algorithm could be written using every 16 points (4 points in each parametric direction; see Appendix B). I-DEAS, which uses a Non-Uniform Rational B-Splines (NURBS) representation is more general but could require more complicated algorithms. A bicubic representation could be used in I-DEAS, being a specific case of the more general NURBS representation. Both approaches would result in large storage requirements. The I-DEAS documentation on universal format definition for surfaces was not written for the uninitiated: a more comprehensive discussion of NURBS can be

found in reference 5. However, since these commercial programs already contain algorithms to calculate the correct surface coefficients based on a given grid connectivity, a simpler approach (from a programming viewpoint) would be to write the grid points out in a neutral/universal file, and then write out the surface connectivity as a session/program file (ASCII file containing menu choices or other commands allowed by the program). In PATRAN, this would require writing commands which tell PATRAN which 16 grid points to connect to form a surface using the '16G' patch creation option. The interpolation algorithms are internal to the PATRAN program. A similar approach could be taken in I-DEAS but would still require a good understanding of NURBS surface definition (the documentation is not very straightforward in this area either). An I-DEAS surface definition menu option exists for 'THRU POINTS', and requires picking boundary and interior points. These commands could be incorporated into an IDEAS program file (.PRG extension). A small amount of experimentation led to inconsistent results. The defined surfaces were not always confined by the boundary points defined (although the edges did pass through the boundary points). This was not a viable approach. This approach is possible (and easy) within PATRAN but was rejected because of a major disadvantage - it is very slow compared with the neutral /universal file approach. A neutral file containing several

thousand entities can be input to the FEM code in a couple of minutes. The time required to run through the several thousand menu commands to recreate those same entities will be one to two orders of magnitude greater. Neutral or universal file surface representation was also rejected after some deliberation. The dubious advantages (and perceived lack of need for this particular capability to complete the mask analysis project) did not justify the increased time and effort which would be required to implement this approach. Developing an algorithm to intelligently pick out a minimum set of digitized points to connect, for good surface representation, was beyond the scope of this project and is being studied by others. Without such an intelligent algorithm, surface representation offers little advantage (as well as some potential disadvantages) over direct element representation. One surface edge must not overlap more or less than exactly one other surface edge and contiguous surfaces must be of the same order (polynomial). Surfaces must be mapped with nodes so that boundaries between adjacent elements have the same number and placement of nodes. This could require a large manual effort if changes in mesh densities from one area of the model to another are desired. If changes in mesh densities are not desired (i.e. each surface is mapped with the same 'n by m' node mesh), then there is no real advantage over first starting with an element representation

(approach b). The main advantages of surface rather than element representation are:

- 1) mesh densities may be experimented with and viewed graphically in real time.
- 2) mesh refinement may be accomplished beyond the maximum resolution of the scanner.

Approach (b) was taken, but with features designed to capture many of the typical advantages of surface representation. A PATRAN surface representation was re-examined at the end of this effort and is detailed in Appendix B. Approach b, direct element representation, is further detailed below. Before the nodes can be connected to form elements, they must be ordered in some meaningful way. This ordering was obtained by taking the sequential ordering and converting it to a two dimensional parametric representation based on the way the scanner obtains the data.

3.4 Parametric Representation of the Data

The longitude, latitude coordinates of the digitized points are integer values and form a good basis for forming a two dimensional, rectangular parameterization of the data to facilitate development of the connectivity (connectivity is not output by the digitizer). The height and width of the parametric space are represented by the minimum and maximum limits of the (longitude, latitude) coordinates. A four noded 'quad' element can be formed by connecting points

found at the following parametric locations (1,1), (2,1), (2,2), (1,2). Having formed this element, one can then move up through the latitudes to connect the points at (1,2),(2,2),(2,3),(1,3). After sweeping through the latitudes, one could sweep through the longitudinal direction until all the elements needed to model the object are formed. Since the object scanned generally does not fill the entire field of view, a scanned point will not exist at every parametric location. But, element connectivity is based on connecting all points in the parametric space. The SCANCAD program therefore has to distinguish between which points (in parametric space) were scanned points, to know which elements to keep. It does this by assigning a non-zero color value to parametric locations containing a scanned point (from the input file from the scanner). If one point of the four is not a scanned point (zero color value), a triangular element is formed. If two or more of the points have a zero color, the element is not formed. The coordinates of the points (x,y,z) are transferred to two dimensional arrays and become functions of their position in the parametric space (i.e.xx(1,1)=x(1), means that the x coordinate value associated with the first longitude and first latitude is assigned the x value of the first point digitized). This can be illustrated by the figure below:

Latitude

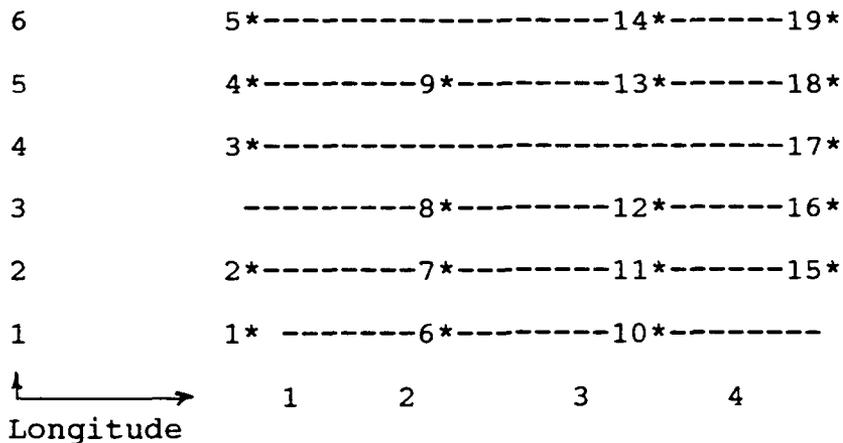


Figure 5. Parameterization of Nodes

The '*' represents a scanned point. The number at its left represents its position in the sequential file (order it was recorded by the scanner).

The scanner digitized 19 points on a body using 6 latitudes at 4 longitude stations (24 locations with 19 scanned points and 5 empty locations). The sequential position of a scanned data point in the input data file is located to the left of the asterisk. The coordinates of point 1 are associated with the 1,1 (longitude,latitude) position which is also the first sequential position in the parametric space (proceeding through all latitudes before moving to the next longitude). The coordinates of point 15 are associated with the 4,2 position, which is the 20th sequential position in the parametric space.

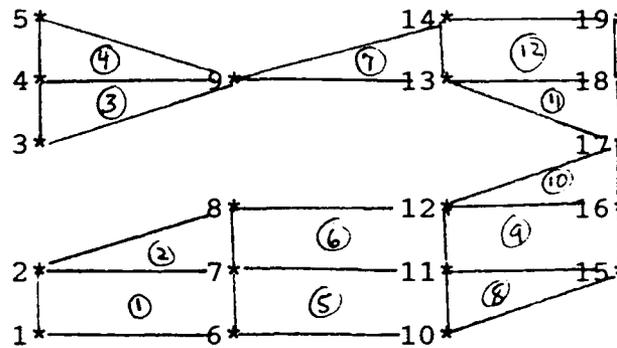


Figure 6. Element Connectivities

Parametrically square elements can now be formed (Figure 6) by connecting the four points whose parametric locations are $(1,1)$, $(2,1)$, $(2,2)$, and $(1,2)$. Element 1 would be a 4 noded quad element connecting points 1,6,7,2. If only 3 scanned points are available, a triangular element can be formed. If less than 3 scanned points are available at the 4 parametric locations interrogated during a particular iteration, that element can be skipped. This algorithm was implemented in the SCANCAD program. Using the previous example, this would result in 12 element connectivities.

The first five element connectivities are as follows:

EID,G1,G2,G3,[G4],Type	EID = Element Label
1 1 6 7 2 Quad	
2 2 7 8 Tri	$G_i = i^{\text{th}}$ Node to be connected
3 3 9 4 Tri	
4 4 9 5 Tri	
5 6 10 11 7 Quad	

Having determined the element connectivity, the SCANCAD program is ready to write out the FEM format file (neutral

or universal) of nodes and/or elements. The conversion of the three dimensional digitized point coordinates to three dimensional node coordinates is now just a matter of formatting according to the FEM program's users' manuals. The cartesian coordinates (x,y,z) of the nodes are output to a universal or neutral file (i.e. 'NODES.UNV').

The element connectivity is output to another universal/neutral file (i.e.'ELEMS.UNV'). Keeping the nodes and element connectivities in separate files has advantages which will be discussed later. The (longitude, latitude) parameterization coordinates of nodes are output to a third universal/neutral file (i.e. 'FLAT.UNV'). This last file assigns color to the node based on radial coordinate, and also assigns a third coordinate (user scalable) based on radial distance (for visualization). The following mapping occurs for an arbitrary point j:

```
x(j) = Longitude(j)
y(j) = Latitude(j)
z(j) = Scale Factor times Radius(j)
color(j) = z(j) divided by range of z times maximum color
number of Ideas (16 colors).
```

The last file is very useful because it can be read into the FEM program and will display the scan in parametric space. The quality of the scan can quickly be assessed (locations of missing data). The location of any detail can be determined in parametric space coordinates as well. This makes it possible to pick out the parametric locations of various details of the model because the (x,y) coordinates

of a node are actually the longitude and latitude corresponding to that point in the three dimensional cylindrical space of the scanner. Graphical examples follow.

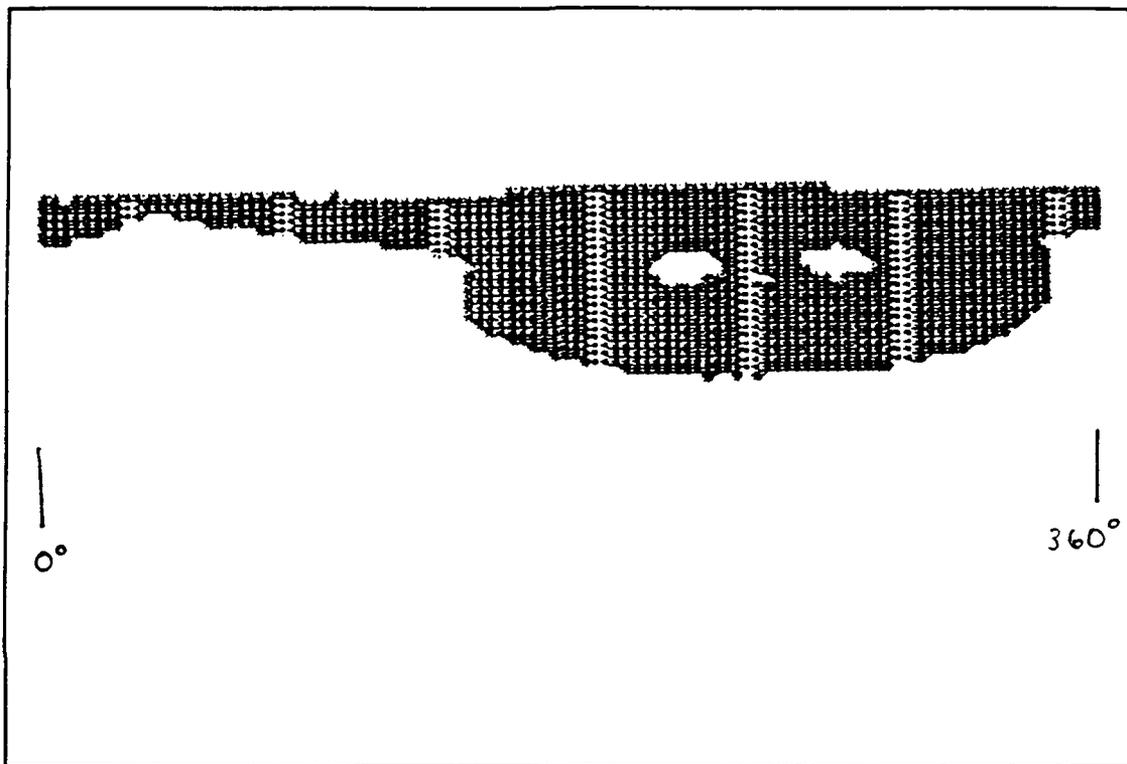


Figure 7. Parametric Representation of Mask

Figure 7 shows an isoparametric (each element has unit dimension in each parametric dimension) mapping of the scan of the mask front. The two large holes visible are the intake and exhalation valve openings. The scan spans 360 degrees, so in reality, the right edge (location of last scan longitude) connects again with the left edge (location of first scan longitude). Figure 8 shows a parametric map of the face. Figure 9 shows a parametric map of the mask

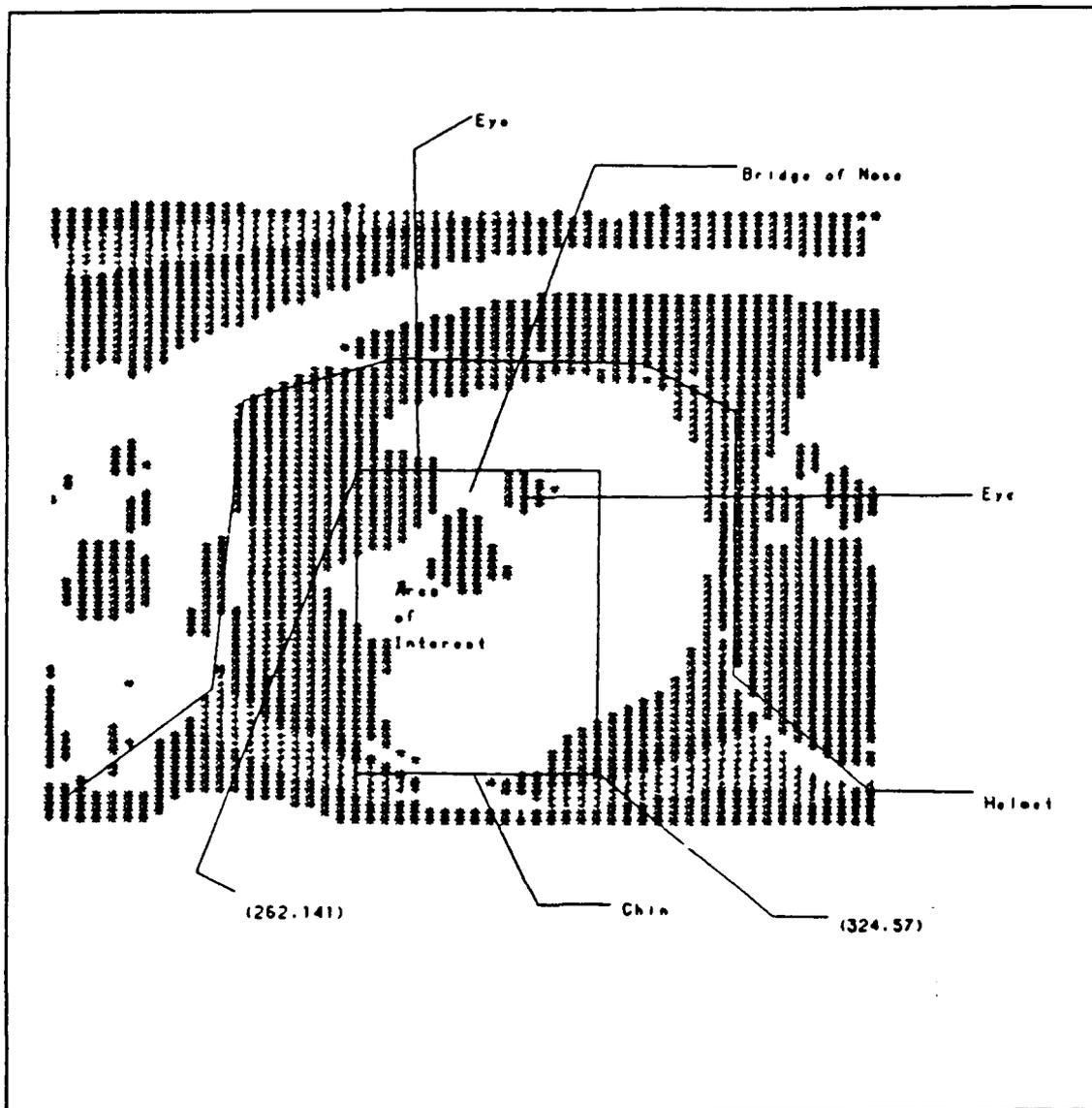


Figure 8. Parameterized Head Scan

seal region. The isoparametric mapping (each quad element is a square with sides of one unit length) as displayed by the CAD program is useful for determining several facts about the scan - location of first and last scan longitude, quality of scan (do holes show up where they shouldn't?), qualitative information on the orientation of the object being scanned

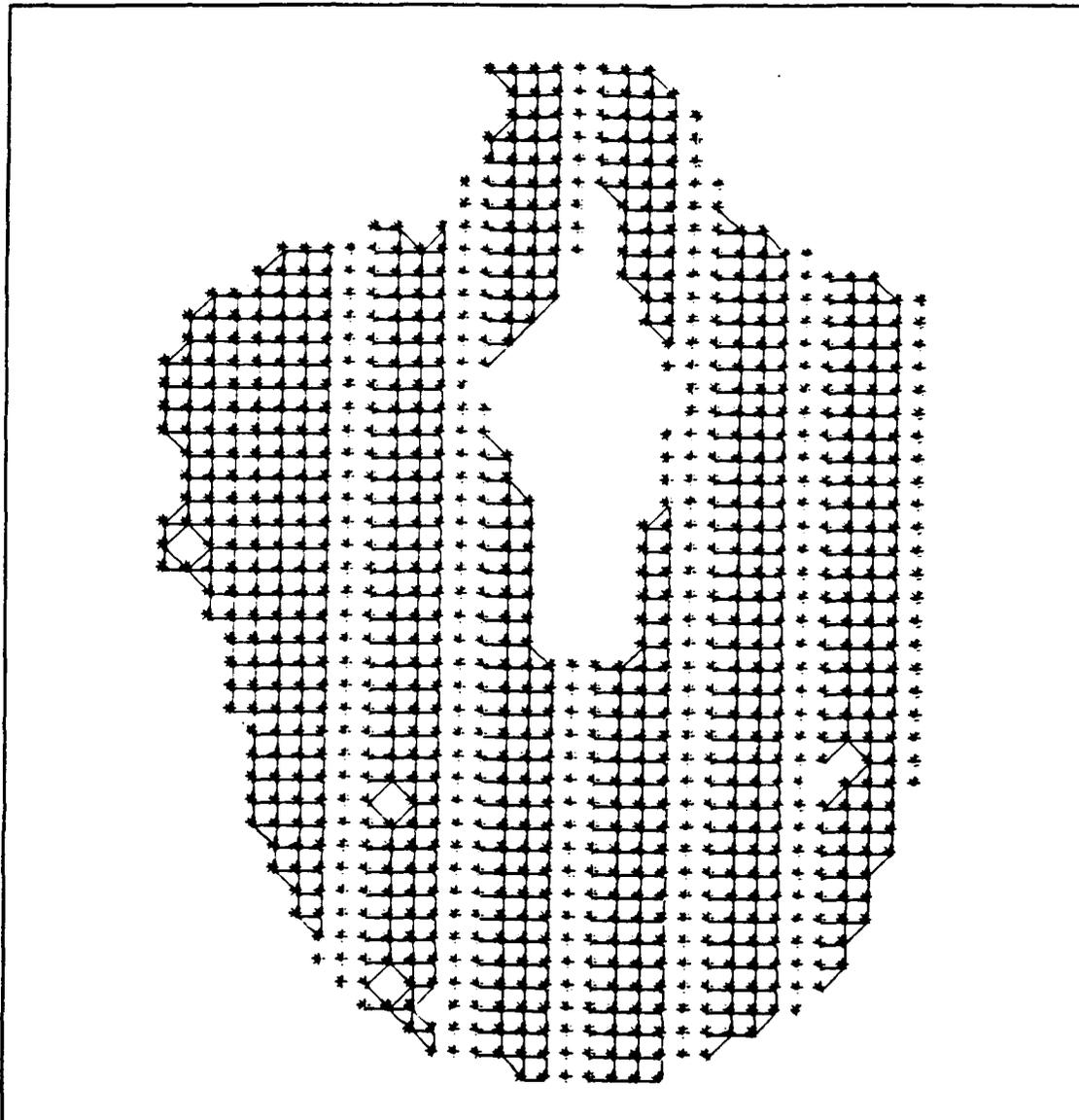


Figure 9. Mask Seal Region

(on center, tilted, etc.). Additionally, the SCANCAD program will output to the screen or print a listing of the scan limits (Appendix C). Up till this point, only a one-to-one mapping of scanned points to nodes has been discussed. To emulate certain features of actual surface representation, the SCANCAD program should allow for selective model reduction. Model reduction is needed

because large finite element models become unwieldy and consume large amounts of computing resources. Selective model reduction was incorporated into the SCANCAD program.

3.5 Model Reduction

Computational capacity limitations revealed that a straight one to one mapping from digitized coordinate data to FEM nodes would be impractical. Manipulation of the model became sluggish at about 1000 - 2000 nodes and elements. A maximum practical limit, depending on the number of users on the system and patience and determination of the user, is no greater than about 3000 nodes and elements. Using Ideas on a Tektronix 4211 graphics terminal and hosted on a VAX 8500 was impractical for a model of the complexity and size of the oxygen mask. There were so many lines on the screen, it was easy to lose perspective. The SUN SparcStation was certainly faster, with the major advantage being real time rotations / translation / and z-clipping during model modification. A dedicated workstation such as a Silicon Graphics (SG) Iris would facilitate model editing. Versions of both PATRAN and I-DEAS are available for the SG system.

Model reduction was achieved by three methods:

- a) selective digitized point data reduction
- b) selective feature extraction
- c) use of symmetry

3.6 *Selective Digitized Point Data Reduction*

As discussed earlier, the node points are represented in a parametric rectangular space where every four points could be connected to form a quadrilateral element (or three points to form a triangle where one point is missing). Model reduction can be achieved by connecting only points in every other longitude and/or every other latitude. This feature was expanded to let the user specify that the program should keep only points which are located in every n th longitude and every m th latitude. The model would be reduced by a factor of $m \times n$. Additional flexibility is added by choosing a slightly more complicated algorithm to specify which latitudes and longitudes to keep. A double ramp function was chosen which allows the user to specify for each parametric direction (longitude or latitude) independently, a starting value, intermediate value and final value for how many longitudes or latitudes to skip when deciding to keep nodes. The intermediate value need not occur at the 50% point, but may be moved either way to bias the node spacing. This approach was also implemented in the SCANCAD program and worked well.

A non-uniform parameterization of the head scan is shown below. A practical approach to using this feature is to initially choose a uniformly reduced (in each parametric direction) mesh density and then view the parametric model in I-DEAS.

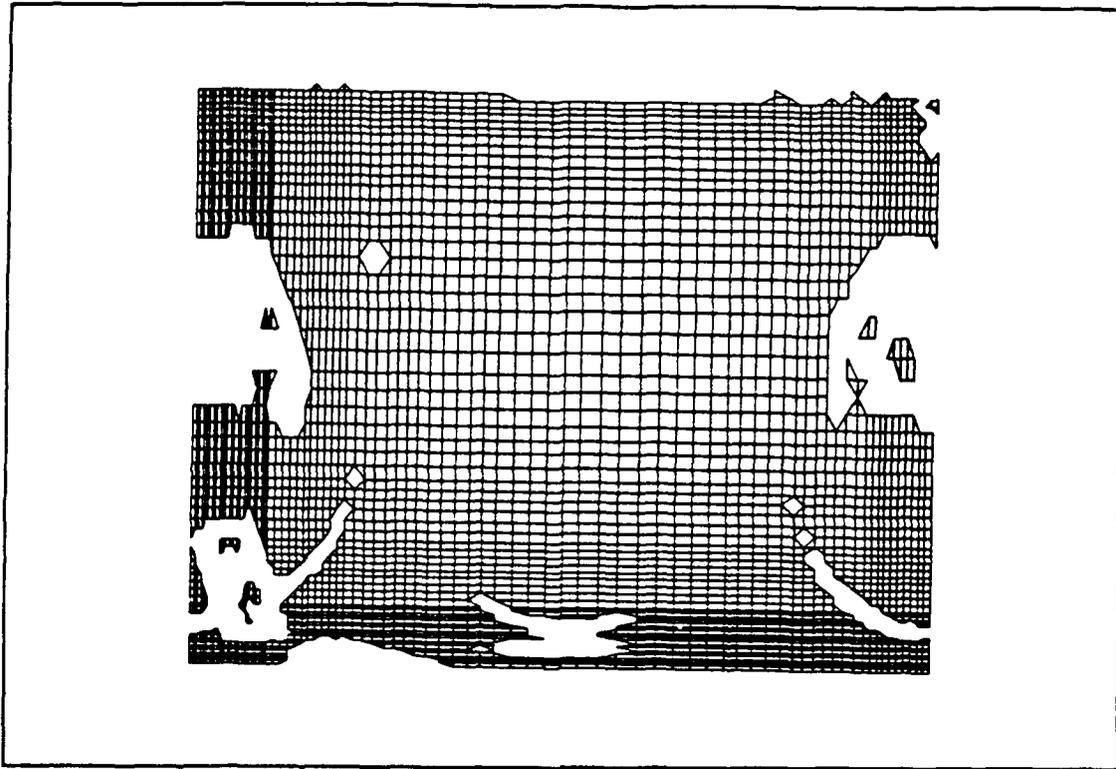


Figure 10. Non-Uniform Parameterization of Head Scan

3.7 Selective Feature Extraction

So far, model reduction has been accomplished by using a mesh of lower density than the maximum resolution of the scanner. Another desired approach of model reduction is to extract only those areas of the scan that you may be interested in rather than including the entire scan. The scan of the face includes approximately 33,000 digitized point coordinates, but the region of interest is that area around the nose and mouth which will couple with the mask. Some of the surrounding regions may also be kept as an aid to visualization. The SCANCAD program therefore includes a feature for including only a 'rectangular' subset of the

parametric space. To distinguish what the limits (minimum longitude, minimum latitude to maximum longitude, maximum latitude of interest) are that include the features you want, the entire model can be converted (with mesh reduction) to the parametric representation. To be able to pick out the features, the color of node points is related to the radial coordinate of the node. The elements can be color coded to be the average of the colors of the connecting nodes (see Figure 8). The color contours produced give an indication of where features (i.e. the nose, mouth, eyes) are located. Alternatively, a single color can be used but a third dimension (z-height) is added to the 2 dimensional parametric mapping based on radial coordinate. The user can list the coordinates of the diagonal points bounding the rectangular region of interest (refer back to Figure 8, where the region of interest is the region of the face around the nose and mouth -- longitudes 262 through 324 and latitudes 57 through 141).

The x-coordinate of a node is the longitude where that point occurs and the y-coordinate is the latitude value. The SCANCAD program can then be run with the limits of interest specified when prompted for. This can also be used for mesh refinement in local regions.

A third method of model reduction is to take advantage of symmetry. This is not a specific feature of the SCANCAD program. Deletion of a redundant half-plane can take place

by feature extraction using SCANCAD or may be done while running the FEM program. The advantage of using SCANCAD is that less time will be required to read the data into the FEM program. A discussion of how symmetry was used in the model development of the mask/face can be found in section 4.1 of this manuscript. It is not as straightforward as merely picking the longitude on the plane of symmetry since the longitudinal direction may not be parallel to the plane of symmetry. Having developed the SCANCAD program, it was then used to develop the finite element data for input to I-DEAS.

4.0 Finite Element Modeling

The finite element modeling task was performed using I-DEAS. A complete model had to be synthesized from universal files of nodes and elements which were read into I-DEAS. Missing data had to be filled in manually. Component models had to be aligned. Material and physical properties had to be added, as well as loads and boundary conditions.

4.1 Model Synthesis

The finite element model had to be synthesized from several scans. The first scan of the mask captured detail primarily on one side (the front) of the mask; the face seal region (the region of primary interest) was not captured by the digitizer. This scan contained 27,000 scanned points

which were placed in a 'rectangular' array spanning the parametric subspace (512 longitudes x 95 latitudes = 48,640 discrete locations). The 27,000 digitized points (labelled sequentially) were assigned their locations in the larger array of 48,640 points, and distinguished from non-digitized points at parametric locations which did not have digitized data associated with them by assigning them a non-zero color value. They were connected as elements using the SCANCAD software.

For finite element analysis, aspect ratio (of elements) is a concern. In the first scan of the oxygen mask, it was suspended on a string from the nose bridge region of the mask. In cylindrical coordinates, this region extends downward in the axial direction while maintaining a small radial distance. Uniform meshing in both the lateral and longitudinal directions will result in elements being overly long in the vertical direction since 512 longitudes must span a small circumferential region. A top down view of this region can be seen in Figure 11. Note the top down view of the mask at left, and the enlarged detail of the voids at the nose bridge region shown at right. Notice how long and thin some of the elements are. Also, notice that there is a void at the center (where the mask was suspended on the string). That void is an unavoidable consequence of the fact that this region is parallel to the scanning beam.

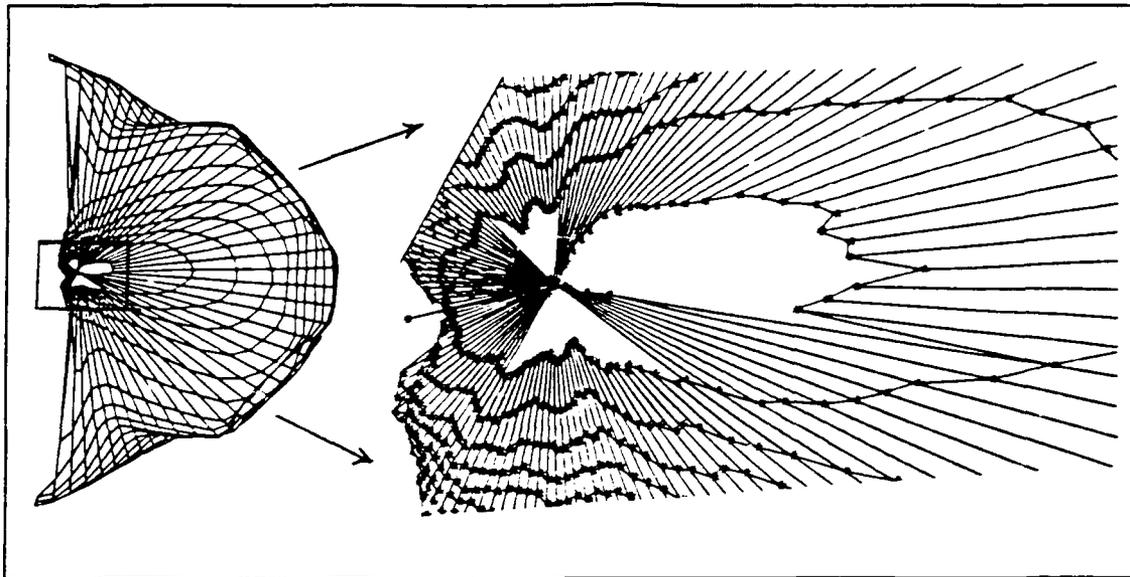


Figure 11. Top Down View of Mask Nose Bridge Region

Choosing a mesh of lower density in the longitudinal direction will tend to produce elements with more reasonable aspect ratios. This was done for various regions which were then manually connected via a transition mesh. Voids can be filled with nodes interpolated between other existing nodes and elements defined to span the gap and connect with other elements.

As noted previously, a single scan does not usually span the entire surface of the object due to various factors. In the case of the MBU-20 mask, with its complex curvatures, a good scan of the front of the mask was achieved, but detail of the seal region was not captured. A second scan was done with the mask oriented differently and produced good detail of the face seal, but not much else. Scanning a concave object such as the seal region causes

connectivity problems. The points on the seal are scanned at a particular longitude. As the scanner scans up the latitudes it comes to a point where the concavity is and starts picking up points on the inside of the mask, but those points really belong on the longitude which is 180 degrees around the scan circle. Connectivity by the SCANCAD program assumes points along a particular longitude should be connected. The result is illustrated below.

Manual deletion of elements and nodes had to be done within I-DEAS to clean up this component. A third scan was done to digitize a face. To combine the face seal with the mask front, each file (of digitized input) was run through the SCANCAD program. The two universal files were read into IDEAS (with node and element ID's offset so IDEAS would not give a 'Duplicate Definition' error). The mask front model was edited to remove the small part of the face seal region at the nose bridge area to make room for the seal region (to prevent overlapping elements). The mask front model and seal region model were defined as separate groups which could be oriented (rotated and translated) relative to each other. The seal was then oriented so that it was in the proper position relative to the mask front. The mask front and seal had to be manually connected in IDEAS by defining transition region elements connecting nodes on the seal to nodes on the front. This was difficult and very time consuming since there were still a large number of nodes and

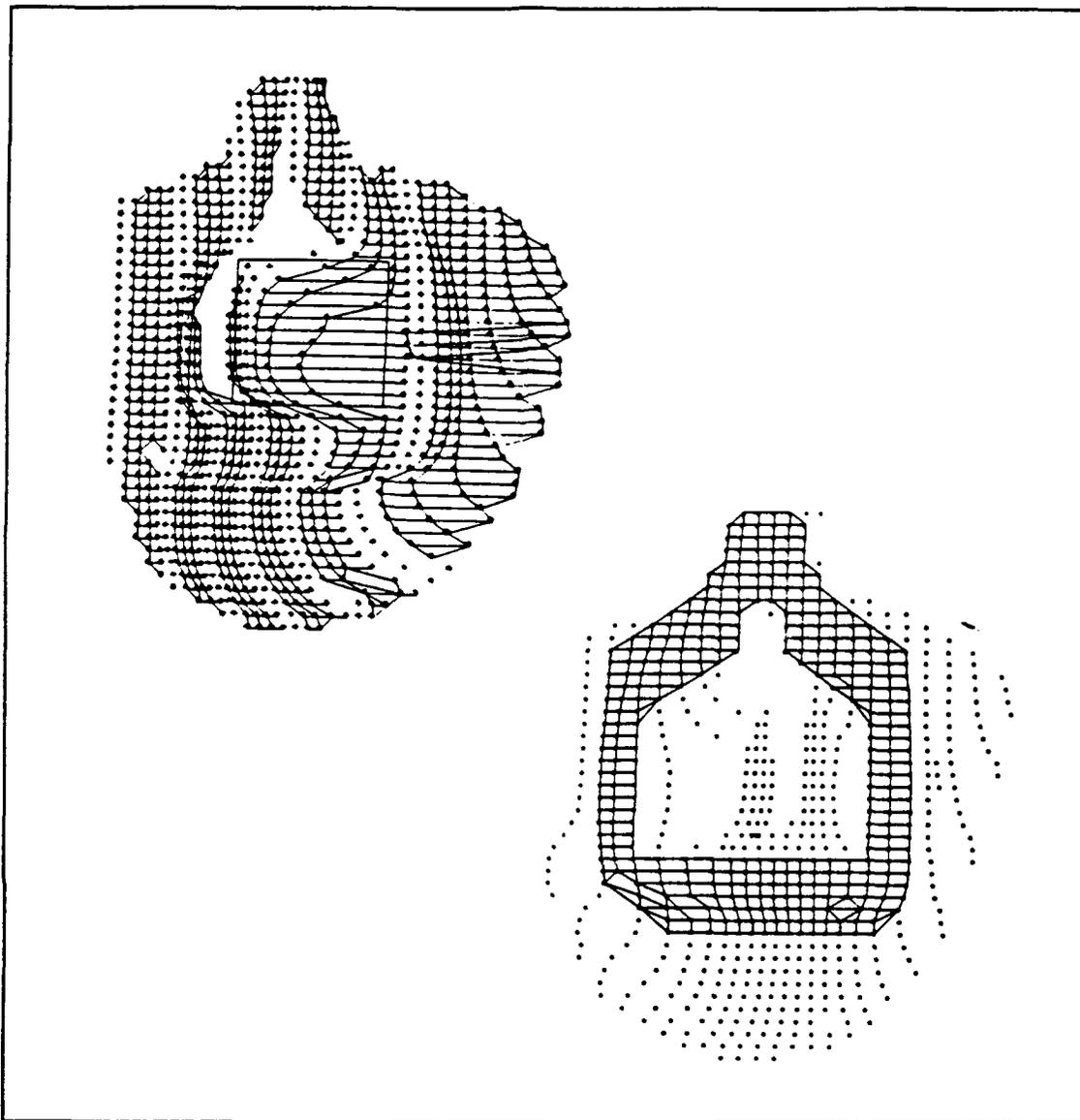


Figure 12. Scan and Connectivity of Seal Region

elements. Complicating the connection was the fact that the seal scan was produced at a different orientation than the first scan (to capture the seal region which was missing from the first scan) and with differing levels of model reduction. The author's face was scanned, the digitized data converted and reduced by the SCANCAD program, read into I-DEAS and oriented to line up with the proper positioning

of the mask. The complete mask model was then offset from the face in the positive z direction. Pushing on the mask in the negative z direction would simulate tightening of the mask on the face. Gap elements (see discussion in Appendix M) were initially used to model the connection of the mask and face. Model reduction was done within SCANCAD, keeping every 7th scanned point on the mask front. This is a fairly coarse model and was used as a first 'cut' to see if all the pieces of this project could be put together. Model refinement can be accomplished in selected areas of a model by superimposing universal files of nodes and / or elements of selected regions. Some manual effort is still required to define transition regions to ensure proper element to element connectivity. Due to the coarseness of the mesh used, the oxygen intake and exhaust holes do not appear very round. If this was an area of interest, a more refined mesh (the full resolution of the scanner) could be superimposed as in Figure 13. Elements could then be defined to connect the nodes, giving a more circular appearance.

Symmetry was also used in reducing the model. The mask and face both contain a vertical plane of symmetry including loading and boundary conditions. The model size can be cut in half and appropriate boundary conditions added. This should be easy to accomplish as long as one longitude station lines up with the plane of symmetry of the object scanned. When the mask was scanned, it was suspended at a

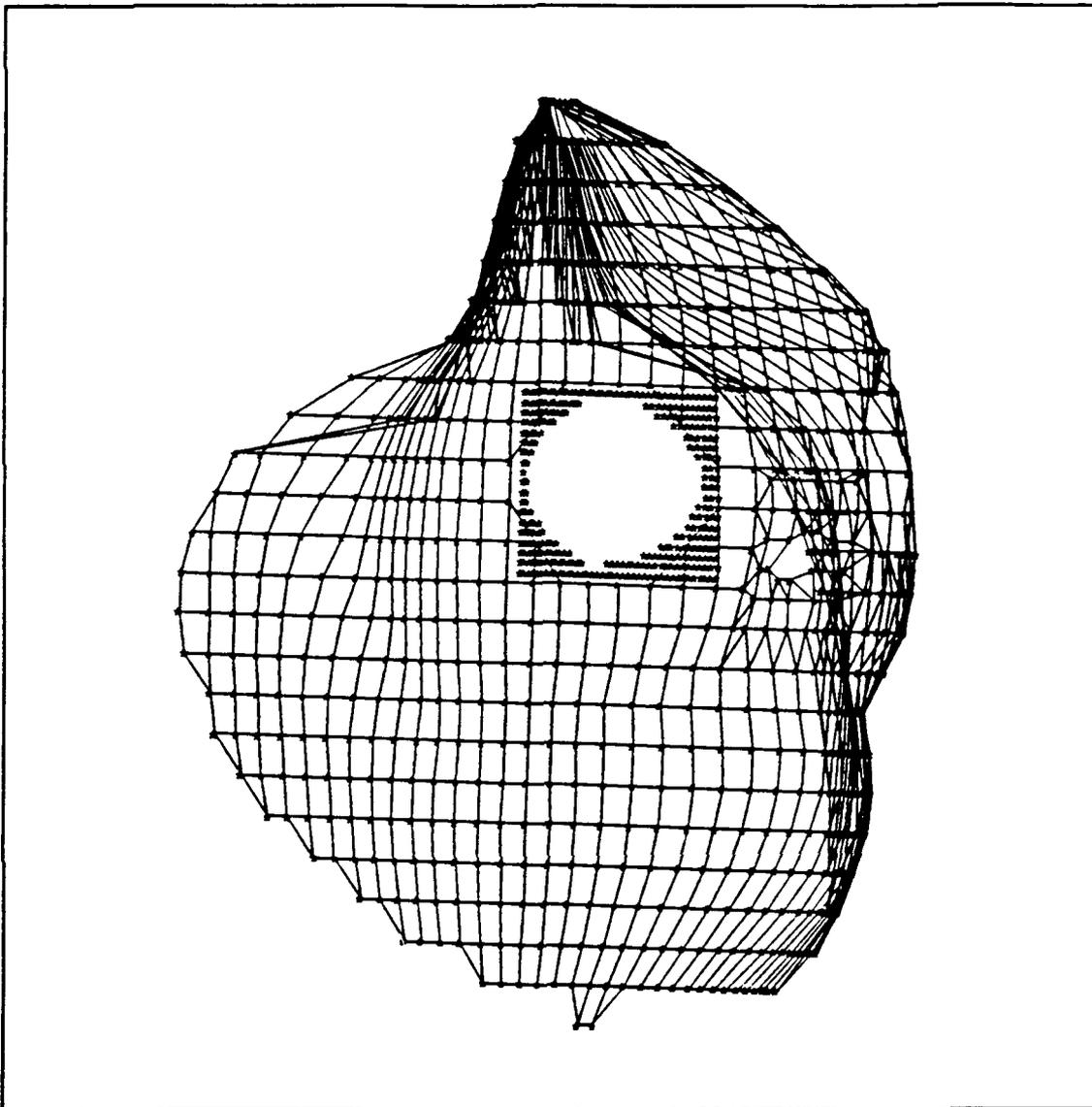


Figure 13. Superposition of Different Mesh Densities

slight angle to the plane of symmetry. This plane was determined by drawing a line (in I-DEAS) from two similar points located equally about the plane of symmetry. The two points chosen were the center of the two large holes (intake and exhaust). The line connecting the center holes is visible in figure 14.

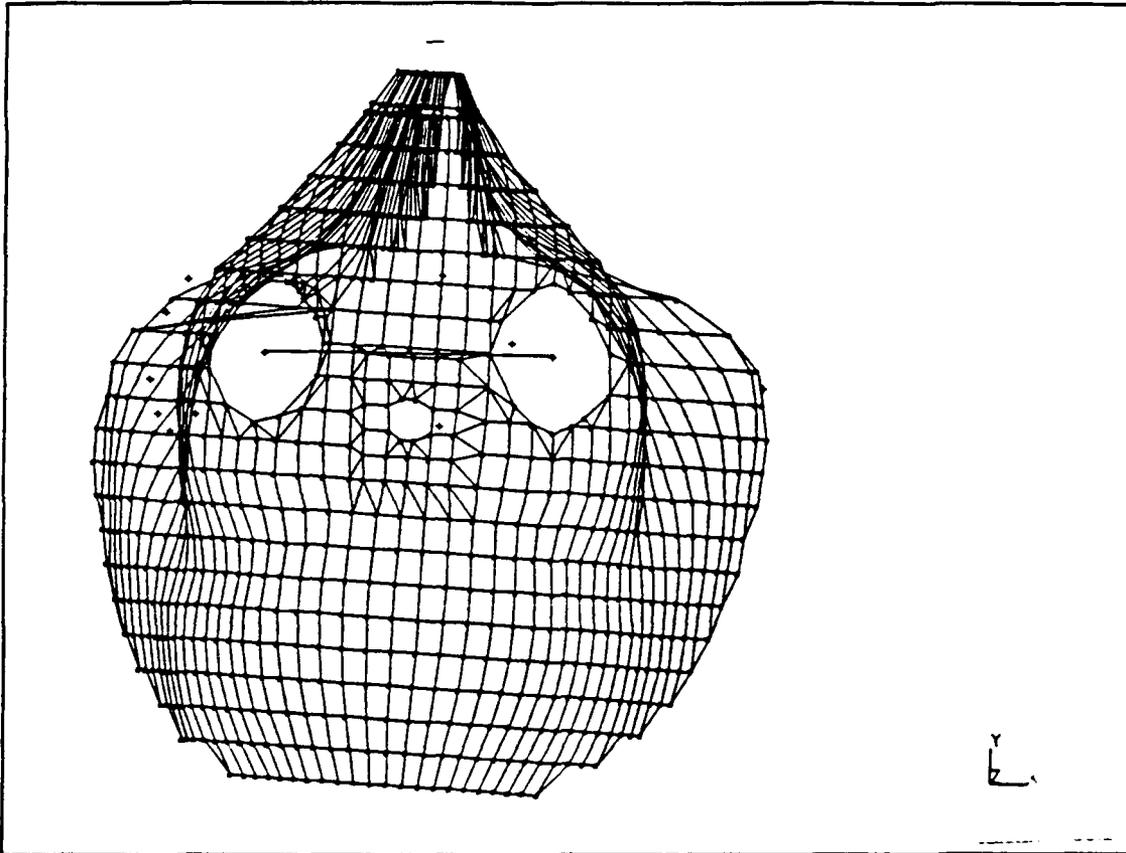


Figure 14. Aligning Plane of Symmetry

The angle was measured and the elements reoriented. Half of the elements were deleted along the longitude nearest the plane of symmetry. The boundary points near that plane then lay in a plane tilted from the plane of symmetry. The boundary points were reoriented by listing their coordinates and moving them to the plane of symmetry. The plane of symmetry was chosen to be the YZ plane, with the mask facing the positive z-direction. The half mask model contained 800 nodes. The full face contained 1600 nodes. The completed model is presented in Figure 15.

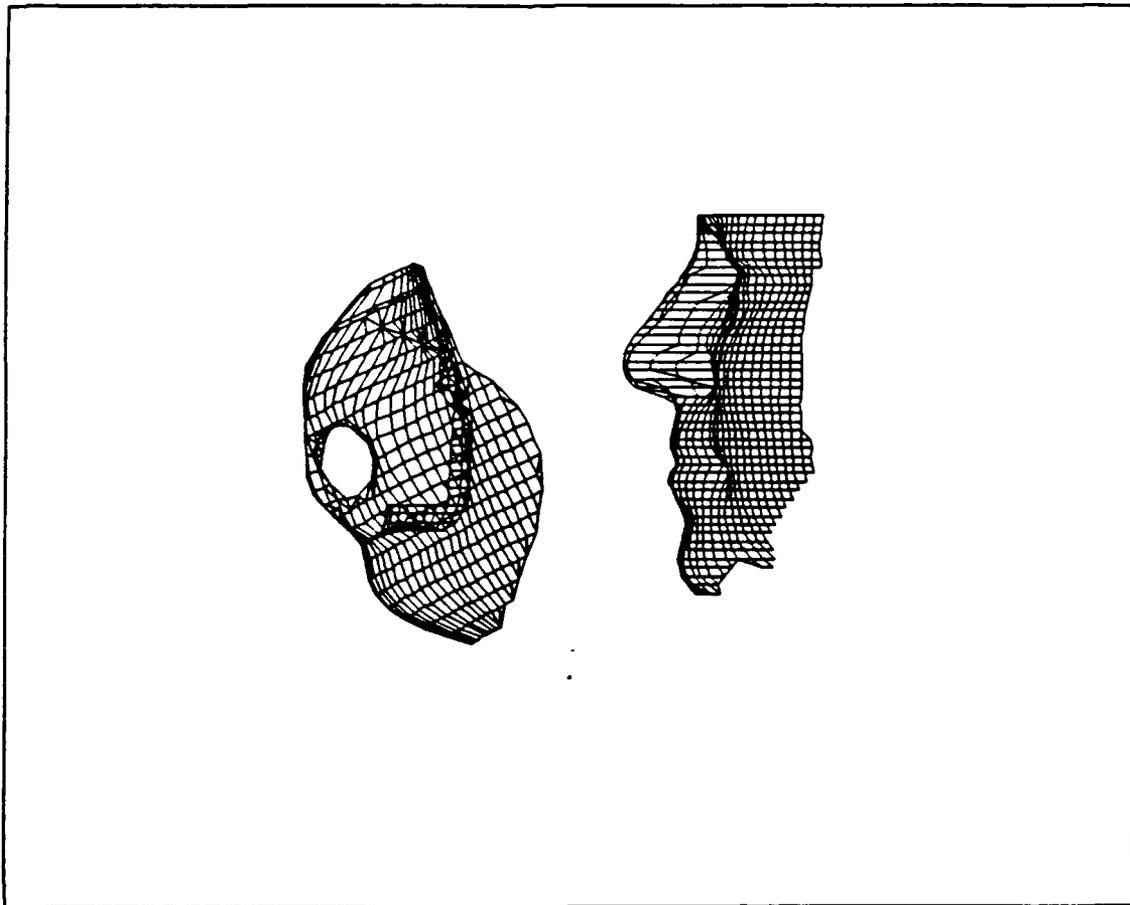


Figure 15. Synthesized Mask and Face Model

The face and mask models were still uncoupled and therefore their interaction could not be analyzed. They were initially connected by 20 gap elements from the face seal region of the mask in the negative z direction to the face. Additional discussion of the gap elements and connectivity is presented in section 6.0. The final model utilized 132 gap elements. The models at this point still needed material and physical properties, boundary conditions and loads defined.

4.2 *Physical Properties*

Since the mask was modeled completely by isotropic thin shell elements, the only additional geometric property which required was the thickness. The thickness at various points of the mask liner were measured with a micrometer and recorded. There were three different thicknesses - .09 inches (2.286mm) for the chin/cheek flap region, .08 inches (2.032 mm) for the area (mostly covered by the hardshell) which attaches to the oxygen hose and .03 inches (.762mm) for the seal region. Depending on boundary conditions, the rigid shell which fits over the liner could be modeled and assigned a thickness. However, since the material properties are many orders of magnitude stiffer than the liner, it can be assumed nearly rigid, and all deformation will take place only in the flexible rubber inner liner. Finally, Gap elements were used to model the coupling of the mask to the face (Figure 16).

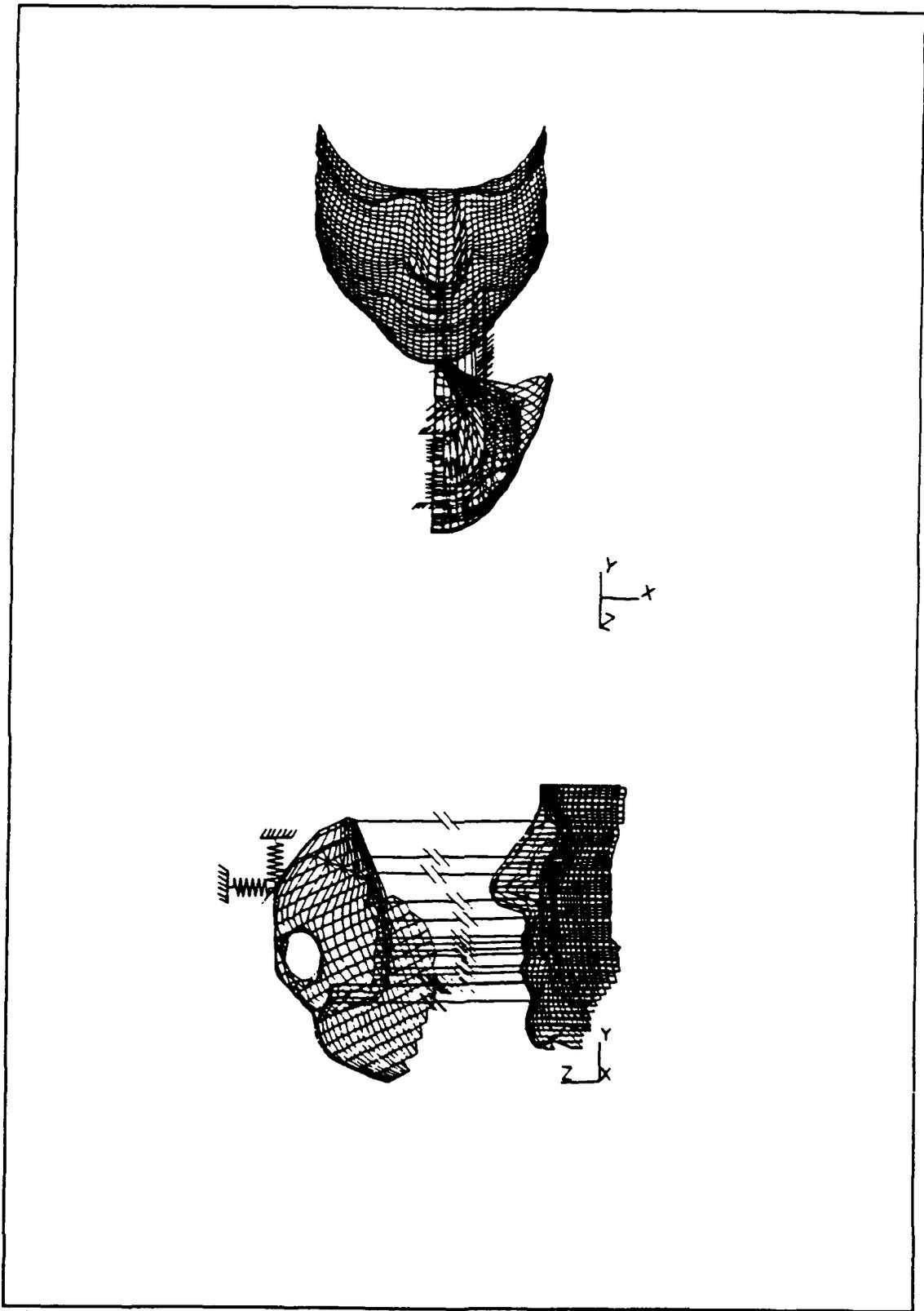


Figure 16. Gap Elements Connect Mask and Face

Properties for the gap elements include gap direction and gap distance. The mask was assumed to be pushed in the negative z coordinate direction by the force of the tightening straps and therefore the gap direction is -z (negative z direction). Some caution must be exercised when using gap elements if forces are not in the gap direction since once the distance in the gap direction has been reached, the gap is considered closed (see Appendix M for a brief discussion of gap elements). When using gap elements, the model connected to the gaps must not be singular (unconstrained) if the gaps were removed. A node-to-ground spring was used to constrain a point on the rigid shell of the mask. The spring was given a stiffness several orders of magnitude lower than any other modulus in the model and allowed rigid body motion of the mask as it slid back toward the face until it was constrained by the closing of gap elements.

4.3 Material Properties

The material of the mask's flexible rubber liner is silicone latex. Mil-M-87163A (Mil Spec for MBU-12 mask for which the MBU-20 is the replacement) requires a minimum hardness of between 40 and 50.(3,14) This corresponds to an elasticity modulus between 200 and 400 pounds per square inch, and a Poisson ratio of nearly .5. (4,247) The units chosen for the mask model were millimeters and milli-Newtons (IDEAS default set of units). The modulus of elasticity

used was 2000 mN/mm^2 (approximately 300 psi). The face was modelled as completely rigid and serves more as an aid to visualizing the mask deformation relative to the face than as an actual finite element model. The face is necessary for computing the distances between the mask seal region and the face. While a rigid face is not very realistic, a flexible face model was beyond the scope of the current effort.

The rigid shell which covers the soft flexible liner is a fiberglass epoxy laminate and serves two purposes. It transfers the force of the restraining straps to the soft liner to keep the mask against the face and it constrains the liner from expanding like a balloon due to positive pressure differential inside the mask. The small deformation of the shell relative to the deformation of the flexible liner is not of interest and therefore any modulus which is several orders of magnitude greater than the flexible rubber could be used: a value of 2.0×10^6 was chosen. An alternative approach would be to only model the flexible liner, and apply appropriate boundary conditions and loads to represent the effect of the shell on the liner. This approach was used when the connection of the mask to the face was modeled by an enforced displacement and no pressure differential was applied. The deformation of the liner away from the seal region was negligible and therefore

the constraint of the rigid shell was not required for that case.

5.0 Finite Element Analysis

The finite element analysis begins with the finite element model development discussed above. Forces and any additional restraints required were added to the model. Cases were defined to examine effects of various loads. The primary model responses of interest in this effort were the displacements of the mask seal relative to the face and the reaction forces on the face in the region mated to the seal. Both of these quantities give a measure of how good a seal is maintained.

5.1 Loads

Two types of forces exist on the mask and need to be accounted for in the model. There are concentrated forces due to the retaining straps securing the mask on the face, and there are distributed pressure forces due to the pressurized flow of oxygen into the mask. The retaining straps are riveted to the rigid shell portion of the mask at 4 points symmetrically located about the YZ plane of symmetry of the mask. The straps hook into the helmet on either side of the face and exert a force component back toward the face (negative z direction) and a component laterally (positive and negative x direction). Since the lateral forces are symmetric about the yz plane, (and they

are connected to a 'rigid' hard shell structure) the lateral components essentially cancel, leaving only a force in the negative z direction. This force acts at two locations on a symmetric half of the mask (one near the top of the hard shell and the other near the bottom). The force exerted will vary depending on the amount of contour mismatch between the mask and face, but will be whatever is required for the pilot to achieve a good seal. The lengths of the two straps are independently adjustable, so the two forces need not be the same (moments can be induced). Various forces were experimented with but, the results obtained (reaction forces) were inconsistent and another approach was taken. The force of the tightening straps was modeled by an enforced displacement of the rigid shell, allowing the seal region to deform. This was done for the unpressurized mask and stress and strain contours were obtained on the seal region.

The distributed pressure forces on the mask can be a maximum of 80mm of Hg (approximately 10 mN/mm^2). (2,18) These pressure forces are cancelled by the rigid shell and retaining straps except where the rigid shell does not surround the pressurized portion of the liner - the seal region and a small region at the top of the liner where it meets the bridge of the nose. Figure 17 shows a pressure force applied to this region, with all other elements (except gap elements) not displayed. Using the enforced

displacement of the seal to conform to the face has a limited usefulness. It can only be used for the unpressurized case. To see if the seal would leak under typical PPB pressures, a model using gap elements at every node (there were 132 of them) of the seal region was used.

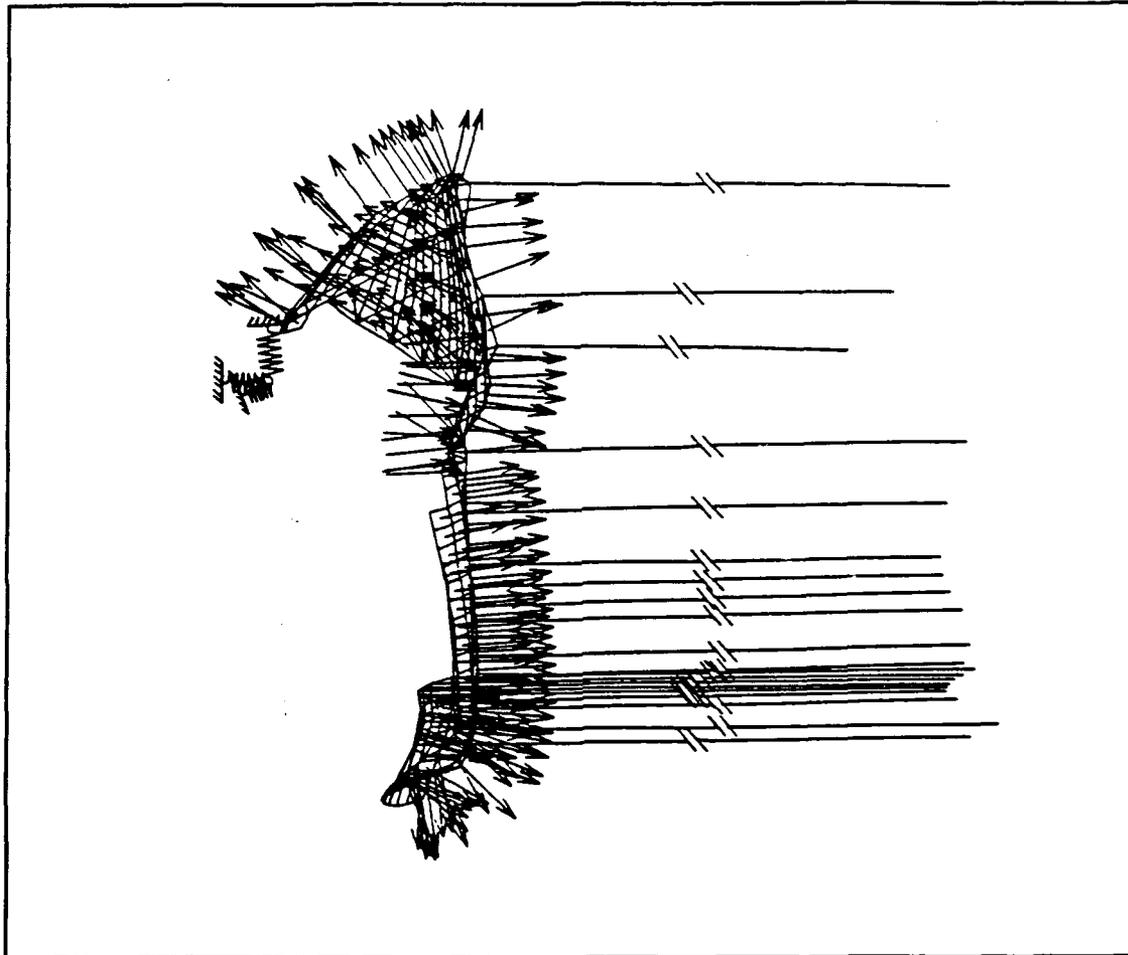


Figure 17. Net Pressure Forces

5.2 Boundary Conditions (Restraints)

Due to the use of symmetry, restraints must be used to prevent incompatible displacements (shears) across the plane of symmetry. These nodes were restrained in the x direction

translation and z rotation. Translation in Y was also restrained because induced moments acting on the weak spring allowed the mask to make a large displacement in the y direction (Figure 18).

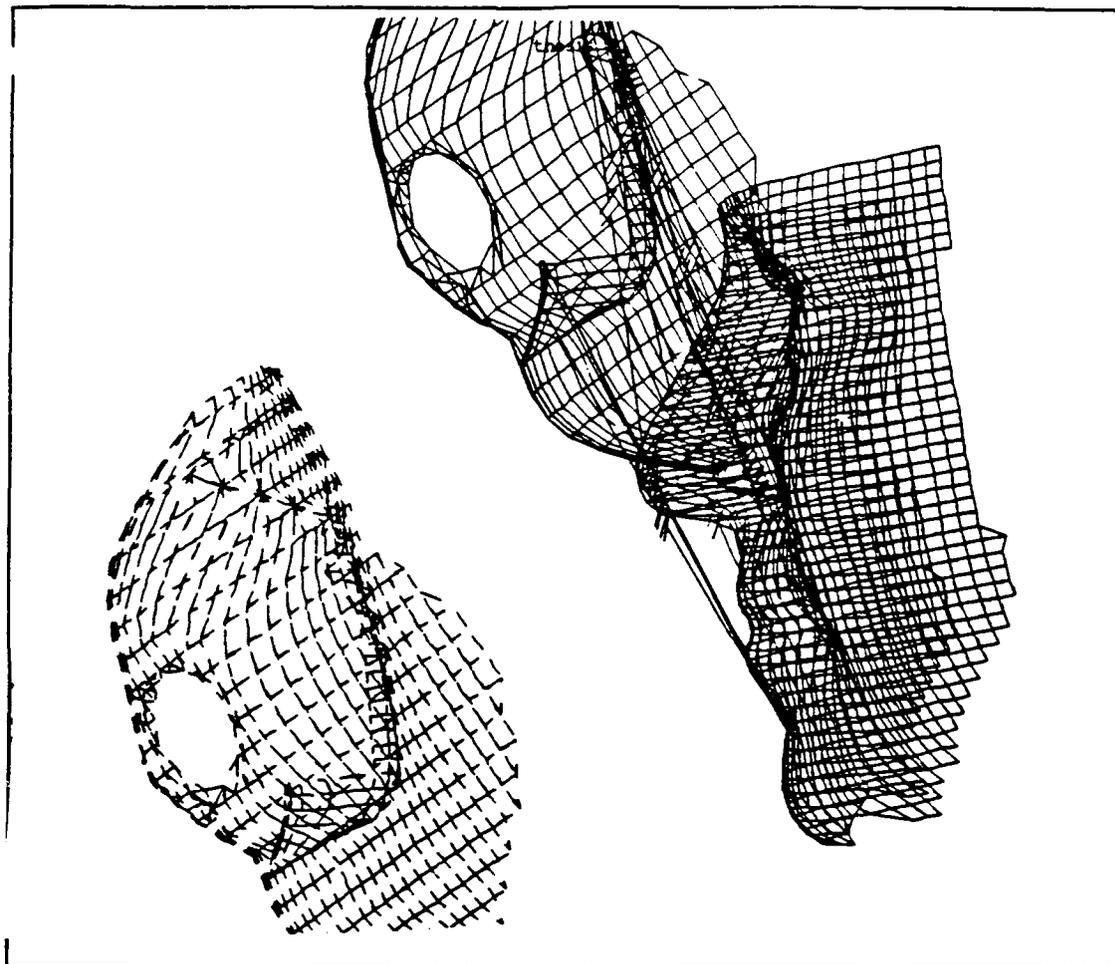


Figure 18. Large 'Rigid Body' Displacement of Mask

The y translation constraint became unnecessary when the restraining strap forces were eliminated in favor of an enforced displacement approach.

6.0 Results / Conclusions

The logic of the initial runs was to apply a force of about 1 pound to the mask shell in the negative z-direction and look at the reaction forces on the face and then steadily increase that force on successive runs. It was expected that not all points on the face (which had gap elements connecting them to the mask seal) would show reaction forces initially. This would mean that the mask did not form a perfect seal at that amount of force from the tightening straps. Successive runs would be done with incrementally greater forces until the results showed that the mask was sealed to the face (reaction forces at all face nodes connected by gap elements). Then, the internal pressure could be applied incrementally and the degradation of the seal noted. This would require more tightening force, which could be applied as well. This approach was followed for several values of tightening strap loads. However, the results were inconsistent and not all points sealed no matter what the force level. Even though 20 gap elements were used, the tightening forces were reacted primarily at only a small subset of the total number of nodes with gap elements. This is not necessarily all bad, given the assumptions of a rigid face and it does mirror the real life situation to the extent that with a mismatch of the seal and face contours, certain spots will contact with more pressure, making for an uncomfortable situation for the

pilot. The face which was digitized was that of the author of this thesis. He did notice a significant mismatch of the mask seal area between the chin and lower lip when he tried on the mask. Also, as the straps were tightened, there was an unequal distribution of forces around the seal, with more force at the top and side/bottom, and less on the sides and center bottom. An internal pressure was applied (in the finite element analysis) to the areas of the mask which are pressurized and are not restrained by the hard shell area. The mask took on an unrealistic displacement because a large pressure was applied to elements (on the seal region) whose nodes were not constrained by GAP elements. This showed that gap elements would have to be defined at every node on the seal region (requiring 74 gap elements to be defined for this coarse model). Gap element definition is a tedious job, since gap definition in I-DEAS requires about 10 steps with selections from several different levels of menus for each gap element. If the rigid shell contacts a rigid face and they are constrained to move in the direction of contact, no additional displacement will occur, no matter how much force is applied. Therefore, it was important to not define any gap elements from the rigid shell to the face at the boundary where the seal region and rigid shell meet.

Another approach was tried - give the rigid shell an enforced displacement back toward the face and look at reaction forces on the face. This approach makes good sense

intuitively, since the pilot will tighten the mask to his face to achieve a specific displacement of the mask to his face. Then internal pressure forces were applied. The same problem of using too few gap elements arose. Some parts of the seal would make contact (where gap elements exist) but other parts were free to deform unrealistically through the face. This created unrealistically large reaction forces at

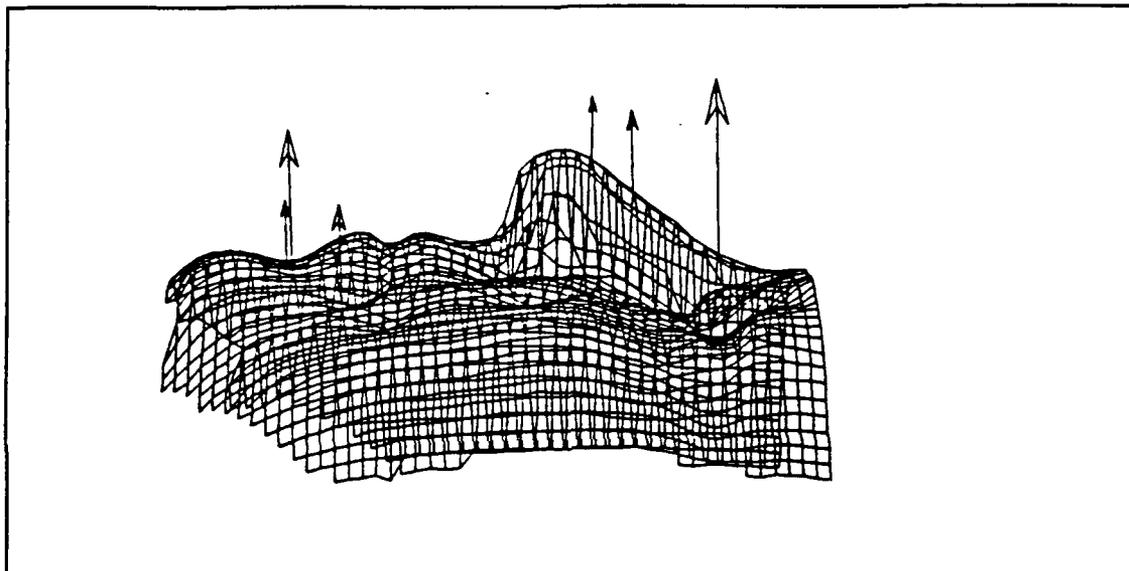


Figure 19. Reaction Forces on Face

certain points. It was suspected that the finite element mesh was too coarse in the area of large strain gradients along the complex curvature of the seal region. A way to test this hypothesis was to give the seal region an enforced displacement which caused the seal to conform exactly to the face. Strain and stress contours could be plotted to see if large gradients exist on the seal region. Figure 20 shows the seal region conforming to the face. As in the case of gap elements, definition of enforced displacements of one

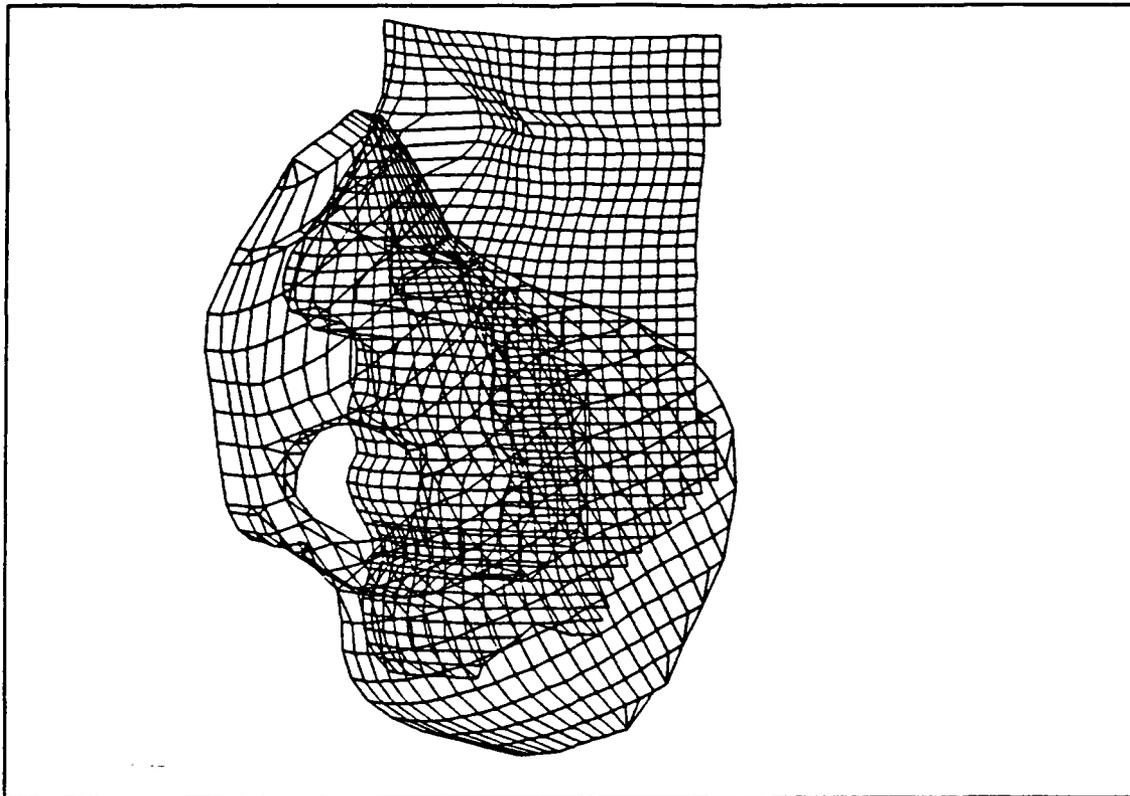


Figure 20. Enforced Conforming Displacement of Seal

model (the mask seal) to conform to another model (the face) would require a large manual effort of measuring the distances between nodes. A FORTRAN program was written called DISPLZ (see Appendix H) to automate that function. Stress and strain contours were plotted and large gradients were found on the seal region (Figure 21) indicating the need for mesh refinement. A more refined mesh on the seal was not too difficult to obtain using SCANCAD and I-DEAS. However, defining the transition mesh to the rest of the mask (using a new more refined model) would require even more manual time and effort and be subject to errors.

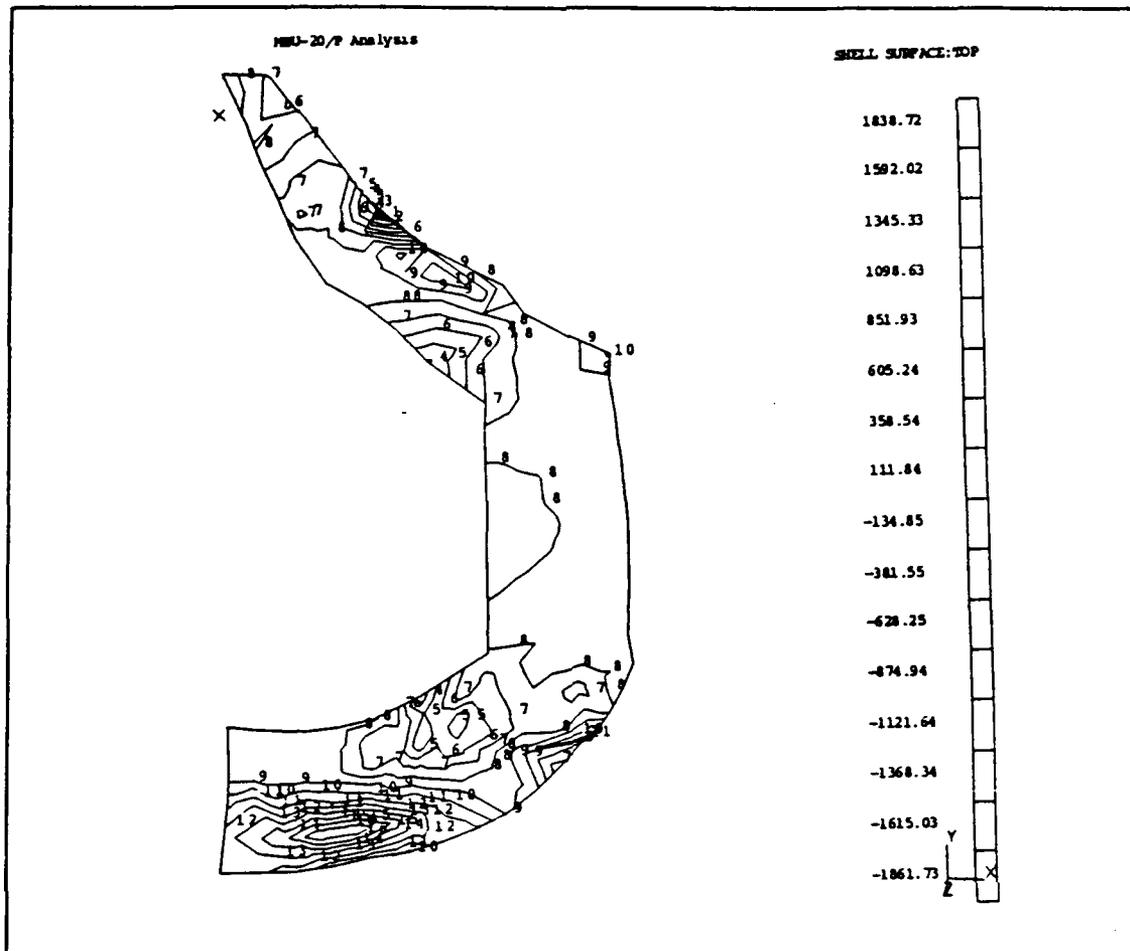


Figure 21. Stress Contours on Seal Region

The large stress gradient towards the top area of the seal is an artificial artifact of the mating of the seal to the face. The GAPZ and DISPLZ programs were used to pick points on the face in as close of a direct line back from nodes on the seal to form the connection. This region of the seal will initially move back toward the face and contact points on the side/front of the nose. The seal will then slide back and laterally along the nose till it reaches its final position. For the finite element model however, once the seal node makes contact with a face node, it cannot move

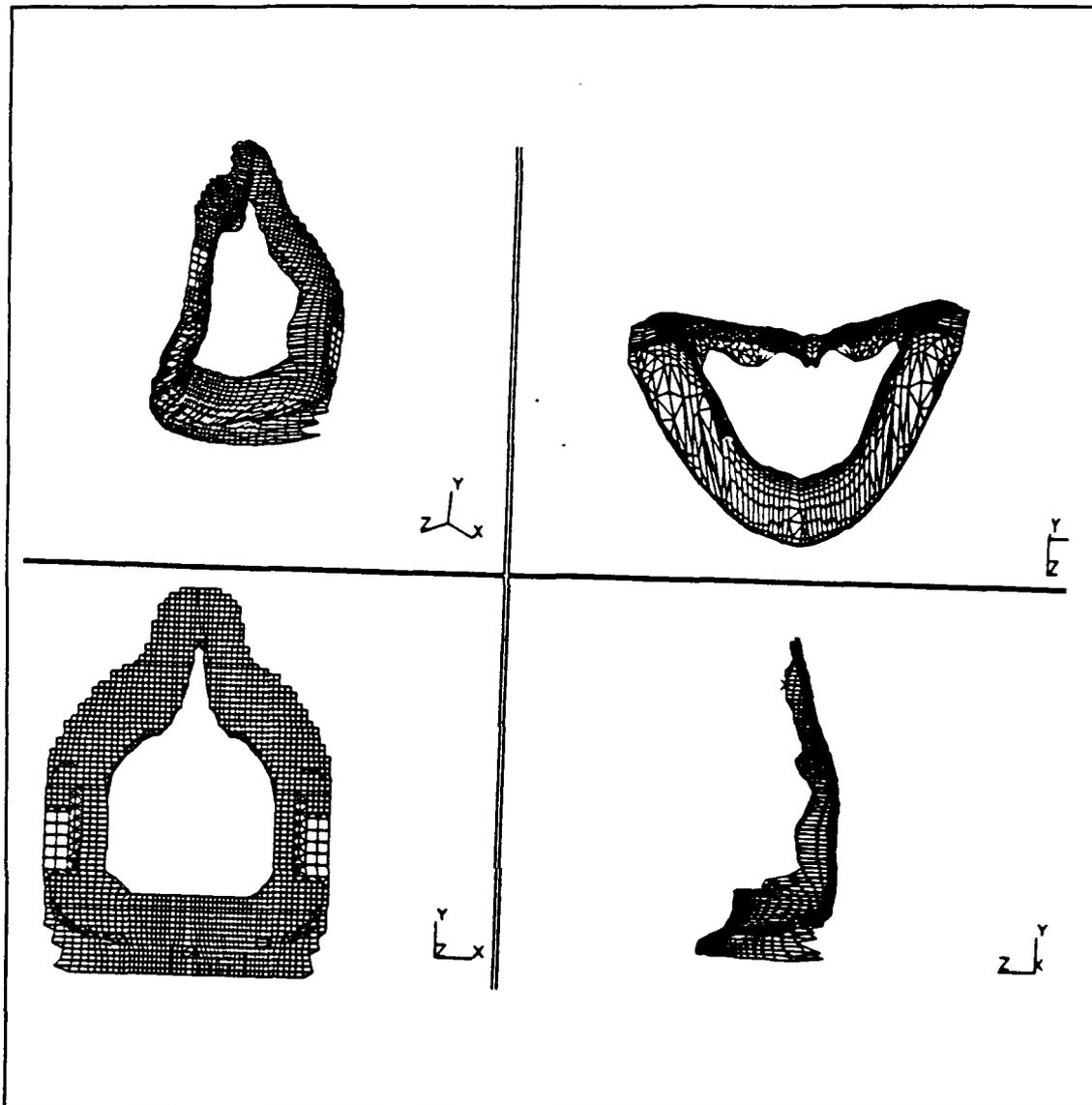


Figure 22. Refined Mesh on Seal Region

further. As the rest of the mask continues to move back onto the face, large stresses develop on the contact area of the seal to the nose. A possible solution is to delete the nose (and any other extraneous areas) from the face before input to DISPLZ or GAPZ.

A final methodology was tried. The mask seal was given an enforced conforming displacement to the face. The

displacement was specified in the Z direction only so slipping in the X and Y directions could occur. Reaction forces were calculated as well as deformed geometry. The deformed geometry and reaction forces were input as the starting point for the next run, including gap elements and internal pressure forces.

In summary, the following load cases were run:

- a) 20 gap elements, no internal pressure or internal pressure applied, 1 to 10 pounds of tightening strap forces. (Inconsistent results)
- b) No gap elements, enforced displacement of seal to conform to face. (Stress contours/gradients indicate need for mesh refinement).
- c) 74 gap elements, no internal pressure, enforced displacement of hard shell back toward face. (Reaction forces and displacements calculated).
- d) 74 gap elements, PPB pressure of 10mN/mm^2 applied to area not constrained by hard shell. (Reaction forces and displacements calculated).
- e) 132 Gap elements, initial deformed shape specified and element forces, plus internal pressure.

It is interesting to compare the reaction force distribution from case c and d. Case c is shown in Figure 23. Case d in Figure 24. In both cases, the hard shell was given the same enforced displacement back toward the face to ensure contact of the seal and face (Figure 26). In case c contact along the side of the face is superficial as well as intermittent spots following the seal region up to the bridge of the nose. There is a large concentration of force

(hot spot) under the lower lip near the side, and the contact forces taper off toward the plane of symmetry at the middle of the face. In case d, the magnitude of the forces is greater, since there is internal pressure and the mask remains tightened to the face. More points appear to be in contact due to the internal pressure pushing the seal toward the face. Still, hot spots still exist where there is more force being reacted due to uneven fit. The mask however does appear to provide a good seal at this high internal pressure. Deformed geometry is shown in Figure 26 along with the undeformed shape (e).

Many useful tools and techniques were developed for developing finite element models from digitized data of 3 d objects. Many features of the IDEAS program were experimented with to see which could be useful for model creation and analyses required in this project. Applying the tools and techniques to the analysis of the MBU-20/P mask was still difficult and/or tedious for some tasks and further tool development is necessary. Inclusion of realistic (flexible) facial modeling is still required and improved methods of mesh refinement and model combination, especially where complex curved surfaces intersect, are needed. For this reason the possible utility of surface representation was revisited in Appendix B. A more refined model of the mask (at least in the seal region) should be developed for detailed analyses. Friction and sliding of

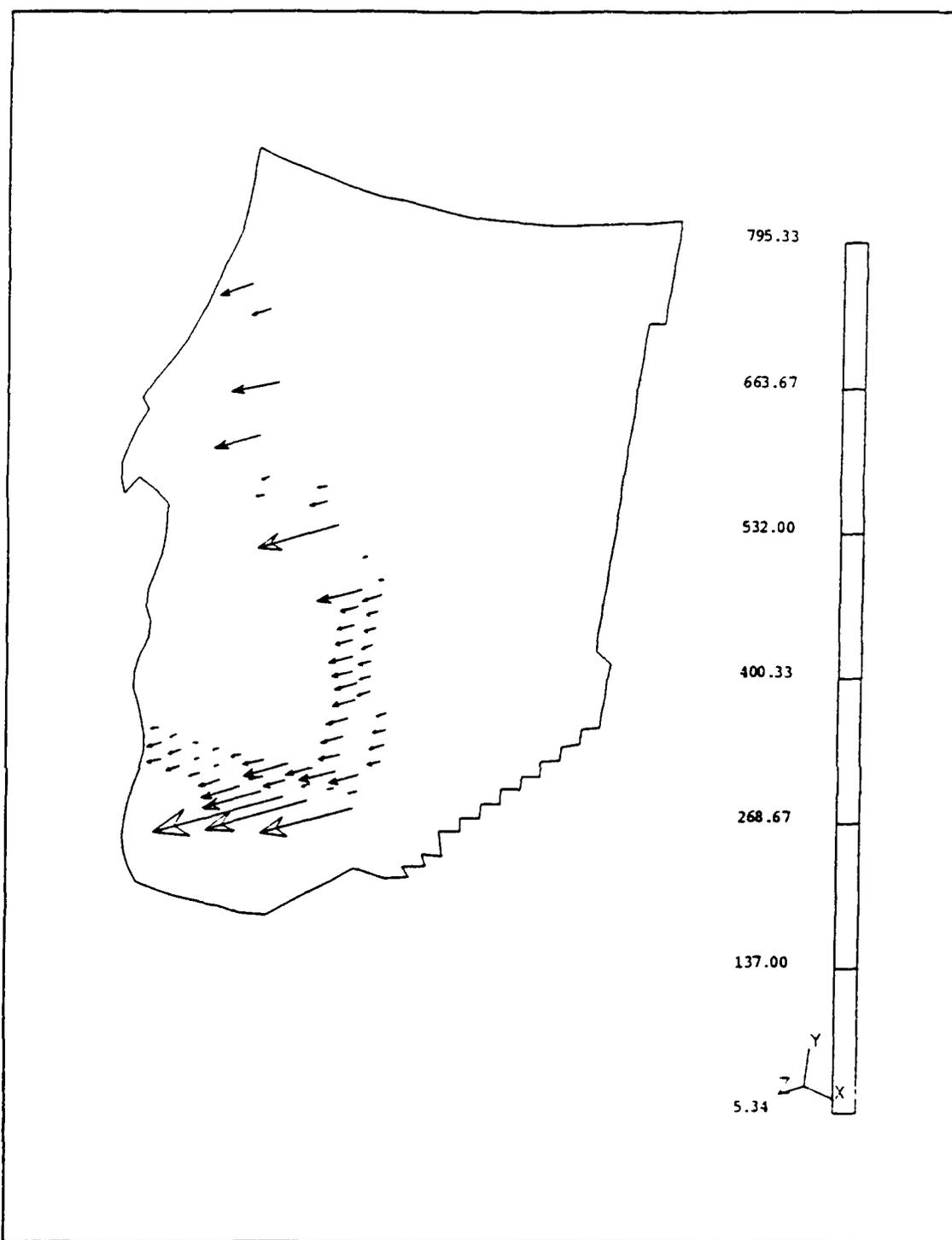


Figure 24. Reaction Forces due to PPB Pressures

and models for accomplishing the goal of being able to do

finite element analysis of complex digitized shapes such as the MBU-20/P mask.

7.0 **Recommendations:** Suggestions for Further Work

- *Enhancements to Computer Interface Program SCANCAD*

- Adapting the SCANCAD program for a Linear Scanner device rather than the rotary device worked with in this effort. Connectivity is based on sweeping through the latitudes and then the longitudes. It could be modified to sweep through x then y.

- Surfaces based on Intelligent Algorithms which would minimize the total number of surfaces required to describe the object and allow for automatic or manual meshing

- Interface to Other CAD Programs

SCANCAD could be easily modified for other formats

- Inclusion of 3D solid elements based on data from medical scanners such as MR (magnetic resonance) and CT (Computed Tomography).

- Embed the SCANCAD program within the CAD program (such as IDEAS which has it's own programming language (IDEAL) and

database management system (Pearl). The advantage here is that the graphics capabilities of the CAD program are available during running of the SCANCAD program allowing virtually instant visual feedback on the results of the choices you make within the SCANCAD program.

- Automate transition mesh formation at intersection of surfaces. This is currently very labor intensive and subject to errors.

Mask Analysis

- Develop flexible face model
- Develop refined mask model
- Compare linear and nonlinear analysis (PATRAN or NASTRAN vs. I-DEAS)
- Experiment with other schemes for coupling the face and mask
- Account for friction and sliding of mask on face

This is a nonconservative action requiring nonlinear analysis methods.

- Dissect mask liner along plane of symmetry and scan to see if better quality scan results (fewer shadow regions)

This Page is Blank

Appendix A. PATRAN vs. Ideas

Appendix A. PATRAN vs. Ideas

Similarities and Differences Relevant to this Project

Similarities:

- ASCII command files may be created (automatically or with an editor). In PATRAN it is a '.SES' file. In Ideas it is a '.PRG' file.

- Binary Model files may be created (using a SAVE command). In PATRAN it is a '.DAT' file. In Ideas it is the '.MF1' and '.MF2' file.

- Compact ASCII model files may be created. In PATRAN it is called a neutral file, usually labelled '.OUT'. In Ideas it is the universal file, usually '.UNV'.

Differences:

- Surface representation. PATRAN uses isoparametric bicubic surfaces requiring 48 coefficients (16 for each X, Y and Z). Ideas uses a NURBS (Non Uniform Rational B-Splines) representation requiring knots, weights and control points also requiring a large number of coefficients to describe it depending on the order of the polynomials in each parametric

direction. The NURBS representation is more general but it is also more complicated to implement.

PATRAN has a nonlinear analysis capability (geometric and/or material). I-DEAS does not.

Appendix B: Derivation of Algorithms for BiCubic Surfaces

*Derivation of Algorithm for Patran Neutral File
Surface Representation*

Patran utilizes isoparametric bicubic surfaces (with the origin of the parametric axes at a corner grid). These surfaces may be uniquely determined by 16 three dimensional locations (grid points). The surface is represented by 48 coefficients. The 48 geometric coefficients needed are:

$$(1) V(0,0) \quad (5) V(0,1) \quad (9) \frac{\partial V}{\partial \xi_2}(0,0) \quad (13) \frac{\partial V}{\partial \xi_2}(0,1)$$

$$(2) V(1,0) \quad (6) V(1,1) \quad (10) \frac{\partial V}{\partial \xi_2}(1,0) \quad (14) \frac{\partial V}{\partial \xi_2}(1,1)$$

$$(3) \frac{\partial V}{\partial \xi_1}(0,0) \quad (7) \frac{\partial V}{\partial \xi_1}(0,1) \quad (11) \frac{\partial^2 V}{\partial \xi_1 \partial \xi_2}(0,0) \quad (15) \frac{\partial^2 V}{\partial \xi_1 \partial \xi_2}(0,1)$$

$$(4) \frac{\partial V}{\partial \xi_1}(1,0) \quad (8) \frac{\partial V}{\partial \xi_1}(1,1) \quad (12) \frac{\partial^2 V}{\partial \xi_1 \partial \xi_2}(1,0) \quad (16) \frac{\partial^2 V}{\partial \xi_1 \partial \xi_2}(1,1)$$

The V is used to represent first X, then Y, then Z. The determination of these 48 coefficients is based on using 16 Grid points in a 4 by 4 uniform parameterization (each point is separated from its neighbor in each parametric direction by 1/3). Computing the 16 coefficients based on the x coordinates leads to 16 equations and 16 unknowns. All 16 equations are based on the following single equation which

was arrived at by keeping all terms with exponents up to and including 3 (of a Pascal's triangle).

$$\begin{aligned}
 x_i = & a_0 \\
 & + a_1 \xi_1 + a_2 \xi_2 \\
 & + a_3 \xi_1^2 + a_4 \xi_1 \xi_2 + a_5 \xi_2^2 \\
 & + a_6 \xi_1^3 + a_7 \xi_1^2 \xi_2 + a_8 \xi_1 \xi_2^2 + a_9 \xi_2^3 \\
 & + a_{10} \xi_1^3 \xi_2 + a_{11} \xi_1^2 \xi_2^2 + a_{12} \xi_1 \xi_2^3 \\
 & + a_{13} \xi_1^3 \xi_2^2 + a_{14} \xi_1^2 \xi_2^3 \\
 & + a_{15} \xi_1^3 \xi_2^3
 \end{aligned}$$

...where the ξ_i are evaluated at the parametric location of the x_i . For Instance:

$$x_1 = x(0,0) = a_0$$

$$x_2 = x(1/3,0) = a_0 + a_1/3 + a_3/9 + a_6/27$$

etc.

This may be put into a matrix form:

$$\{x_i\} = [A_{ij}] \{a_j\}:$$

The full A matrix is shown on the next page. Assume that we have the coordinates of the 16 grid points making up the surface. We can determine the 16 coefficients a_i by inverting the A matrix and using that in the following matrix equation:

$$\{a_i\} = [A_{ij}]^{-1} \{x_j\} \quad i=1,2,\dots,16; \quad j=1,2,\dots,16$$

$$\{b_i\} = [A_{ij}]^{-1} \{y_j\}$$

The $[A]^{-1}$ Matrix

Knowing the coefficients, will allow one to calculate the required inputs to Patran:

$$\begin{aligned} \frac{\partial x}{\partial \xi_1} &= a_1 + 2a_3\xi_1 + a_4\xi_2 + 3a_6\xi_1^2 + 2a_7\xi_1\xi_2 + a_8\xi_2^2 + 3a_{10}\xi_1^2\xi_2 \\ &\quad + 2a_{11}\xi_1\xi_2^2 + a_{12}\xi_2^3 + 3a_{13}\xi_1^2\xi_2^2 + 2a_{14}\xi_1\xi_2^3 + 3a_{15}\xi_1^2\xi_2^3 \\ \frac{\partial x}{\partial \xi_2} &= a_2 + a_4\xi_1 + 2a_5\xi_2 + a_7\xi_1^2 + 2a_8\xi_1\xi_2 + 3a_9\xi_2^2 + a_{10}\xi_1^3 + 2a_{11}\xi_1^2\xi_2 \\ &\quad + 3a_{12}\xi_1\xi_2^2 + 2a_{13}\xi_1^3\xi_2 + 3a_{14}\xi_1^2\xi_2^2 + 3a_{15}\xi_1^3\xi_2^2 \\ \frac{\partial^2 x}{\partial \xi_1 \partial \xi_2} &= a_4 + 2a_7\xi_1 + 2a_8\xi_2 + 3a_{10}\xi_1^2 + 4a_{11}\xi_1\xi_2 + 3a_{12}\xi_2^2 + 6a_{13}\xi_1 \\ &\quad + 6a_{14}\xi_1\xi_2^2 + 9a_{15}\xi_1^2\xi_2^2 \end{aligned}$$

x	(0,0)	=	a ₀	=	x ₁
x	(1,0)	=	a ₀ + a ₁ + a ₃ + a ₆	=	x ₄
x, ξ ₁	(0,0)	=	a ₁		
x, ξ ₁	(1,0)	=	a ₁ + 2a ₃ + 3a ₆		
x	(0,1)	=	a ₀ + a ₂ + a ₅ + a ₉	=	x ₁₃
x	(1,1)	=	Σa _i i=0 to 15	=	x ₁₆
x, ξ ₁	(0,1)	=	a ₁ + a ₄ + a ₈ + a ₁₂		
x, ξ ₁	(1,1)	=	a ₁ + 2a ₃ + a ₄ + 3a ₆ + 2a ₇ + a ₈ + 3a ₁₀ + 2a ₁₁ + a ₁₂ + 3a ₁₃ + 2a ₁₄ + 3a ₁₅		
x, ξ ₂	(0,0)	=	a ₂		
x, ξ ₂	(1,0)	=	a ₂ + a ₄ + a ₇ + a ₁₀		
x, ξ ₁ ξ ₂	(0,0)	=	a ₄		
x, ξ ₁ ξ ₂	(1,0)	=	a ₄ + 2a ₇ + 3a ₁₀		
x, ξ ₂	(0,1)	=	a ₂ + 2a ₅ + 3a ₉		
x, ξ ₂	(1,1)	=	a ₂ + a ₄ + 2a ₅ + a ₇ + 2a ₈ + 3a ₉ + a ₁₀ + 2a ₁₁ + 3a ₁₂ + 2a ₁₃ + 3a ₁₄ + 3a ₁₅		
x, ξ ₁ ξ ₂	(0,1)	=	a ₄ + 2a ₈ + 3a ₁₂		
x, ξ ₁ ξ ₂	(1,0)	=	a ₄ + 2a ₇ + 2a ₈ + 3a ₁₀ + 4a ₁₁ + 3a ₁₂ + 6a ₁₃ + 6a ₁₄ + 9a ₁₅		

The same calculations can be repeated for y and z. Note that the 48 geometric coefficients must be calculated for each patch of 16 grids as well as the determination of the 48 coefficients a_i, b_i, c_i. The same [A]⁻¹ matrix is used throughout.

The 16 Grids would use a 4 by 4 mesh from the parametric mapping of points, similar to the way the algorithms were developed for four noded quad element connectivity. The implementation is left as follow on work.

Appendix C: Scan Limit Analysis Listing

Analysis of Scanned Data

Total Number of scan points	:	27257
Maximum number of Scan Latitudes	:	66.00000
Maximum number of Scan Longitudes	:	511
Points Required for Rectangular Mesh:		33726.00
Minimum scan Latitude	:	7
Maximum scan Latitude	:	72
Minimum scan Longitude	:	1
Maximum scan Longitude	:	511
Minimum scan Longitude	:	1
Maximum scan Radius	:	164
Minimum scan Radius	:	70

Appendix D: Sample Intermediate Data

Sample Data File

1	141	7.62	7.62	218.82	0.09
1	142	8.22	8.22	220.38	0.10
1	143	8.13	8.13	221.95	0.10
1	144	8.19	8.19	223.51	0.10
1	145	8.29	8.29	225.07	0.10
1	146	8.19	8.19	226.63	0.10
1	147	8.00	8.00	228.20	0.10
1	148	7.74	7.74	229.76	0.10
1	149	7.68	7.68	231.32	0.09
1	150	7.36	7.36	232.89	0.09
1	151	7.14	7.14	234.45	0.09
1	152	6.85	6.85	236.01	0.08
1	153	6.69	6.69	237.58	0.08
1	154	6.46	6.46	239.14	0.08

(...Data Deleted...)

510	156	5.63	5.63	242.26	-0.14
510	157	5.31	5.31	243.83	-0.13
510	158	4.70	4.70	245.39	-0.12
510	159	4.10	4.09	246.95	-0.10
510	160	3.20	3.20	248.52	-0.08
510	161	1.31	1.31	250.08	-0.03
511	141	8.51	8.51	218.82	-0.10
511	142	8.22	8.22	220.38	-0.10
511	143	8.29	8.29	221.95	-0.10
511	144	8.35	8.35	223.51	-0.10
511	145	8.29	8.29	225.07	-0.10
511	146	8.13	8.13	226.63	-0.10
511	147	7.90	7.90	228.20	-0.10
511	148	7.62	7.62	229.76	-0.09
511	149	7.52	7.52	231.32	-0.09
511	150	7.23	7.23	232.89	-0.09
511	151	6.91	6.91	234.45	-0.08
511	152	6.85	6.85	236.01	-0.08
511	153	6.53	6.53	237.58	-0.08
511	154	6.24	6.24	239.14	-0.08
511	155	5.76	5.76	240.70	-0.07
511	156	5.63	5.63	242.26	-0.07
511	157	5.31	5.31	243.83	-0.06
511	158	4.70	4.70	245.39	-0.06
511	159	3.97	3.97	246.95	-0.05
511	160	3.14	3.14	248.52	-0.04
511	161	1.31	1.31	250.08	-0.02
99999	1	1	1	1	1

**Appendix E: Procedures/Instructions for using the SCANCAD
Program**

Appendix E: Procedures/Instructions for using the SCANCAD Program

The following format was adopted for input record fields for the intermediate data file:

FIELD	VARIABLE (Array Size)	TYPE
Longitude	G(j) (50000)	Integer
Latitude	T(j) (50000)	Integer
Radius	R(j) (50000)	Real
X-Coordinate	X(j) (50000)	Real
Y-Coordinate	Y(j) (50000)	Real
Z-Coordinate	Z(j) (50000)	Real

The fields are freely formatted (delineated by a comma or blank). A segment of a typical input file is shown in appendix D. The end of file (EOF) identifier is a completely filled record with the first field having a value greater than 99000. The EOF record can be added to the data file with a text editor such as the EDT editor on the VAX. SCANCAD is an interactive user-friendly program. You will be prompted for information. The following prompts will appear:

Prompt 1>

Enter the Name of the Input Data File [test.dat]:

Your Response>

You should enter the name of the intermediate data file containing the digitized coordinates. The file should conform to the format specified in the main body of text entitled 'Input Data'. A valid response is shown in square brackets.

After the program analyzes the input data, it will print the scan limits to the screen. The next prompt asks if you want it printed to a file as well.

Prompt 2>

Enter 1 - to write this Data to a File
 2 - to continue
 3 - to Exit

Your Response>

You may type a 1,2,or 3 followed by the return key.
Choosing 2, continues normal execution of the program.
Choosing 3 exits the program. Choosing 1 causes the program
to prompt you for a file name to write the data to.

Prompt 3>

Menu of Output Formats
1 - SDRC Ideas
2 - PDA PATRAN
3 - Both
4 - Quit

Your Response> Enter a 1,2,3 or 4. 4 exits the program.
1,2,3 continue by requesting you specify filenames for the
various files to be created. Typical prompts follow:

Prompt 4>

Enter a name for Ideas universal file of Nodes

Your Response>

Type a filename including extension. Filenames may have up
to 8 characters followed by a period followed by 3
characters.

Prompt 5>

Enter a name for Ideas .UNV file of Elements

Your Response>

Type a name for the file you want to contain the element
connectivities.

Prompt 6>

Enter numbers n1,n2,n3,p4 - mesh will keep
every n1th to n2th to n3th latitude, with n2 being
positioned p4 of the way between, i.e. an increasing
/decreasing symmetric ramp function [1,5,1,.5]

Your Response>

Sample input is shown in square brackets. Reduced
resolution will result. You should first analyze the scan
before deciding on reduction. This can be done by first
choosing 1,1,1,.7 for full resolution of the scanner.

Prompt 7>

Enter numbers m1,m2,m3,q4 - mesh will keep
every m1th to m2th to m3th longitude, with m2 being
positioned q4 of the way between, i.e. a decreasing/
increasing ramp function biased to upper longitudes
[10,4,6,.7]

Your Response>

Sample input is shown in square brackets. Reduced resolution will result. See comments on Response 6.

The next prompts allow for selecting only a subset of the scanned data for conversion and output

Prompt 8>

Minimum Latitude is (minlat)
Enter new minimum latitude. Scanned points at lower latitudes will be excluded [50].

Your Response>

(minlat) is the value determined by the program when analyzing the input data. Enter an integer value such as shown in square brackets.

Prompts 9,10,11> similar to 8 but requesting maximum latitude value to be kept, minimum longitude, and maximum longitude. These 4 values define a parametrically rectangular area of interest.

Prompt 12>

Enter an offset starting label for Nodes [1000]

Your Response>

Since several component models may be created and then combined, you might want to designate a starting ID number for each component to avoid duplicate element labels.

Prompt 13>

Enter a Scale factor for contour map (0=flat)

Your Response> Any value to give a third dimension to parametric 2 D map for visualizing where different features are located.

Prompt 14>

Enter a Material Property Number

Your Response>

Enter an integer value for the associated material which will be defined in the FEM program.

Prompt 15>

Enter a minimum Color Number (1 thru 15)

Your Response>

Enter a color number between 1 and 15 to color your elements as an aid to visualization. You may want different components colored differently.

Prompt 16>

Enter a maximum Color Number (1 thru 15)
(choose the same as minimum color for single color
for all elements)

Your Response>

Together with 15 it defines the range of colors to be used
for the elements created.

Prompt 17>

Color Patterns

- 1 - Vertical Striping
- 2 - Horizontal Striping
- 3 - Radial Distance Color Contours

Enter a 1,2 or 3. (Choice 3 overrides previous color
choices)

Your Response>

Allows for various color patterns, switching at each new
longitude (choice 1) each new latitude (choice 2) or taking
the average of the colors of the nodes it connects (choice
3) which is based on radial distance.

Prompt 18>

Enter a new offset starting label for elements [0]

Your Response>

Any integer value, 0 to start element labels at 1.

*At this point, the three files have been created, and
SCANCAD prints the message below:*

Message>

Universal File of Nodes is : (filename)
Universal File of Elements is : (filename)
Universal File of Nodes mapped to 2-D is: Flat.unv

You are now ready to run I-DEAS.

Appendix F: SCANCAD Program Listing

PROGRAM SCANCAD

```
Implicit integer (g,t)
common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
common
/BLK3/mcolr(511,300),JMINLAT,JMAXLAT,JMINSCAN,JMAXSCAN
common felemsi,fnodesi,felemsp,fnodesp
COMMON
/BLK4/MINSCAN,MAXSCAN,MINLAT,MAXLAT,MINRAD,MAXRAD,NUMPTS
common /BLK5/g(50000),t(50000)
```

C reads in scanned data in ascii form

c Print Title Screen/Menu

CALL INTRO

c Read in the Scanned Data

CALL INSCAN

c Put data into rectangular array

CALL RECTANG

c print out analysis of scanned data

CALL SCANAL

c get user to specify output files

CALL OUTSPEC

C cycle through again?

CALL REDO

END

C-----

SUBROUTINE REDO

```
1 print 2
write(*,*) ' 1 - Continue '
write(*,*) ' 2 - Quit '
do 385 ijk=1,5
print 2
385 continue
```

```
write(*,*)'Press the 1 to continue, the 2 to quit'
read(*,*) nquitk
if(nquitk.eq.2) goto 3
```

```

        CALL OUTSPEC
2   format(' ')
3   RETURN
   END
C-----
--

SUBROUTINE OUTSPEC
  Implicit integer (g,t)
  DIMENSION BP(4)
  common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
  common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
  common
/BLK3/mcolr(511,300),JMINLAT,JMAXLAT,JMINSCAN,JMAXSCAN
  common felemsi,fnodesi,felemsp,fnodesp
  COMMON
/BLK4/MINSCAN,MAXSCAN,MINLAT,MAXLAT,MINRAD,MAXRAD,NUMPTS
  common /BLK5/g(50000),t(50000)
C   DIMENSION BP(18)
  character*15 felemsi,felemsp,fnodesi,fnodesp
c   Give menu of outputs
  do 57 j=1,10
  print 2
  57 continue
  write(*,12)'
Formats                                     Menu of Output
  do 58 j=1,5
  print 2
  58 continue
  write(*,*) '                               1- SDRC Ideas
  ,
  write(*,*) '                               2- PDA PATRAN
  ,
  write(*,*) '                               3- Both
  ,
  write(*,*) '                               4- Quit

  print 2
  print 2
  print 2
  write(*,*) '                               Enter a 1,2,3 or 4
  ,
  read(*,*) mout

C - NEED TO INCLUDE SOME SUB MENUS AND BREAK THIS MODULE UP

  if(mout.eq.4)stop
  if((mout.gt.4).or.(mout.lt.1)) goto 57
  if((mout.eq.1).or.(mout.eq.3))Then
  print 2
  WRITE(*,*) 'ENTER a NAME for Ideas Universal File
of Nodes'

```

```

        READ(*,'(A15)') FNodesi
        open(8,file=FNodesi,status='NEW')
        print 2
        WRITE(*,*) 'ENTER a NAME for Ideas Unv. File of
Elements'
        READ(*,'(A15)') Felemsi
        open(6,file=Felemsi,status='NEW')
        print 2
        open(9,file='Flat.unv',status='NEW')
        endif
        if((mout.eq.2).or.(mout.eq.3))Then
        print 2
        WRITE(*,*) 'ENTER a NAME for Patran Neutral File
of Nodes'
        READ(*,'(A15)') FNodesp
        open(10,file=FNodesp,status='NEW')
        print 2
        WRITE(*,*) 'ENTER a NAME for Patran Neutral File
of Elements'
        READ(*,'(A15)') Felemsp
        open(11,file=Felemsp,status='NEW')
        open(12,file='Flat.nut',status='NEW')
        print 2

        endif

```

c nodes

```

        write(8,47)'      -1'
        write(8,47)'      15'
        write(9,47)'      -1'
        write(9,47)'      15'

```

c put x,y,z points into rectangular grid and then ask
user to specify
c reduction level

```

        write(*,*)'Enter numbers n1,n2,n3,p4 - mesh will keep
every nlth '
        write(*,*)'to n2th to n3th latitude, with n2 being
positioned'
        write(*,*)'p4 of the way between, i.e. an
increasing/decreasing'
        write(*,*)'symmetric ramp function [1,5,1,.5]'
        read(*,*) nskip,nskip2,nskip3,p4

```

```

        write(*,*)'Enter numbers m1,m2,m3,q4 - mesh will keep
every mlth'
        write(*,*)'to m2th to m3th longitude, switching at
q4 of the'
        write(*,*)'way between, i.e. a decreasing, increasing
ramp, '
        write(*,*)'biased toward the upper longitudes

```

```

[10,4,6,.7]'
  read(*,*) mskip,mskip2,mskip3,q4

      write(*,*) 'Minimum Latitude is ',jminlat
  write(*,*) 'Enter new Minimum Latitude, (Scanned points
at lower '
  write(*,*) 'latitudes will be excluded [50]''
  read(*,*) minlat
  if(minlat.lt.1) minlat=1
  maxlatt=maxlat
  write(*,*) 'Maximum Latitude is ',jmaxlat
  write(*,*) 'Enter new Maximum Latitude, (Scanned points
at higher '
  write(*,*) 'latitudes will be excluded [100]''
  read(*,*) maxlat
c   if(maxlat.gt.maxlatt) maxlat=maxlatt
      write(*,*) 'Minimum Longitude is ',jminscan
  write(*,*) 'Enter new Minimum Longitude '
  read(*,*) minscan
  if(minscan.lt.1) minscan=1
  maxscann=maxscan
      write(*,*) 'Maximum Longitude is ',jmaxscan
  write(*,*) 'Enter new Maximum Longitude '
  read(*,*) maxscan
c   if(maxscan.gt.maxscann) maxscan=maxscann
  write(*,*) 'Enter an offset for the starting Label for
Nodes[1000] '
  read(*,*) mkn
  print 2
      write(*,*) 'Enter a Scale Factor for Contour Map
(0=FLAT)'
  read(*,*) zscale
  k=mkn

      nnskip=nskip
      mmskip=mskip
      j=minscan
      i=minlat
do 20 while (j.lt.maxscan)
do 19 while (i.lt.maxlat)
  k=k+1
  x(k)=xx(j,i)
  y(k)=yy(j,i)
  z(k)=zz(j,i)
  r(k)=rr(j,i)
  ncolr(k)=mcolr(j,i)
if(j.eq.minscan)ny=k-mkn
numpts=k-mkn
if(ncolr(k).ne.0) then
  xj=real(j)
  yi=real(i)
  zi=zscale*(r(k)-minrad)/(MAXRAD-MINRAD)
  if((mout.eq.1).or.(mout.eq.3))then
    write(8,178)k,0,0,7,x(k),y(k),z(k)

```

```

        write(9,178)k,0,0,ncolr(k),xj,yi,zi
    endif
c   PATRAN
    if((mout.eq.2).or.(mout.eq.3))then
        write(10,500)'1',k,' ','2'
        write(10,501)x(k),y(k),z(k)
        write(10,502)'1G','6'
        write(12,500)'1',k,' ','2'
        write(12,501)xj,yi,zi
        write(12,502)'1G','6'

        endif
    endif

    i=i+nskip
    anincr=float(i-minlat)/float(maxlat-minlat)
    if(anincr.le.p4)then
        nskip=nnskip+ifix(float(nskip2-nnskip)*anincr/p4)
    endif
    if(anincr.gt.p4)then
nskip=nskip2+ifix(float(nskip3-nskip2)*(anincr-p4)/(1.-p4))
    endif

19 end do
    j=j+mskip
    i=minlat
    nskip=nnskip
    amincr=float(j-minscan)/float(maxscan-minscan)

if(amincr.le.q4)mskip=mmskip+ifix(float(mskip2-mmskip)*aminc
r/q4)
    if(amincr.gt.q4)then
mskip=mskip2+ifix(float(mskip3-mskip2)*(amincr-q4)/(1.-q4))
    endif
20 end do

    numcol=numpts/ny
    write(*,*)'total number of points is ',numpts

    write(*,*)numcol,' across and ',ny,' down'
        write(8,47)'    -1'
        write(9,47)'    -1'
    close(8)
        close(9)
    close(10)
    close(12)

    write(*,*)'Press the 1 to continue, the 2 to quit'
    read(*,*) nquitk
    if(nquitk.eq.2) stop

```

```

maxcolor=15
write(*,*)' Enter a Material Property Number '
read(*,*) nmat
write(*,*)' Enter a Minimum color Number (1 thru 15) '
read(*,*) mincolor
write(*,*)' Enter a Maximum color Number (1 thru 15) '
write(*,*)'(choose the same as minimum color for
single'
write(*,*)'color for all elements)'
read(*,*) maxcolor
write(*,*)' '
write(*,*)'          Color Patterns          '
write(*,*)' '
write(*,*)'          1 - Vertical Stripe Coloring      '
write(*,*)'          2 - Horizontal Stripes            '
write(*,*)'          3 - Radial Distance Color Contours
'
write(*,*)'          '
write(*,*)' Enter a 1, 2 or 3 (choice 3 overrides color
choices'
write(*,*)' made previously) [2]'
read(*,*) mvh
if((mvh.ne.1).and.(mvh.ne.2).and.(mvh.ne.3)) mvh=2
write(*,*) 'Enter a new offset for Starting Labels for
Elements[0]'
read(*,*) mke
k=mke
c elements
write(6,47)'          -1'
write(6,47)'          780'
nmcolor=1

do 40 j=1,numcol
do 31 i=1,ny-1

if(nmcolor.gt.maxcolor) then
nmcolor=mincolor
endif

k=k+1
bp(1) = (j-1)*ny+i+mkn
bp(2) = bp(1)+1
bp(3) = bp(2)+ny
bp(4) =bp(1)+ny
c write out quad element if all points are good
if((ncolr(bp(1)).ne.0).and.(ncolr(bp(2))).ne.0).and.
& (ncolr(bp(3)).ne.0).and.(ncolr(bp(4))).ne.0)) then
c change color at each new column

if(mvh.eq.3)nmcolor=(ncolr(bp(1))+ncolr(bp(2))+ncolr(bp(3))+
& ncolr(bp(4)))/4
if((mout.eq.1).or.(mout.eq.3))then
write(6,209)k,94,1,1,1,nmat,nmcolor,4

```

```

write(6,210) IFIX(bp(1)), IFIX(bp(2)), IFIX(bp(3)), IFIX(bp(4))
endif
c patran
    if((mout.eq.2).or.(mout.eq.3))then
write(11,500)'2',k,'4','2'
write(11,503)'4',' ',nmat
write(11,504)bp(1),bp(2),bp(3),bp(4)
endif
goto 30
endif

c    if point 1 is bad, other three are good make a triangle
    if((ncolr(bp(1)).eq.0).and.(ncolr(bp(2)).ne.0).and.
& (ncolr(bp(3)).ne.0).and.(ncolr(bp(4)).ne.0)) then

if(mvh.eq.3)nmcolor=(ncolr(bp(2))+ncolr(bp(3))+ncolr(bp(4)))
/3
    if((mout.eq.1).or.(mout.eq.3))then
write(6,209)k,91,1,1,1,nmat,nmcolor,3
write(6,211) IFIX(bp(2)), IFIX(bp(3)), IFIX(bp(4))
endif
    if((mout.eq.2).or.(mout.eq.3))then
write(11,500)'2',k,'3','2'
write(11,503)'3',' ',nmat
write(11,505)bp(2),bp(3),bp(4)
endif
goto 30
endif

c    if point 2 is bad, other three are good make a triangle
    if((ncolr(bp(1)).ne.0).and.(ncolr(bp(2)).eq.0).and.
& (ncolr(bp(3)).ne.0).and.(ncolr(bp(4)).ne.0)) then

if(mvh.eq.3)nmcolor=(ncolr(bp(1))+ncolr(bp(3))+ncolr(bp(4)))
/3
    if((mout.eq.1).or.(mout.eq.3))then
write(6,209)k,91,1,1,1,nmat,nmcolor,3
write(6,211) IFIX(bp(1)), IFIX(bp(3)), IFIX(bp(4))
endif
    if((mout.eq.2).or.(mout.eq.3))then
write(11,500)'2',k,'3','2'
write(11,503)'3',' ',nmat
write(11,505)bp(1),bp(3),bp(4)
endif
goto 30
endif

c    if point 3 is bad, other three are good make a triangle
    if((ncolr(bp(1)).ne.0).and.(ncolr(bp(2)).ne.0).and.
& (ncolr(bp(3)).eq.0).and.(ncolr(bp(4)).ne.0)) then

if(mvh.eq.3)nmcolor=(ncolr(bp(1))+ncolr(bp(2))+ncolr(bp(4)))
/3
    if((mout.eq.1).or.(mout.eq.3))then

```

```

write(6,209)k,91,1,1,1,nmat,nmcolor,3
  write(6,211)IFIX(bp(1)),IFIX(bp(2)),IFIX(bp(4))
  endif
  if((mout.eq.2).or.(mout.eq.3))then
write(11,500)'2',k,'3','2'
write(11,503)'3',' ',nmat
write(11,505)bp(1),bp(2),bp(4)
  endif
goto 30
endif

c  if point 4 is bad, other three are good make a triangle
  if((ncolr(bp(1)).ne.0).and.(ncolr(bp(2)).ne.0).and.
& (ncolr(bp(3)).ne.0).and.(ncolr(bp(4)).eq.0)) then

if(mvh.eq.3)nmcolor=(ncolr(bp(1))+ncolr(bp(2))+ncolr(bp(3)))
/3
  if((mout.eq.1).or.(mout.eq.3))then
write(6,209)k,91,1,1,1,nmat,nmcolor,3
  write(6,211)IFIX(bp(1)),IFIX(bp(2)),IFIX(bp(3))
  endif
  if((mout.eq.2).or.(mout.eq.3))then
write(11,500)'2',k,'3','2'
write(11,503)'3',' ',nmat
write(11,505)bp(1),bp(2),bp(3)
  endif
goto 30
endif

30  if(mvh.eq.2) nmcolor=nmcolor+1
31  continue
  if(mvh.eq.1)      nmcolor=nmcolor+1
    if(mvh.eq.2)  nmcolor=mincolor
40  continue
  if((mout.eq.1).or.(mout.eq.3)) write(6,47)' -1'
  close(6)
  print 12
  if((mout.eq.1).or.(mout.eq.3))then
    write(*,*) 'universal file of Nodes is :
',fNodesi
    write(*,*) 'universal file of Elements is :
',felemsi
    write(*,*) 'universal file of Nodes mapped to 2d is:
FLAT.unv'
  endif
  print 2
  if((mout.eq.2).or.(mout.eq.3))then
    write(*,*) 'Neutral file of Nodes is :

```

```

',fNodesp
    write(*,*) 'Neutral file of Elements is :
',felemsp
    write(*,*) 'Neutral file of Nodes mapped to 2d is:
FLAT.nut'
endif

```

```

C  FORMATS -----
2    FORMAT(' ')
12   format('1',a60)
33   format('$',1x,a50)
44   format(2x,a25,i3)
47   format(a6)
53   format('+',a22,i6,a9,a8)
178  format(4(i10),1p3e13.5)
187  format(5(i10))
188  format(1p3d25.16)
209  format(8(i10))
210  format(4(i10))
211  format(3(i10))
500  format(a2,i8,a8,a8)
501  format(3e16.9)
502  format(a2,a8)
503  format(2a8,i8)
504  format(4i8)
505  format(3i8)
    RETURN
    END

```

```

C -----
    Subroutine Intro
      character*60 title
      character*9 mdate
      character*8 mtime
      call DATE(mdate)
      do 1 j=1,25
        print 2
1     continue
      print 12
        write(*,*) '
Program'
        print 2
        print 2
        write(*,*) '
Bitterman'
        write(*,*) '
Technology'
        write(*,*) '
        print 2
        write(*,*) '
        do 3 j=1,5
          print 2
          print 2

```

Welcome to the SCAN/CAD

Developed by Capt. B.
Air Force Institute of
Summer 1991'
Today is : ',mdate

```

3  continue
2  FORMAT(' ')
12 format('1',a60)
    RETURN
    end
C-----
SUBROUTINE INSCAN
    IMPLICIT integer (g,t)
    common
/BLK4/minscan,maxscan,minlat,maxlat,minrad,maxrad,numpts
    common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
    common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
    common
/BLK3/mcolr(511,300),jminlat,jmaxlat,jminscan,jmaxscan
    common /BLK5/g(50000),t(50000)
    CHARACTER*8 MTIME
    character*15 fir,fnodesi,felemsi,fnodesp,felemsp
    character*15 fscanlis
    WRITE(*,33) 'ENTER THE NAME OF THE INPUT DATA FILE
[test.dat]: '
    READ(*,'(A15)') FIN
    print 2
    OPEN(5,FILE=FIN,STATUS='OLD')
    PRINT 2
    print 2

c  Need to get some idea of min and max scan latitudes
c  Starting Values
    minlat=300
    minscan=511
    maxscan=0
    maxlat=0
    minrad=300
    maxrad=0

c  initialize xx,yy,zz,mcolr arrays to zero

    do 5 j=1,511
do 4 i=1,300
    xx(j,i)=0.0
    yy(j,i)=0.0
    zz(j,i)=0.0
    rr(j,i)=0.0
    mcolr(j,i)=0
4  continue
5  continue
    jcount=0
do 55 j=1,50000
    read(5,*)g(j),t(j),r(j),x(j),y(j),z(j)
    if(g(j).gt.99900) goto 56
    if(g(j).lt.minscan) minscan=g(j)

```

```

        if(t(j).gt.maxlat) maxlat=t(j)
        if(t(j).lt.minlat) minlat=t(j)
        if(r(j).gt.maxrad) maxrad=r(j)
        if(t(j).lt.minrad) minrad=r(j)
            numpts = j
        jcount=jcount+1
        if(jcount.eq.100)then
            call time(mtime)
            WRITE(*,53)'SCANNED POINTS READ :',J,'
: ',mtime
            jcount=0
            endif
            maxscan=g(j)
55 continue
56 continue
        close(5)
            jminlat=minlat
            jmaxlat=maxlat
            jminscan=minscan
            jmaxscan=maxscan

2      FORMAT(' ')
12 format('1',a60)
33 format('$',1x,a50)
53 format('+',a22,i6,a9,a8)
        RETURN
    end

```

```

C-----
      SUBROUTINE RECTANG
      IMPLICIT INTEGER (g,t)
      common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
      common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
      common
/BLK3/mcolr(511,300),JMINLAT,JMAXLAT,JMINSCAN,JMAXSCAN
      common
/BLK4/minscan,maxscan,minlat,maxlat,minrad,maxrad,numpts
      common /BLK5/g(50000),t(50000)
      do 155 j=1,numpts
          xx(g(j),t(j))=x(j)
          yy(g(j),t(j))=y(j)
          zz(g(j),t(j))=z(j)
          rr(g(j),t(j))=r(j)
          mcolr(g(j),t(j))=(r(j)-minrad)/(maxrad-minrad)*14+1
          if(g(j).gt.99900) goto 156
          if(g(j).lt.minscan) minscan=g(j)
          if(t(j).gt.maxlat) maxlat=t(j)
          if(t(j).lt.minlat) minlat=t(j)
              numpts = j
              maxscan=g(j)
155 continue
156 continue

```

```

c   save original model scan limits

      RETURN
    end
C-----
      SUBROUTINE SCANAL
      common
/BLK4/minscan,maxscan,minlat,maxlat,minrad,maxrad,numpts
      CHARACTER*15 FSCANLIS
      do 157 j=1,10
      print 2
      print 2
157 continue
      write(*,12)'Analysis of Scanned Data '
      print 2
      print 2
      print 2

      write(*,*)'Total Number of scan points      :
',numpts
c   dots per line
      dpl=maxlat-minlat+1
      write(*,*)'Maximum number of Scan Latitudes  :
',dpl
      mscans=maxscan-minscan+1
      write(*,*)'Maximum number of Scan Longitudes :
',mscans
      totpts=mscans*dpl
      write(*,*)'Points Required for Rectangular Mesh:
',totpts
      print 2
      print 2
C
      write(*,44)'Minimum scan Latitude : ',minlat
      write(*,44)'Maximum scan Latitude : ',maxlat
      print 2
      write(*,44)'Minimum scan Longitude : ',minscan
      write(*,44)'Maximum scan Longitude : ',maxscan
      print 2
      write(*,44)'Maximum scan Radius      : ',maxrad
      write(*,44)'Minimum scan Radius      : ',minrad
      print 2
      print 2

      write(*,*) 'Enter      1 - to write this Data to a File'
      write(*,*) '          2 - to continue '
      write(*,*) '          3 - Exit   '
      print 2
      read(*,*) mprint
      if(mprint.eq.3) stop
      if(mprint.eq.2) RETURN
      if(mprint.eq.1) then
      print 12

```

```

        WRITE(*,33) 'ENTER a NAME for Scan Analysis FILE
[scan.lis]:'
        read(*,'(a15)') fscanlis
        open(4,file=Fscanlis,status='NEW')
        write(4,12)'Analysis of Scanned Data '
        print 2
        print 2
        print 2

        write(4,*)'Total Number of scan points      :
',numpts
C      dots per line
        write(4,*)'Maximum number of Scan Latitudes  :
',dpl
        write(4,*)'Maximum number of Scan Longitudes :
',mscans
        write(4,*)'Points Required for Rectangular Mesh:
',totpts
        print 2
        print 2
C
        write(4,44)'Minimum scan Latitude  : ',minlat
        write(4,44)'Maximum scan Latitude   : ',maxlat
        write(4,44)'Minimum scan Longitude : ',minscan
        write(4,44)'Maximum scan Longitude : ',maxscan
        write(4,44)'Maximum scan Radius    : ',maxrad
        write(4,44)'Minimum scan Radius     : ',minrad
        endif
        close(4)
2      FORMAT(' ')
12     format('1',a60)
33     format('$',1x,a50)
44     format(2x,a25,i3)
        RETURN
        end
C-----

```

Appendix G: FRAME Formatted Output

PTIONS

0, 0, 1

JITS

I, N

I, N, PA

TABLE

INTS

0011, -1.91E-05, -82.452, 8.802
0012, 3.43E-05, -85.572, 8.919
0013, -2.43E-05, -88.702, 9.488
0018, 2.8236, -3.4063, -3.4663
0034, 4.3686, -9.0598, -3.9674
0039, 5.2751, -14.7804, -7.4229
0047, 6.14218, -82.452, 9.1282
0048, 6.15, -85.572, 9.3305
0049, 6.1572, -88.702, 9.8586
0060, 7.1569, -23.062, -10.7303
0087, 9.5076, -88.702, 9.488
0089, 9.5436, -85.572, 8.919
0090, 9.5476, -82.452, 8.802
0100, 10.5897, -26.182, -10.5786
0101, 10.6035, -23.062, -10.7303
0102, 10.6111, -29.312, -10.901
0143, 13.3057, -88.702, 8.2328
0146, 13.3614, -85.572, 7.5628
0147, 13.371, -82.452, 7.5281
0171, 15.1348, -32.432, -11.9292
0172, 15.1757, -29.312, -12.321
0173, 15.2035, -26.182, -12.656
0192, 17.433, -88.702, 5.27257
0193, 17.4836, -82.452, 4.846
0195, 17.524, -85.572, 4.51746
0221, 19.534, -35.562, -11.14
0222, 19.7357, -32.432, -12.7196
0226, 19.8363, -29.312, -13.4778
0244, 22.35, -88.702, -.91997
0245, 22.4358, -82.452, -1.4
0246, 22.6413, -85.572, -2.6444
0257, 23.6556, -38.692, -8.778
0269, 24.4277, -35.562, -13.424
0299, 28.5177, -88.702, -10.6417
0302, 28.8428, -85.572, -12.23
0305, 29.0958, -82.452, -13.4898
0307, 29.1354, -79.3779, -13.684
0308, 29.17, -76.684, -13.854
0309, 29.212, -73.393, -14.062
0310, 29.2527, -70.25, -14.2605
0311, 29.293, -67.109, -14.459
0312, 29.3296, -64.266, -14.6385
0313, 29.3719, -60.975, -14.8464
0314, 29.42, -57.234, -15.0827
0315, 29.4488, -54.99, -15.2245
0316, 29.491, -51.6979, -15.4324
0317, 29.539, -47.9575, -15.6686
0320, 29.618, -41.812, -16.0568
0322, 29.6774, -38.692, -16.36
0353, 34.0946, -88.702, -14.6092
0358, 34.4765, -85.572, -16.1844
0361, 34.734, -82.452, -17.2355
0364, 34.8226, -41.812, -17.611

Appendix H: DISPLZ Program

Appendix H: DISPLZ Program

The DISPLZ program was written to automate the procedures for creating an enforced displacement definition for one component to conform exactly to another one. Specifically, this was used to write universal file format boundary conditions to displace the mask seal region to conform with the face. Without this program, one would have to manually decide which nodes on the mask seal were in line with which nodes on the face. Then for every node on the seal, one would have to measure the distance between the nodes, then define a nodal displacement restraint. Doing this for more than about 10 points would get extremely tiresome. The Displz program was written to do this outside of I-DEAS. From within I-DEAS, you must specify which group of nodes to output as a FRAME formatted file (see Appendix G for sample). First, you must define a group that you want the displacements defined on (i.e. the seal region). Since the model was defined with different colors for different areas of the mask, this was easy enough. Then the elements were deleted leaving only the nodes. This set of nodes was output from I-DEAS in FRAME format (the simplest format, basically ID,x,y,z). The same thing was done for the face. DISPLZ reads in the two sets of nodes. For the first set (i.e. the seal) it looks at the second set of nodes and determines the corresponding node (least distance in the xy plane). It then writes out the appropriate displacement in

I-DEAS universal format. There were 74 nodes in the seal set and 851 nodes in the face set. Seventy Four displacement records were written. The analysis was performed and the displacements of the seal were as prescribed. A program listing is found following the figures below. Note also, this same approach was taken to automate definition of a large number of GAP elements (See Appendix I).

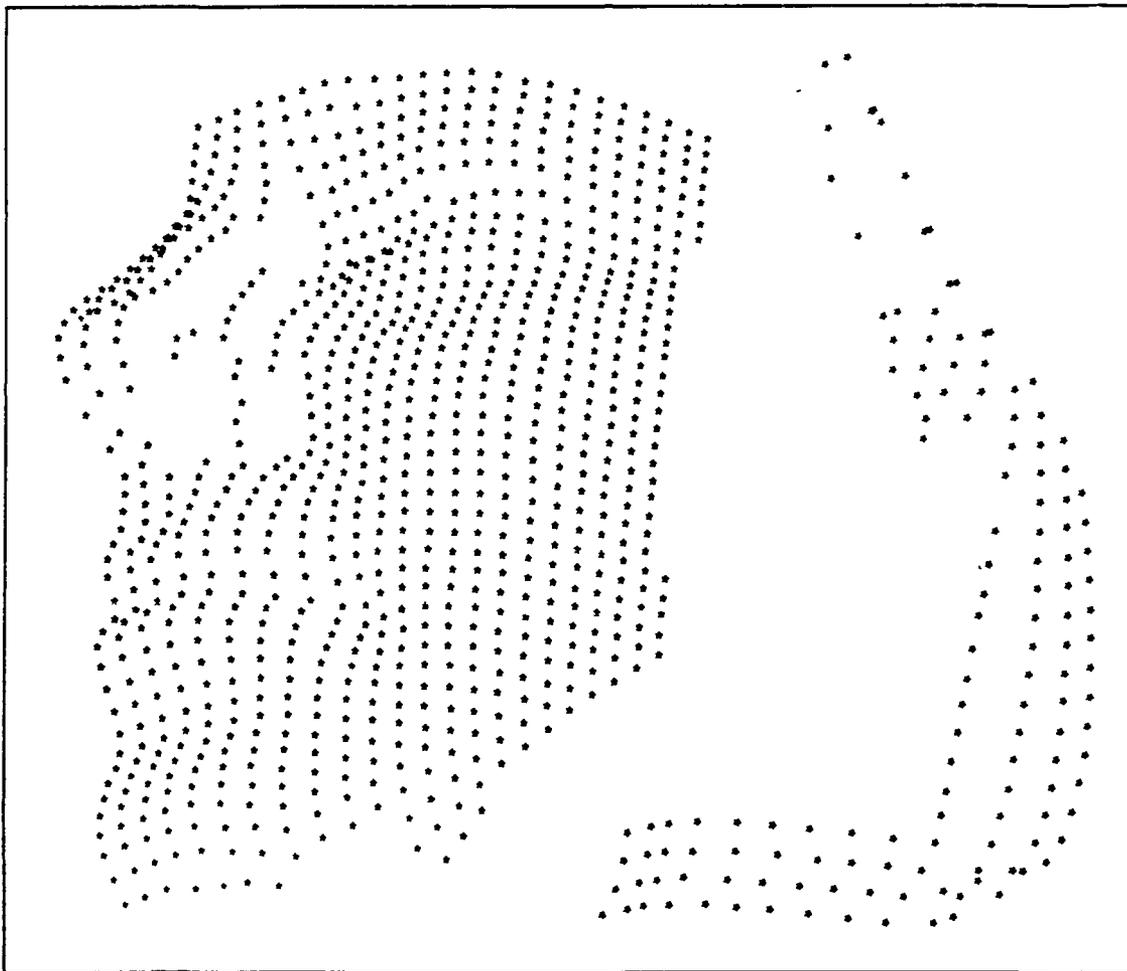


Figure 25. Nodes on Mask Seal and on Face

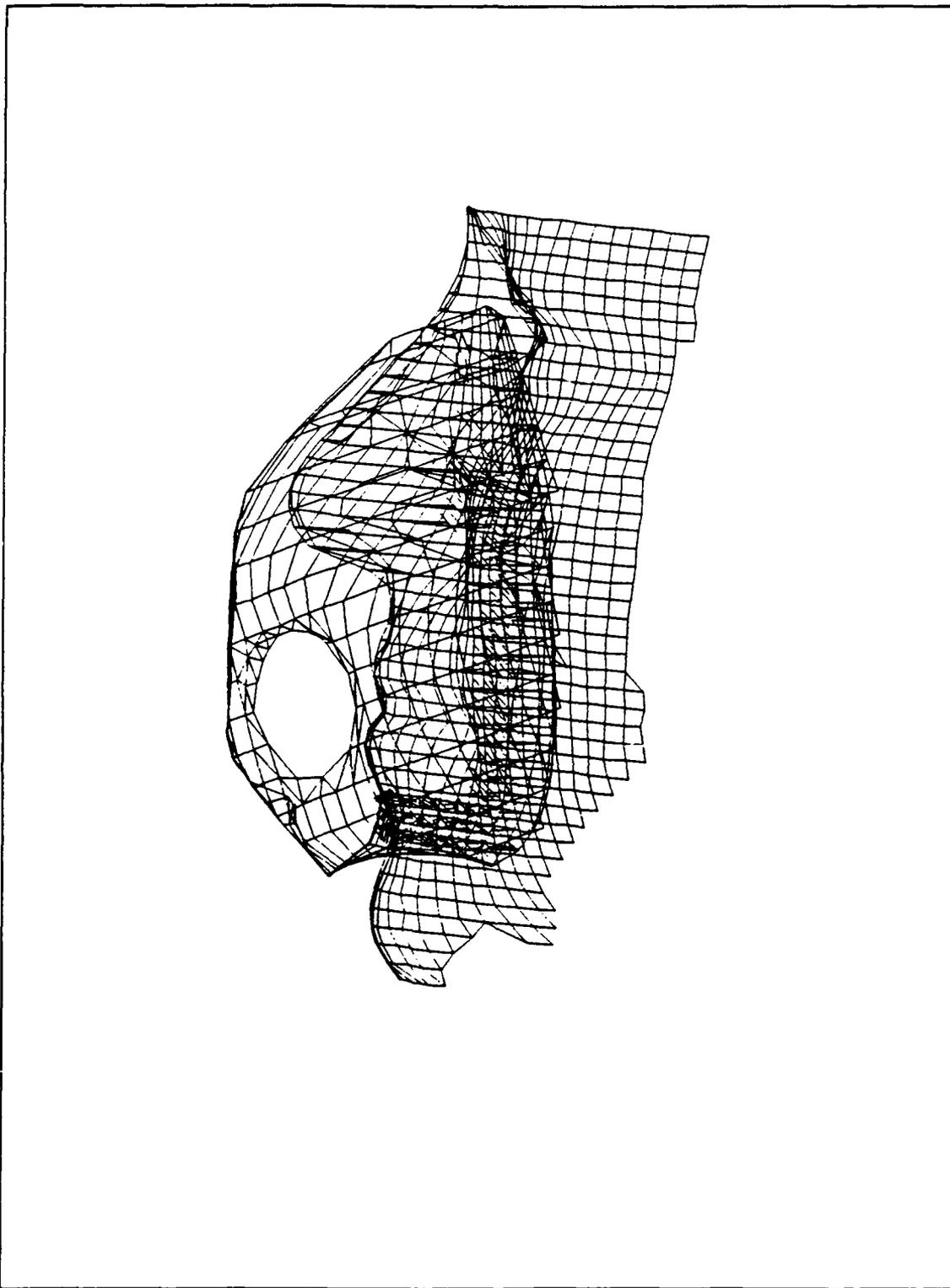


Figure 26. Conforming Displacement of Seal

```

    program displz
c     displaces nodes in the z direction toward other body
of nodes
    dimension ia(10000), x(10000), y(10000),
z(10000), dz(10000)
    dimension jb(20000), xx(20000), yy(20000), zz(20000)
    character*15 fina
    character*15 finb

    dxy=10000.0
    dzmax=0.0
    dzmin=10000.

c
    write(*,*) 'Enter the name of the file containing the
nodes'
    write(*,*) 'that you want displaced from (enclose in
quotes)'
    read(*,*) fina
    write(*,*) 'How many nodes are in that file?'
    read(*,*) napts
    write(*,*) 'Enter the name of the file containing the
nodes'
    write(*,*) 'that you want displaced to (enclose in
quotes)'
    read(*,*) finb
    write(*,*) 'How many nodes are in that file?'
    read(*,*) nbpts
    OPEN(4,FILE=FINA, STATUS='OLD')
    OPEN(5,FILE=FINB, STATUS='OLD')
    DO 10 I=1, NAPTS
10    READ(4,*) ia(i), x(i), y(i), z(i)
        continue
        do 20 j=1, nbpts
20    read(5,*) jb(j), xx(j), yy(j), zz(j)
            close(4)
            close(5)

c
c     data has been read in
c     Now, calculate corresponding point on body b and z
distance
c
    do 30 i=1, napts
    do 25 j=1, nbpts
    dist2=(xx(j)-x(i))**2+(yy(j)-y(i))**2
    if(dist2.lt.dxy) then
    dz(i)=zz(j)-z(i)
    dxy=dist2
    endif
25    continue
    dxy=10000.
    if(dz(i)*dz(i).gt.dzmax*dzmax) dzmax=-dz(i)
    if(dz(i)*dz(i).lt.dzmin*dzmin) dzmin=dz(i)
30    continue

```

```

c   write header to output universal file of restraints
    open(7,FILE='DISPLZ.UNV', STATUS='NEW')
    WRITE(7,1)'    -1'
    WRITE(7,1)'    755'
    lmn=1
    mno=0
    d0=0.0
    write(7,2)lmn,lmn
    write(7,3)'1-', 're', 'st', 'ra', 'in', 't ', 'se', 't ', '1 '
    write(*,*)'Do you want to subtract off minimum
distance? '
    write(*,*)'This will leave one point fixed '
    write(*,*)'enter 1 for Yes, 2 for No, or 3 to subtract
max '
    read(*,*) mfix
    if(mfix.eq.2) dzminn=0.0
    if(mfix.eq.1) dzminn=dzmin
    if(mfix.eq.3) dzminn=-dzmax
    DO 40 I=1,NAPTS
    write(7,4)ia(i),lmn,lmn,lmn,lmn,mno,mno,mno,mno
c
c   write(7,5)d0,d0,dz(i)-dzmax,d0,d0,d0
    write(7,5)d0,d0,dz(i)-dzminn,d0,d0,d0
40  continue
    write(7,1)'    -1'
    write(*,*) 'Universal file of Enforced displacement in'
    write(*,*) 'Z direction is DISPLZ.UNV'
    close(7)

1   FORMAT(A6)
2   format(2I10)
3   format(9a2)
4   format(2(I10),7(I2))
5   format(1p6e13.5)
    end

```

Appendix I: GAPZ Program

Appendix I: GAPZ Program

The GAPZ program was written to automate GAP element definition which is very labor intensive in I-DEAS. Each GAP element must have defined the connecting nodes, the inter-node distance, the GAP direction, its own physical and material property record. GAPZ creates GAP elements from one body to another in the negative z direction. The programming logic is similar to the DISPLZ program. The figure below shows the face and mask connected by 74 gap elements which were produced using the GAPZ program. Program listing follows the figure.

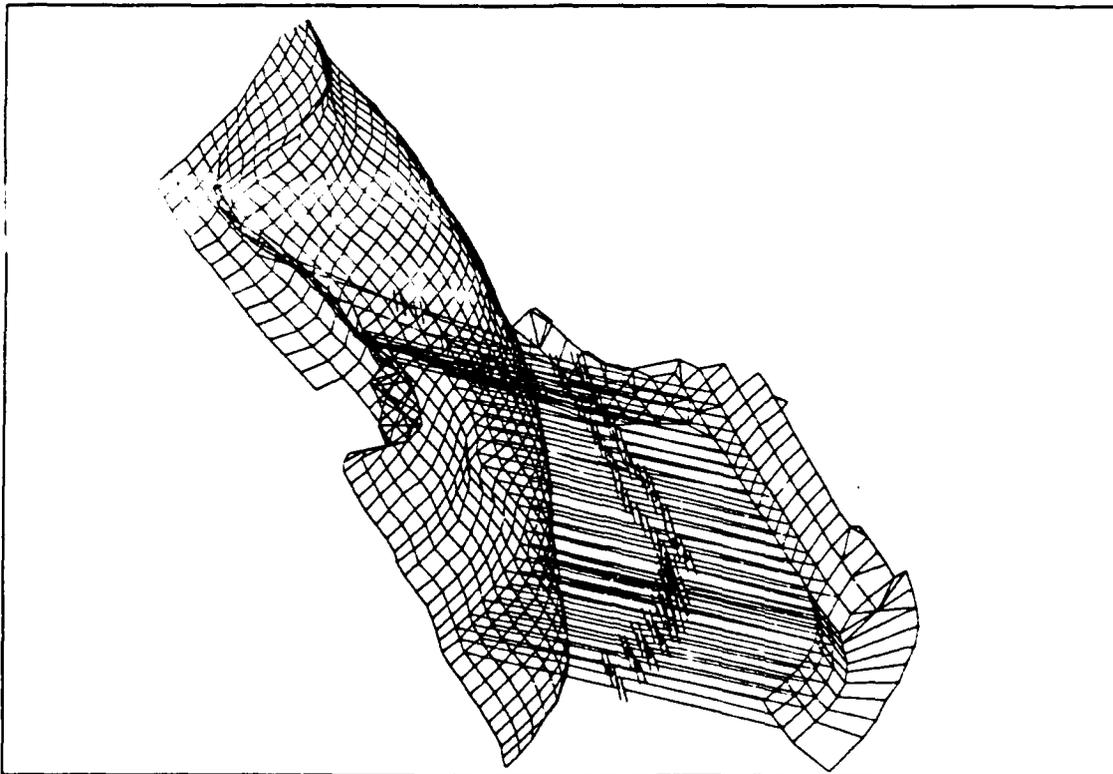


Figure 28. GAP Elements Created by GAPZ

```

    program gapz
c      creates gaps in the z direction toward other body of
nodes
    dimension ia(10000), x(10000), y(10000),
z(10000),dz(10000)
    dimension jb(20000), xx(20000), yy(20000),
zz(20000),kc(10000)
    character*15 fina
    character*15 finb

    dxy=10000.0
    dzmax=0.0
    dzmin=10000.0

c
    write(*,*) 'Enter the name of the file containing the
nodes'
    write(*,*) 'that you want gaps from (enclose in
quotes)'
    read(*,*) fina
    write(*,*) 'How many nodes are in that file?'
    read(*,*) napts
    write(*,*) 'Enter the name of the file containing the
nodes'
    write(*,*) 'that you want gaps to (enclose in quotes)'
    read(*,*) finb
    write(*,*) 'How many nodes are in that file?'
    read(*,*) nbpts
    OPEN(4,FILE=FINA, STATUS='OLD')
    OPEN(5,FILE=FINB, STATUS='OLD')
    DO 10 I=1,NAPTS
10    READ(4,*) ia(i), x(i),y(i),z(i)
    continue
    do 20 j=1, nbpts
20    read(5,*) jb(j), xx(j),yy(j), zz(j)
    enddo
    close(4)
    close(5)

c
c      data has been read in
c      Now, calculate corresponding point on body b and z
distance
c
    do 30 i=1,napts
    do 25 j=1, nbpts
    dist2=(xx(j)-x(i))**2+(yy(j)-y(i))**2
    if(dist2.lt.dxy) then
    dz(i)=zz(j)-z(i)
    kc(i)=jb(j)
    dxy=dist2
    endif
25    continue
    dxy=10000.
    if(dz(i)*dz(i).gt.dzmax*dzmax) dzmax=dz(i)
    if(dz(i)*dz(i).lt.dzmin*dzmin) dzmin=dz(i)

```

```

30 continue
write(*,*)'max distance = ',dzmax
write(*,*)'min distance = ',dzmin
c write header to output universal file of restraints
open(7,FILE='GAPZ.UNV', STATUS='NEW')
WRITE(7,1)' -1'
WRITE(7,1)' 780'
condir=3.0
write(*,*)'Enter Contact direction -3 for press fit in
Z, 3 for '
write(*,*)'gap in positive z direction'
read(*,*) condir
if(abs(condir).gt.6) condir=3.0
lmn=1
mno=0
nop=151
d0=0.0
DO 40 I=1,NAPTS
jprop=100+i
jcol=7
jlabel=i+500000
numnodes=2
jmat=3
write(7,6)jlabel,nop,lmn,jprop,lmn,jmat,jcol,numnodes
write(7,2)ia(i),kc(i)
c
40 continue
jvals=17
write(7,1)' -1'
write(7,1)' -1'
write(7,1)' 772'
write(*,*)'do you want distance subtracted off? 1=yes
2=no'
write(*,*)'selecting yes will cause contact at one
point'
read(*,*)msubtr
if(condir.lt.0.0) dzmin=dzmax
if(msubtr.eq.2) dzmin=0.0

do 100 i=1,napts
jprop=i+100
z0=0.0
write(7,7)jprop,nop,jvals
write(7,8)'No','de','-t','o ','no','de',' G','ap',jprop
c min distance is subtracted so contact is made at at least
one point
write(7,9) abs(dz(i)-dzmin),condir,0.0,0.0,0.0,0.0
write(7,9)z0,z0,z0,z0,z0,z0
write(7,9)z0,z0,z0,z0,z0,z0
write(7,9)z0,z0,z0,z0,z0,z0
write(7,11)z0,z0,z0,z0

100 continue

```

```
write(*,*) 'Universal file of Gap Elements in'  
write(*,*) 'Z direction is GAPZ.UNV'  
close(7)  
  
1  FORMAT(A6)  
2  format(2I10)  
3  format(9a2)  
4  format(2(I10),7(I2))  
5  format(lp6e13.5)  
6      format(8i10)  
7  format(3i10)  
8  format(8(a2),i4)  
9  format(lp6e13.6)  
11 format(lp4e13.6)  
end
```

APPENDIX J: RELABL Program

APPENDIX J: RELABL Program

When defining model reduction ramp functions in SCANCAD, it may be desirable to start with other than the first longitude position. This can be done by only selecting a segment of the model. If you want to define the ramp function over the whole parametric space, and there are known planes of symmetry it may have been more advantageous if the first longitude lined up with the plane of symmetry. The RELABL program lets the user define a new longitude as the first longitude. The parametric map's left edge now starts at this longitude. This is illustrated below.

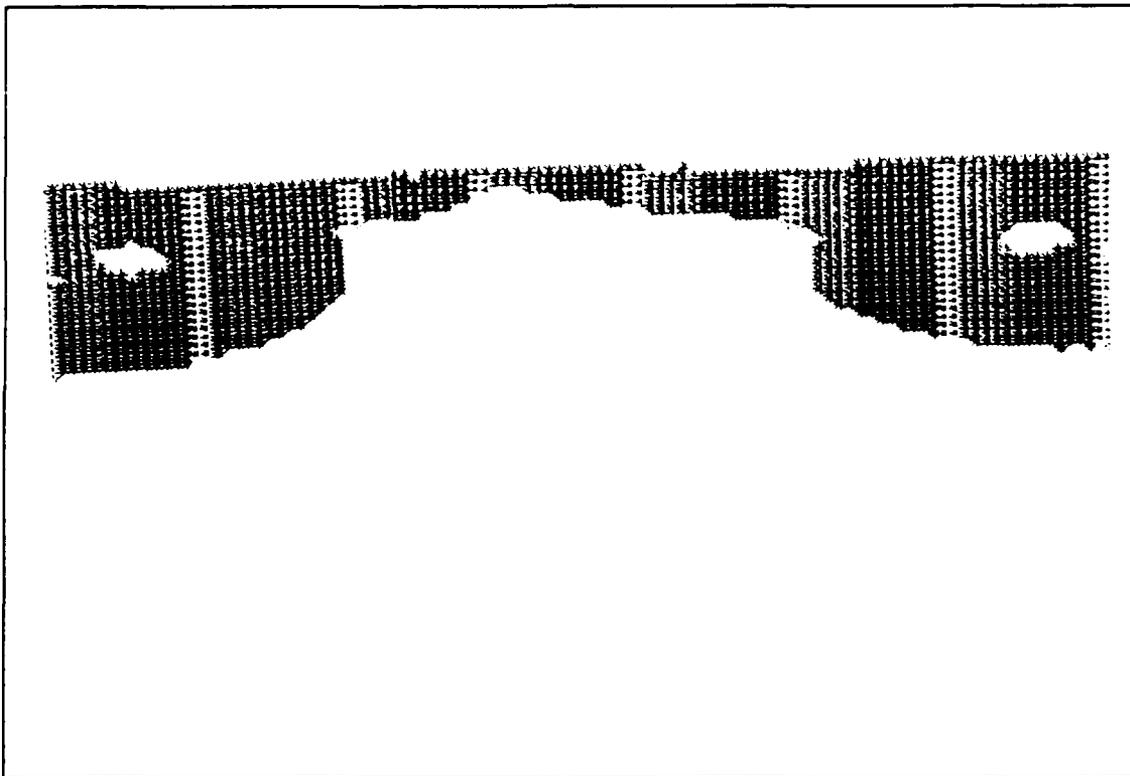


Figure 29. Change of Starting Longitude

Compare this to Figure 7.

```

PROGRAM RELABEL
implicit integer (g,t)
common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
common
/BLK3/mcolr(511,300),JMINLAT,JMAXLAT,JMINSCAN,JMAXSCAN
common felemsi,fnodesi,felemsp,fnodesp
COMMON
/BLK4/MINSCAN,MAXSCAN,MINLAT,MAXLAT,MINRAD,MAXRAD,NUMPTS
common /BLK5/g(50000),t(50000)

```

C reads in scanned data in ascii form

c Print Title Screen/Menu

CALL INTRO

c Read in the Scanned Data

CALL INSCAN

END

```

-----
C-----
Subroutine Intro
character*60 title
character*9 mdate
character*8 mtime
call DATE(mdate)
do 1 j=1,25
  print 2
1 continue
  print 12
  write(*,*) '
Program'
  print 2
  print 2
  write(*,*) '
Bitterman'
  write(*,*) '
Technology'
  write(*,*) '
  print 2
  write(*,*) '
  do 3 j=1,5
  print 2
  print 2
3 continue
2 FORMAT(' ')
12 format('1',a60)
RETURN
Welcome to the Relabel
Developed by Capt. B.
Air Force Institute of
Summer 1991'
Today is : ',mdate

```

```

end
C-----
SUBROUTINE INSCAN
  IMPLICIT integer (g,t)
  common
/BLK4/minscan,maxscan,minlat,maxlat,minrad,maxrad,numpts
  common
/BLK1/r(50000),x(50000),y(50000),z(50000),ncolr(50000)
  common
/BLK2/xx(511,300),yy(511,300),zz(511,300),rr(511,300)
  common
/BLK3/mcolr(511,300),jminlat,jmaxlat,jminscan,jmaxscan
  common /BLK5/g(50000),t(50000)
  CHARACTER*8 MTIME
  character*15 fin,fnodesi,felemsi,fnodesp,felemsp
  character*15 fscanlis,fout
  WRITE(*,33) 'ENTER THE NAME OF THE INPUT DATA FILE
[test.dat]: '
  READ(*,'(A15)') FIN
  print 2
  OPEN(5,FILE=FIN,STATUS='OLD')
  WRITE(*,33) 'ENTER THE NAME OF THE output FILE
[test.out]: '
  READ(*,'(A15)') Fout
  print 2
  OPEN(6,FILE=Fout,STATUS='new')
  PRINT 2
  print 2

  do 55 j=1,50000
    read(5,*)g(j),t(j),r(j),x(j),y(j),z(j)
    if(g(j).gt.99900) goto 56
C *Double the g and t for mesh refinement

    if(g(j).lt.minscan) minscan=g(j)
    if(t(j).gt.maxlat) maxlat=t(j)
    if(t(j).lt.minlat) minlat=t(j)
    if(r(j).gt.maxrad) maxrad=r(j)
    if(t(j).lt.minrad) minrad=r(j)
    numpts = j
    jcount=jcount+1
    if(jcount.eq.100)then
      call time(mtime)
      WRITE(*,53)'SCANNED POINTS READ :',J,'
: ',mtime
      jcount=0
      endif
    maxscan=g(j)
55 continue
56 continue
  close(5)
  jminlat=minlat
  jmaxlat=maxlat
  jminscan=minscan

```

Time

```

jmaxscan=maxscan
jstart=1
write(*,*) 'which longitude should become number 1 ?'
read(*,*) number1
do 100 j=1,numpts
g(j)=g(j)+1-number1
if(g(j).le.0) then
g(j)=g(j)+maxscan
jstart=jstart+1
endif
100 continue

do 200 j=jstart,numpts
write(6,201)g(j),t(j),r(j),x(j),y(j),z(j)
200 continue
do 300 j=1,jstart-1
write(6,201)g(j),t(j),r(j),x(j),y(j),z(j)
300 continue
write(6,*)'999999,1 1 1 1 1'
close (6)

2      FORMAT(' ')
12  format('l',a60)
33  format('$',lx,a50)
53  format('+',a22,i6,a9,a8)
201 format(2i5,4f8.2)
      RETURN
end

```

C-----

Appendix K: SCANCAD Variables Dictionary

Appendix K: SCANCAD Variables Dictionary

fin - Name of the input file of scanned coordinates

fnodesi - name of the ideas universal file of nodes to
be output

fnelemsi - name of the Ideas universal file of elements to
be output

fnodesp - name of the Patran neutral file of nodes to be
output

felemsp - name of the Patran neutral file of elements to be
output

fscanlis - name of the file output containing an
analysis of the scanned data

minlat - minimum latitude value in the scanned data

minscan - minimum longitude value in the scanned data

minrad - minimum radial distance in the scanned data

maxlat - maximum latitude value in the scanned data

maxrad - maximum radial distance in the scanned data

maxscan - maximum longitude value in the scanned data

g(j) - longitude value of the jth digitized (scanned)
point

t(j) - latitude value of the jth digitized (scanned)
point

r(j) - radial distance of the jth digitized scan point

x(j) - x coordinate of the jth digitized scan point

y(j) - y coordinate of the jth digitized scan point

z(j) - z coordinate of the jth digitized scan point

xx(j,i) - x coordinate of the point located at the intersection of the jth longitude and the ith latitude.

yy(j,i) - y coordinate of the point located at the intersection of the jth longitude and the ith latitude.

zz(j,i) - z coordinate of the point located at the intersection of the jth longitude and the ith latitude.

rr(j,i) - r (radial) coordinate of the point located at the intersection of the jth longitude and the ith latitude.

mcolr(j,i) - color value associated with the point located at the intersection of the jth longitude and the ith latitude

jcount - counter to determine when to update the screen display

numpts - total number of scanned points read in

dpl - maximum number of latitudes containing scanned data within a single longitude

mscans - maximum number of longitudes containing scanned data

totpts - (Minimum) Total number of points required for a rectangular mesh containing all scanned points (mscans x dpl).

nskip - starting value for keeping scanned points only in every nth latitude position.

nskip2 - intermediate value for keeping only every nth
latitude

nskip3 - final value for keeping only every nth latitude
position

p4 - real value between 0 and 1 determines placement of
nskip2

mskip - starting value for keeping scanned points only in
every nth longitude position.

mskip2 - intermediate value for keeping only every nth
longitude

mskip3 - final value for keeping only every nth
longitude position

q4 - real value between 0 and 1 determines placement of
mskip2

jminlat - minimum latitude value of scanned data before
any reduction

jmaxlat - maximum latitude value of scanned data before
any reduction

jmaxscan - maximum scan longitude before any reduction

jminscan - minimum scan longitude before any reduction

mkn - offset value for starting label of nodes

zscale - scaling factor for z direction perpendicular to 2D
parametric mapping of nodes.

k - index for sequential numbering of nodes

ncolr(k) - color value associated with node label k
(+mkn user supplied offset value)

ny - number of points in the latitude direction in the
reduced model

nnskip - actual number of latitudes to jump to during a
particular iteration

mmskip - actual number of longitudes to jump to during a
particular iteration

numcol - number of longitudes kept in the reduced model

amincr - distance between first and last longitude at a
particular iteration step, proceeding from minimum to
maximum longitude

anincr - distance between first and last latitude at a
particular iteration step, proceeding from minimum
to maximum latitude

nquitk - allows for program termination at intermediate
stages

nmat - material ID to be assigned to elements

mincolor - minimum color to be assigned to elements

maxcolor - maximum color to be assigned to elements

mvh - determines color coding of elements (horizontal,
vertical or based on radial distance)

mke - offset value for element labels

bp(1) thru bp(4) - boundary points for connecting 4
points to form a quad element or 3 points to form a
triangular element.

mout - determines if output format is for Ideas or
Patran or both

nmcolor - color determined for an individual element

Appendix L: List of Equipment

List of Equipment

(Purpose)

Hardware

AFIT Computers

- Vax 8500, Tektronix Terminals Host/Display CAD Software, Compile FORTRAN programs
- SUN Sparc Station Host/Display CAD Software

AAMRL Computers

- Silicon Graphics Iris Display/Translate digitized data
 - Cyberware Color 3D Digitizer Model 4020 RGB /PS - D Digitize 3D Mask 512 x 256 Resolution
- Cyberware Laboratory Inc
Monterey CA 93940
408 373-1441

Software

- FORTRAN Compiler Interface program development
- SDRC I-DEAS FEM Modeling & Analysis Program
SDRC
2000 Eastman Dr
Milford Oh 45150
(513) 576-2789
- PDA PATRAN FEM Modeling & Analysis Program
PDA
1560 Brookhollow Dr
Santa Ana, CA 92705

Miscellaneous

- Micrometer Measure thicknesses of various parts of the oxygen mask

Appendix M. Discussion of GAP Elements

Some problems became evident after the initial analysis using the gap elements. The gap element will not allow additional motion after the gap is closed. This means that no sliding is permitted and unreasonably large strains may result. Also, the gap element does not seem to be suited toward large displacements. The gap elements each have to be individually defined requiring a large manual effort. An analysis with gap elements was done with gap elements at all nodes of the face seal region. This required writing a program to automate gap element creation (GAPZ.FOR).

Appendix N. NEWSHAPE Program

This program reads in the original model nodes (1st file) and the nodal displacements (2nd file) and writes out a new file of nodes in the displaced position for input to I-DEAS. The element connectivities can be kept the same.

```

    program newshape
c      displaces nodes from original configuration to new
configuration
    dimension ia(10000), x(10000), y(10000), z(10000)
    dimension jb(20000), xx(20000), yy(20000),
zz(20000),dum(50000)
    character*15 fina
    character*15 finb

    dxy=10000.0
    dzmax=0.0
    dzmin=10000.

c
    write(*,*) 'Enter the name of the file containing the
nodes'
    write(*,*) 'that you want displaced from (enclose in
quotes)'
    read(*,*) fina
    write(*,*) 'How many nodes are in that file?'
    read(*,*) napts
    write(*,*) 'Enter the name of the file containing the
nodal'
    write(*,*) 'displacements(enclose in quotes)'
    read(*,*) finb
    write(*,*) 'How many nodes are in that file?'
    read(*,*) nbpts
    OPEN(4,FILE=FINA, STATUS='OLD')
    OPEN(5,FILE=FINB, STATUS='OLD')
    DO 10 I=1,NAPTS
    READ(4,*) ia(i), x(i),y(i),z(i)
10  continue
    do 20 j=1, nbpts
    read(5,*) jb(j), xx(j),yy(j),
zz(j),dum(j),dum(j),dum(j)
20  enddo
    close(4)
    close(5)
c      write header to output universal file of nodes

    open(7,FILE='newnode.unv', STATUS='NEW')
    WRITE(7,1)'    -1'
    WRITE(7,1)'    15'

c
c
    do 30 i=1,napts
    do 25 j=1, nbpts
c      z displacement only is added
    if(ia(i).eq.jb(j)) then
    z(i)=z(i)+zz(j)
    goto 1785
    endif
25  continue
1785  continue
    write(7,1784)ia(i),0,0,7,x(i),y(i),z(i)

```

```
30 continue
1784     format(4(i10),1p3e13.5)
      write(7,1) '      -1'
      write(*,*) 'Universal file of new placement of nodes
in'
      write(*,*) 'Z direction is newnode.unv'
      close(7)

1  FORMAT(A6)
2  format(2I10)
3  format(9a2)
4  format(2(I10),7(I2))
5  format(1p6e13.5)
end
```

Appendix O. REAC Program

This program reads in reaction or element forces calculated by a FEM program and writes them out as nodal forces for input to I-DEAS as a universal file.

```

program reac
c   creates nodal forces in Z direction to counter negative
z
c   displacement
c it takes in frame formatted reaction force data and
outputs the
c   reaction forces in z
      dimension ia(10000), x(10000), y(10000), z(10000)
      dimension xx(20000), yy(20000), zz(20000)
      character*15 fina

      write(*,*) 'Enter the name of the file containing the
reaction'
      write(*,*) 'forces at nodes (enclose in quotes)'
      read(*,*) fina
      write(*,*) 'How many nodes are in that file?'
      read(*,*) napts
      write(*,*) 'Enter a value to use as an absolute limit
on '
      write(*,*) 'the forces'
      read(*,*) fmax
      OPEN(4,FILE=FINA, STATUS='OLD')
      DO 10 I=1,NAPTS
      READ(4,*) ia(i),x(i),y(i),z(i),xx(i),yy(i),zz(i)
      if(z(i).gt.fmax) z(i)=fmax
      if(z(i).lt.-fmax) z(i)=-fmax
      if(y(i).gt.fmax) y(i)=fmax
      if(y(i).lt.-fmax) y(i)=-fmax
      if(x(i).gt.fmax) x(i)=fmax
      if(x(i).lt.-fmax) x(i)=-fmax

c   z(i)=-z(i)
10  continue
      close(4)
c   write header to output universal file of nodes

      open(7,FILE='reac.unv', STATUS='NEW')
      WRITE(7,1)'      -1'
      WRITE(7,1)'      782'

c
      write(7,300) 1,1
      write(7,3) 're','ac','ts'

c
      do 30 i=1,napts
      if(z(i).ne.0.0) then
      write(7,310) ia(i),10,0,0,1,0,0,0
      write(7,5) x(i),y(i),z(i),0.,0.,0.
      endif
30  continue
      write(7,1)'      -1'
      write(*,*) 'Universal file of reactionforces'
      write(*,*) ' is reac.unv'
      close(7)

```

```
1  FORMAT(A6)
2  format(2I10)
3  format(9a2)
4  format(2(I10),7(I2))
5  format(1p6e13.5)
31 format(20A2)
300 format(2(i10))
310 format(2(i10),6(i2))
end
```

Bibliography

1. AFFTC-TR-87-05 Qualitative Evaluation of the Tactical Life Support System (TLSS) in the F-15, 1Lt M.R.Harbert,HSD/RDSL, Brooks AFB Texas 78235-5000)
2. SAFE Journal Vol 18, No 1,"High Altitude/Low Profile Positive Pressure Breathing (HA/LP PPB) Oxygen Mask Assembly", PP 18-24, Wilcox & Engle
3. MIL-M-87163A(USAF), Military Specification, Mask, Oxygen, MBU-12/P.
4. 'Engineering Design With Rubber',Payne and Scott, Interscience Publishers, 1960.
5. "On NURBS: A Survey ", IEEE Computer Graphics and Applications, Jan 1991, Pg. 55, Les Piegl
6. Computer Graphics in Medicine: The Past Decade IEEE Computer Graphics and Applications, Jan 1991, Pg 52, M. Rhodes
7. Conversations with TSgt McIver, USAF Medical Center/SGT (June 1991)
8. 'Concepts and Applications of Finite Element Analysis', Cook, 3rd Ed. Wiley, 1989
9. GENTEX Corp Specification 3936401
10. "Facial Surface Scanner", IEEE Computer Graphics and Applications, Nov 1991, Pg 72, Vannier, et al

Vita

PII Redacted

Captain Bruce H. Bitterman [REDACTED]

[REDACTED]. He enlisted in the Air Force in 1977 and worked as an aircraft crew chief. He obtained a B.S. in Aerospace Engineering from the University of Texas, Austin through the AECP program. In 1983 Captain Bitterman was commissioned in the Air Force. His first commissioned assignment was as a structural engineer at the Air Force Weapons Laboratory. He earned an M.A in Computer Resource Management from Webster University at night school. He served as Chief Flight Systems Engineer for ASD/SD and ASD/AE -PRAM/RAMTIP before entering AFIT for the Aeronautical Engineering Master's Program.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE Dec 1991	3. REPORT TYPE AND DATES COVERED Master's Thesis
----------------------------------	----------------------------	---

4. TITLE AND SUBTITLE Application of Finite Element Modeling and Analysis to the Design of Positive Pressure Oxygen Masks	5. FUNDING NUMBERS
--	--------------------

6. AUTHOR(S) Captain Bruce H. Bitterman	
--	--

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT/ENY WPAFB, OH 45433	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAE/ENY/91D-22
---	---

9. SPONSORING, MONITORING AGENCY NAME(S) AND ADDRESS(ES) AAMRL/HEG WPAFB, OH 45433	10. SPONSORING MONITORING AGENCY REPORT NUMBER
--	--

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved For Public Release; Distribution Unlimited	12b. DISTRIBUTION CODE A
---	---------------------------------

13. ABSTRACT (Maximum 200 words) Rapid development of finite element models (FEM) from digitized data is a needed capability for reverse engineering, particularly at Armstrong Aerospace Medical Research Laboratory (AAMRL). The pilot's oxygen mask was scanned using a 3-D digitizer at AAMRL. FORTRAN code was developed to transform the digitized data to FEM format readable by two commercial programs, I-DEAS and PATRAN. Static analyses were done using non-linear gap elements to connect the mask and face. Reaction forces reveal the fit at various loadings.
--

14. SUBJECT TERMS Finite Elements, Oxygen Mask, Digitized Data, Modeling and Analysis	15. NUMBER OF PAGES 127
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL
---	--	---	----------------------------------