Order Number 1347043

Mechanical computer aided engineering of surface unit supports: An analytical design and structural analyses of a new electric range surface unit support comparing its strength to an existing design, applying solid and finite element modeling and analyses, verified by simple static loading experiments

Kern, Jeffrey Alan, M.S.

University of Louisville, 1991

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

MECHANICAL COMPUTER AIDED ENGINEERING OF SURFACE UNIT SUPPORTS

An analytical design and structural analyses of a new electric range
surface unit support comparing its strength to an existing design,
applying solid and finite element modeling and analyses,
verified by simple static loading experiments.

By

Jeffrey A. Kern
B.S.M.E., Purdue University, 1985

A Thesis
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science

Department of Mechanical Engineering
University of Louisville
Louisville, Kentucky

December, 1991

MECHANICAL COMPUTER AIDED ENGINEERING OF SURFACE UNIT SUPPORTS

An analytical design and structural analyses of a new electric range surface unit support comparing its strength to an existing design, applying solid and finite element modeling and analyses, verified by simple static loading experiments.

By

Jeffrey A. Kern
B.S.M.E., Purdue University, 1985

A Thesis Approved on

November 15, 1991
(DATE)

by the following Reading Committee:

Julius P Wong
Thesis Director

Michael A Cassaro

Michael L. Day

ABSTRACT

This thesis determines analytically the degree to which an electric range surface unit's structural integrity would change if its delta shaped support was replaced with a Y shaped design. It develops a new Y shaped support to fit the unit's sheathed heater, and to rest in the same trim ring as the delta support, minimizing the material required. The new support and the delta design were modeled and analyzed using finite element analyses (FEA). The analyses compare the support's responses to static loads. The static loads applied represent a load test which has historically been used to compare relative strength of surface unit supports.

This thesis could have been completed using traditional design engineering and laboratory evaluation methods (i.e., designing parts, building samples, and running the load test). It was performed as an analytic study to show that design engineering productivity can be increased through the use of "Mechanical Computer Aided Engineering" (MCAE). The results of the analyses were in good agreement with the results of the physical load test that was modeled.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the assistance of Dr. J. P. Wong for is assistance and guidance in the completion of this thesis.

The author also wishes to thank Dr Wong and Dr. M. L. Day for his assistance in teaching the author the basics of finite element analysis, and for Dr. Day's assistance in working out the problems encounter in the completion of the analyses performed.

The author wishes to express his thanks to GE Appliances for the use of their equipment, software, and computers, especially for those that were loaned to the author for an extended period of time for use in his home. If it weren't for this, the author might never have had the ability to complete this work. In addition to this the author wishes to thank his managers at GE for their patience in waiting for the completion of this project.

The author is indebted to his wife, and daughters, for without there help and patient understanding of the countless hours spent away from them, this work would not have been possible.

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

(continued)

LIST OF ILLUSTRATIONS

(continued)

Page

CHAPTER I

INTRODUCTION

THESIS

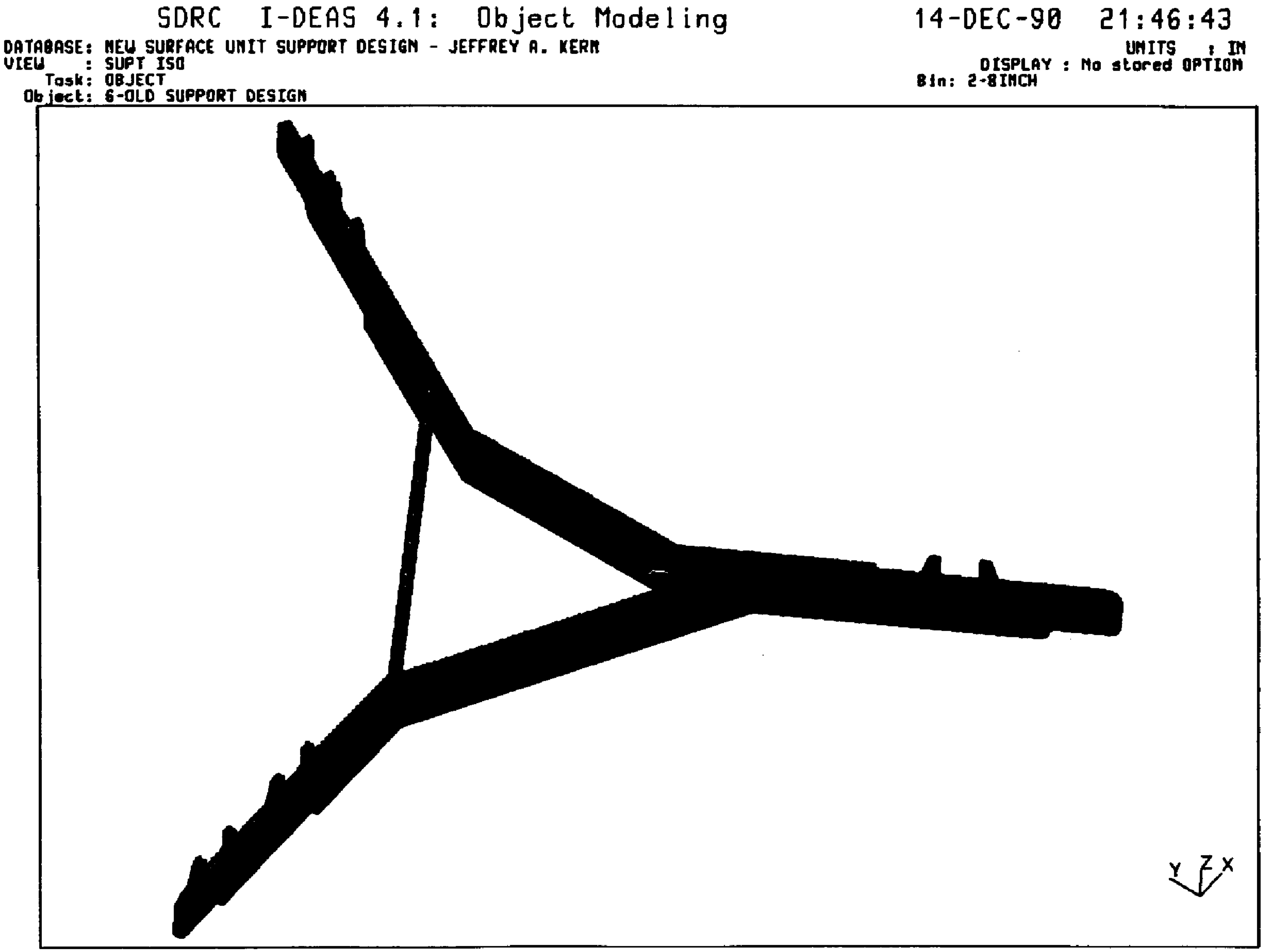
GE Appliances (GEA) uses relatively expensive "delta" shaped supports for the Calrod^{R} [2] surface units ("burners") they produce for cooking on the electric ranges they manufacture. Their competitors use simple "Y" shaped supports for their surface units, which use less material (fairly expensive stainless steel) and fewer pieces to construct. This puts GEA at a cost disadvantage. But, in order to maintain high quality standards, GEA has set strength requirements for their surface units which has caused them to retain the more expensive support designs.

This thesis develops and analyzes a simple "Y" shaped support for one of GEA's Calrod^{R} Surface Units. This design is to replace the "delta" shaped support that GEA currently uses, without any modification of the heating element. The new design is to perform acceptably from a structural standpoint, and to require less material than the current design. Thus, the intent of the new support design is material cost productivity, provided the pay back period on the tooling and implementation costs are acceptable.

[2] Calrod is a registered trademark of the General Electric Company.

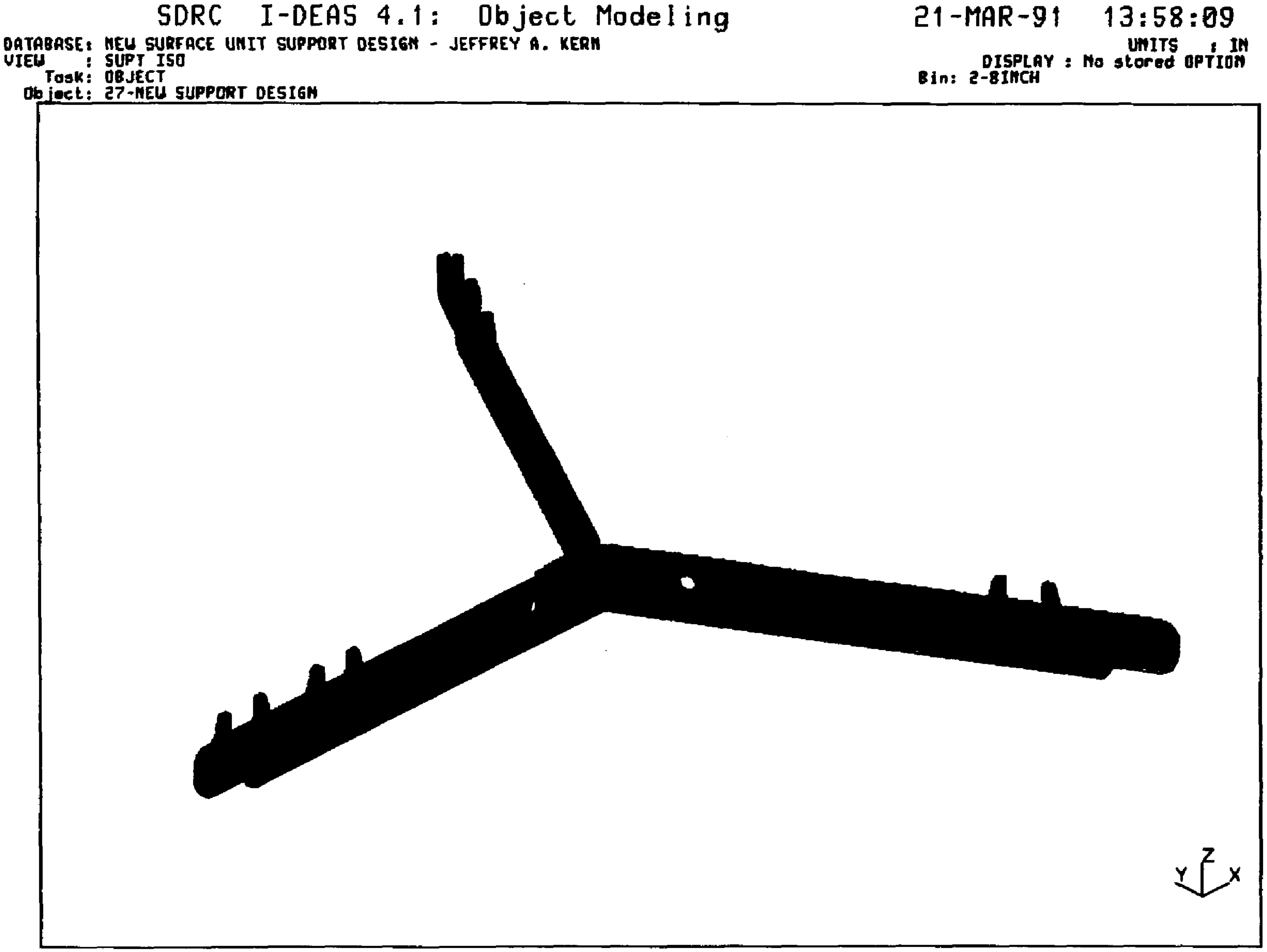
Figure 1. shows the current "delta" support design, which will be referred to as the "old" support design. Figure 2. shows the "Y" support design, which will be referred to as the "new" support design.

The complete surface unit assembly has several structural requirements. But, since only the support is being changed, its response to static loading is all that was analyzed. The units must pass a test under a twenty-five pound (25 lbs) static load. This test is conducted with weighted concave and convex pots full of boiling water, to simulate a "life time" of use by an avid "home canner". These pots create thermal variations in the unit's heater coil and can cause distortion of the coil. But, since the positions of the "tines" used to crimp the supports to the coils were not changed, this effect was not analyzed. The material used for the supports is a stainless steel alloy. Its properties do not vary significantly within the range of operating temperatures, and the supports' temperatures are reasonably uniform during this test. Therefore, no thermal expansion or thermal loading was modeled. The FEA performed was a static loading with a uniform flat plate load of 25 lbs coupled to the supports with "gap" elements. Both the new and old support designs were modeled for comparison of the resulting stresses. Several versions of the basic models were created with various mesh sizes and simply loaded to determine an optimum mesh size for the final analyses.



```
        SDRC  I-DEAS 4.1:  Object Modeling                 14-DEC-90  21:46:43
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN              UNITS   : IN
VIEW    : SUPT ISO                                           DISPLAY : No stored OPTION
   Task: OBJECT                                              Bin: 2-8INCH
 Object: 6-OLD SUPPORT DESIGN
```

FIGURE 1. - OLD "DELTA" SUPPORT DESIGN



```
                SDRC  I-DEAS 4.1:  Object Modeling               21-MAR-91  13:58:09
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                  UNITS   : IN
VIEW    : SUPT ISO                                          DISPLAY : No stored OPTION
    Task: OBJECT                                         Bin: 2-8INCH
  Object: 27-NEW SUPPORT DESIGN
```

FIGURE 2. - NEW "Y" SUPPORT DESIGN

SOFTWARE

Three of the MCAE software packages available to GE Appliance (GEA) employees in Louisville, KY, are: SDRC^{TM}[3] IDEAS^{TM}[4] MCAE system (solid modeling and engineering analyses); ANSYS^{R}[5] finite element analysis (FEA) programs; and CALMA DDM^{TM}[6] Computer Aided Design (CAD). They were used for the design of the new surface unit support, and analysis of the relative strength of the two supports.

If a design project is limited to one or two of the major design functions of: physical layout; product illustration; CAD drawing creation; FEA analyses; numerical control (NC) tool path creation; or mold filling analyses, it may be most economical to utilize specific software directly. But, in cases (such as this one) were several of these functions are required, the creation of a common data base which can be used in each of these areas can save time, and prevent errors caused by the creations of duplicate geometric models.

SDRC's I-DEAS software is a "complete" Mechanical Computer-Aided Engineering (MCAE) system. It consists of a set of integrated "families" of programs which access one data base that the user creates interactively through use of graphics computer terminals. The I-DEAS system uses Solids Modeling to create "solids" with the dimensional, material, and physical properties of the objects being modeled, which

[3] SDRC is a service trademark of Structural Dynamics Research Corp.

[4] I-DEAS is a trademark of Structural Dynamics Research Corp.

[5] ANSYS is a registered trademark of Swanson Analysis, Inc.

[6] CALMA DDM is a trademark of Calma Company, a wholly owned subsidiary of the General Electric Company.

are for use with the Engineering Analyses available within I-DEAS, or in external software packages, such as ANSYS. Thus, I-DEAS software was used extensively for this thesis.

Because of its capabilities for solving complex (e.g., nonlinear) problems, ANSYS finite element analysis programs have been used for the majority of the FEA work done by GEA. Because of the capabilities ANSYS offers beyond that of I-DEAS Engineering Analysis Family (e.g., its greater number of element types and non-linear capabilities) I-DEAS is most often used as a preprocessor of the FEA data to prepare models for transfer to ANSYS, and for post processing of the results from ANSYS solution runs. These methods were used for this thesis.

The CAD package available in I-DEAS, DRAFTING, utilizes the three dimensional (3D) geometry from the I-DEAS solid models for the creation of two dimensional (2D) CAD drawings. Thus, if the creation of CAD drawings is the only engineering function to be performed for a particular project, a full 3D CAD package may be more appropriate. This, coupled with GEA's CAD Department's use of Calma software, required Calma DDM CAD software to be the final CAD package used to create the production drawings for the new surface unit support. But, it is possible to transfer the 3D data from I-DEAS to Calma for the creation of DDM 3D CAD drawings; and this capability was used.

In cases where a project has a tight schedule, or where an external supplier will be developing the product and production drawings, or where 2D engineering "sketches" are all that is needed (e.g., for use by makers), DRAFTING is an excellent tool for the job. Also, DRAFTING can be used without an expensive CAD workstation. It can be accessed via terminals and PCs which have graphics capabilities, and a mouse or tablet type input devices. There are limitations on some of the fancier I-DEAS features that can be used, but almost any PC with a graphics display and a mouse can access GEA's installations of I-DEAS DRAFTING using Tektronix [7] emulation software. In fact, with the Personal Computer Systems Architecture (PCSA) network of DEC [8] software modules that GEA has installed, anyone on that system should be able to access I-DEAS and ANSYS using the EMU-TEK [9] Tektronix emulator resident on the network. Thus, DRAFTING was used for this thesis.

The I-DEAS system is available for both "mainframe" computer systems, and for personal computers (PCs) and self-contained workstations. At GEA, I-DEAS is available on select processors on the "AP35" and "AP1" VAX [10] nodes and workstations. Some installations do not include all of the I-DEAS program families, and thus guidance from GEA's computer systems personnel may be required to locate specific functions.

The I-DEAS system is menu driven. The version (Rev. 4.1) that was used has a scrolling menu system. Some I-DEAS packages are available for PCs and workstations which overlay the menus in display windows. In either case, these menus do not impeded the speed at which the experienced user can process information. Because, as well as picking the desired function from the menus, extensive strings of mnemonic acronyms

[7] Tektronix is a registered trademark of Tektronix, Inc.

[8] DEC is a trademark of Digital Equipment Corp.

[9] EMU-TEK is a trademark of FTG Data Systems.

[10] VAX is a trademark of Digital Equipment Corp.

can be input in single keyboard entries. This bypasses all intermediate menus, and thus function like a command, rather than a menu, driven system.

The ANSYS finite element software is a command line driven system. ANSYS model data bases can be created interactively during an ANSYS preprocessing session, or through the use of file editors to create input files. ANSYS data is stored in multiple files, some are in binary coded which can not be read or edited, and some are in ASCII format which can be read and edited. I-DEAS was used as the primary FEA preprocessor. The ANSYS input files (ASCII) were written out from I-DEAS, edited, and then transferred into ANSYS. Once the geometry and analysis data was in ANSYS, ANSYS's preprocessing was used to verify that the models were constructed as desired. ANSYS was used interactively to write out input files for the ANSYS solutions program. The ANSYS solution program was run in batch mode. Some post processing was be done in ANSYS, and then data was transfer back to I-DEAS for final display of the results.

I-DEAS is capable performing Computer-Aided Manufacturing (CAM) functions. It can be used to generated NUMERICAL_CONTROL (NC) tool paths for use in programing manufacturing machines, and to produce prototypes with laser and stereolythography equipment. But, for this thesis, a supplier's laser equipment was used to produce prototype parts. Also, GEA's Model Shop generates its NC tool paths with the Calma software. So, these functions were not used, and will not be discussed. Likewise, some of I-DEAS other software Families containing plastics molding, mold flow and cooling analyses, and TEST_DATA_ANALYSIS functions were not used in this thesis, and will not be discussed.

Both I-DEAS and ANSYS provide excellent color graphics, aiding the construction of models, and performance of analyses. Full advantage was taken of these capabilities for this thesis. It allowed various colors to be used to distinguish the different parts, materials, nodes, elements, and loads, etc. But, because of the difficulties presented in duplicating color illustrations, all figures that appear in this thesis are presented in black and white. They have been annotated to convey the information that is provided by these programs' color graphics.

To assist GEA's engineers, and hopefully others, in the application of these MCAE tools for future projects, appendices were written describing the I-DEAS and ANSYS commands used for this thesis. But, software (I-DEAS and ANSYS in particular) is in a state of constant evolution, and thus specific commands and features will vary from one version and installation to another. I-DEAS VERSION 4.1 and ANSYS version 4.3 were available to the author at the start of this thesis, and were used. But, higher revisions had become available by the time this thesis was finished, and other will become available in the future. The author recommends users take advantages of their advancements. Therefore, the author has attempted to limit this work's appendices to the discussion of basic commands that, in general, are expected to appear in all revisions of this software, due to their use in the basic methodology followed in the application of these tools. Thus, although specific commands will change, these appendices should help others know what basic functions are available and what they are capable, which will help them search within higher revisions of I-DEAS and ANSYS to locate and use improved versions of these commands. The specifics of these commands can easily be explored when using I-DEAS and ANSYS.

CHAPTER II

I-DEAS

INTRODUCTION

The I-DEAS system is based on constructing solid models of the physical parts that are being designed and analyzed. Solids are complete 3D representations of those parts. The models will generally be viewed or represented on the graphics display screen in wire frame form. They can also be displayed as shaded images or as wire frames with the rear portions hidden from view. These display options allow the solids to be seen as if they were physical parts. Both of these display options will be discussed later in this text. But, no matter how the solids are displayed, the models will contain complete part information. They not only contain geometric dimensions, like any wire frame 3D CAD model, they also contain all of the material properties of a true physical part. Although the material properties, such as density, are not always input in the early modeling phases, they do exist, even if they are left as the systems' default values. The solids' weights and centers of gravity etc. can be calculated by I-DEAS if desired. Also, once multiple parts are defined, systems of parts can be assembled, and their interactions and interferences can be analyzed.

I-DEAS provides several methods for creating solids. They can be constructed with basic solids, referred to as "primitives", which include cylinders, cones, and spheres. The user can also create unique solids, and solid "tools" by extruding or revolving profiles of the desired parts in space. Both primitives and user created solids can be joined with and cut from one another to construct more complex solids.

I-DEAS ENTITY CREATION

To create the solid models required for this thesis, I-DEAS was used on GE Appliance's computer system, and a data base, or set of "model files" was created. The methods used, and descriptions of the I-DEAS commands applied are presented in Appendix VII. Once these steps were taken, geometric entities and the solid models were created. The follow four chapters are devoted to the entities that were created for this thesis in the design of the new surface unit support, and for the analysis of old and new support designs.

CHAPTER III

WORKING GEOMETRY

INTRODUCTION

The construction of user created solids can be done by extruding or revolving profiles of the desired part in space. These profiles are 2D outlines of the cross section of the solid being constructed, which can include holes. These Profiles can be constructed directly through the creation of Profile Points, and connecting them with Profile Curves. But, most often it is necessary to create a layout of the parts being designed, or to rough out the design in 2D, in order to be able to create these profiles. In I-DEAS this is done through the creation of "working sets" or "wire frames".

GEOMETRY CREATION

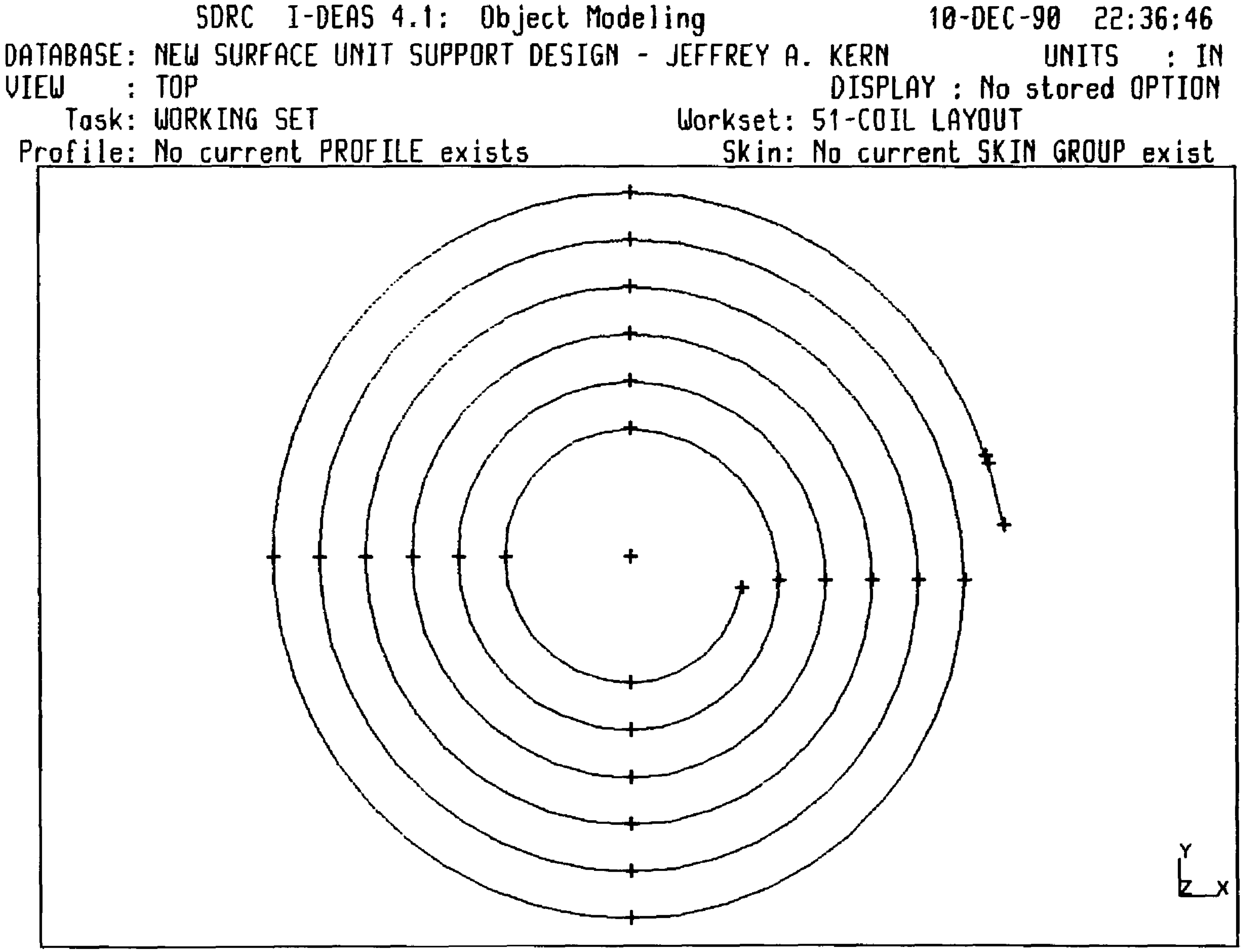
I-DEAS provides commands for the creation of: POINTS, LINES, CIRCLES, ARCS, FILLETS, SPLINES, entities FROM_OBJECTs; and for the specification of defaults for these entities. Each command contains multiple options for use in construction of these entities.

The menus provided for setting defaults allows the user to specify the way new entities will be labeled, their color, the line style used, and the symbol used to display points. These options are particularly useful in differentiating between entities. For example, the use of color to differentiate between various points and curves within a working set can be very helpful. Often it is desirable to create the geometry for several Profiles of separate parts in a single Working Set. Applying a different color to each part or profile's geometry can make cursor picking of their curves easier in the next operation, which is the creation of 2D Profiles. Once entities are created their attributes can be changed, without the need for recreating them, by using MODIFY commands.

The working geometry entity creation commands provide features found in many CAD packages. But, unlike a CAD packages (e.g., I-DEAS DRAFTING Family), only geometry is represented solid modeling, and dimensions can not be displayed. But, entities and dimensions can be verified through the use of commands presented in the LIST menus.

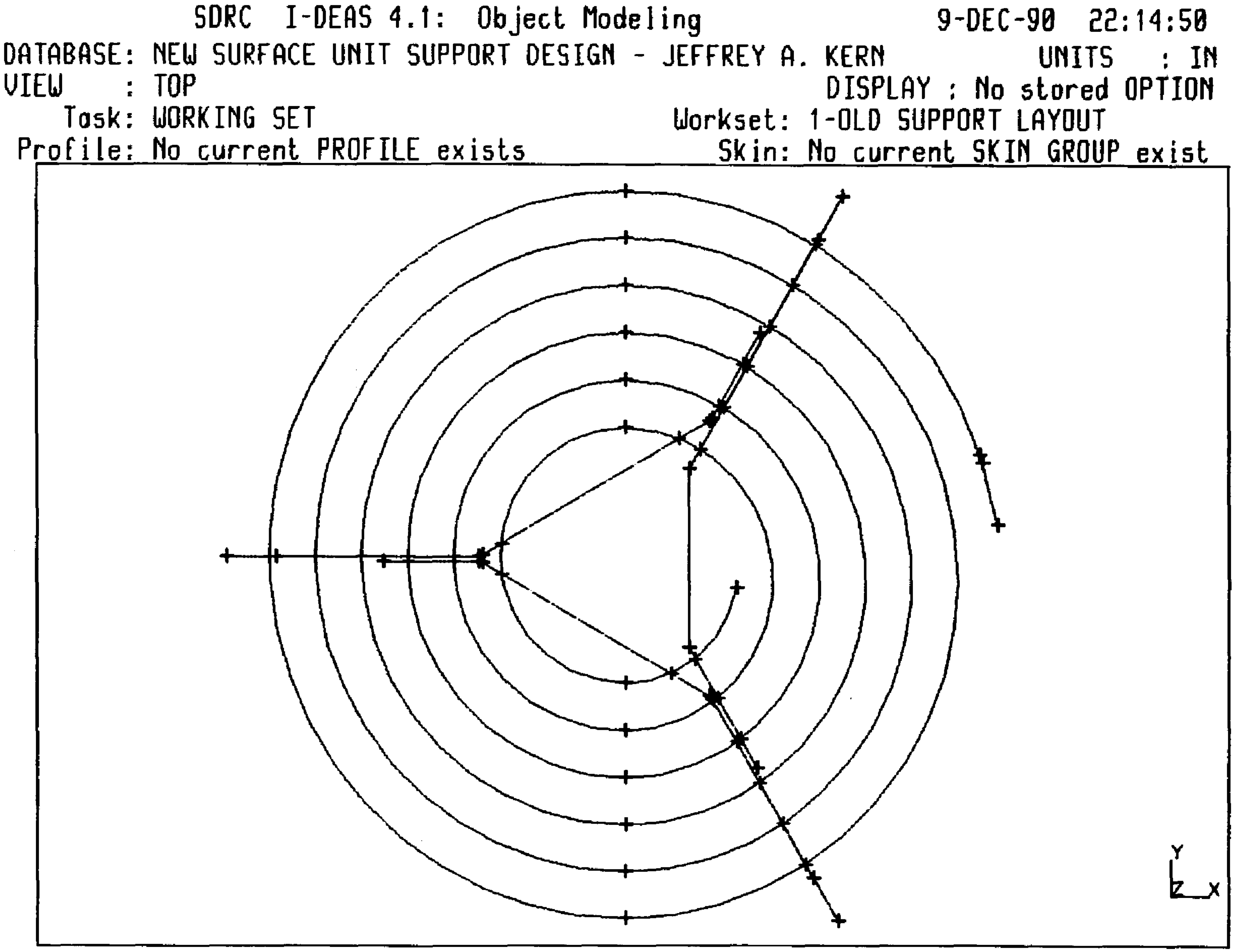
WORKING SETS

Several WORKING_SETs were created for this thesis. FIGURES 3. through 7. show some of these sets. FIGURE 3. shows the layout of the center line of the surface unit's heating coil. FIGURES 4. and 5. show the old and new coil support layouts created over the heating coil layout.



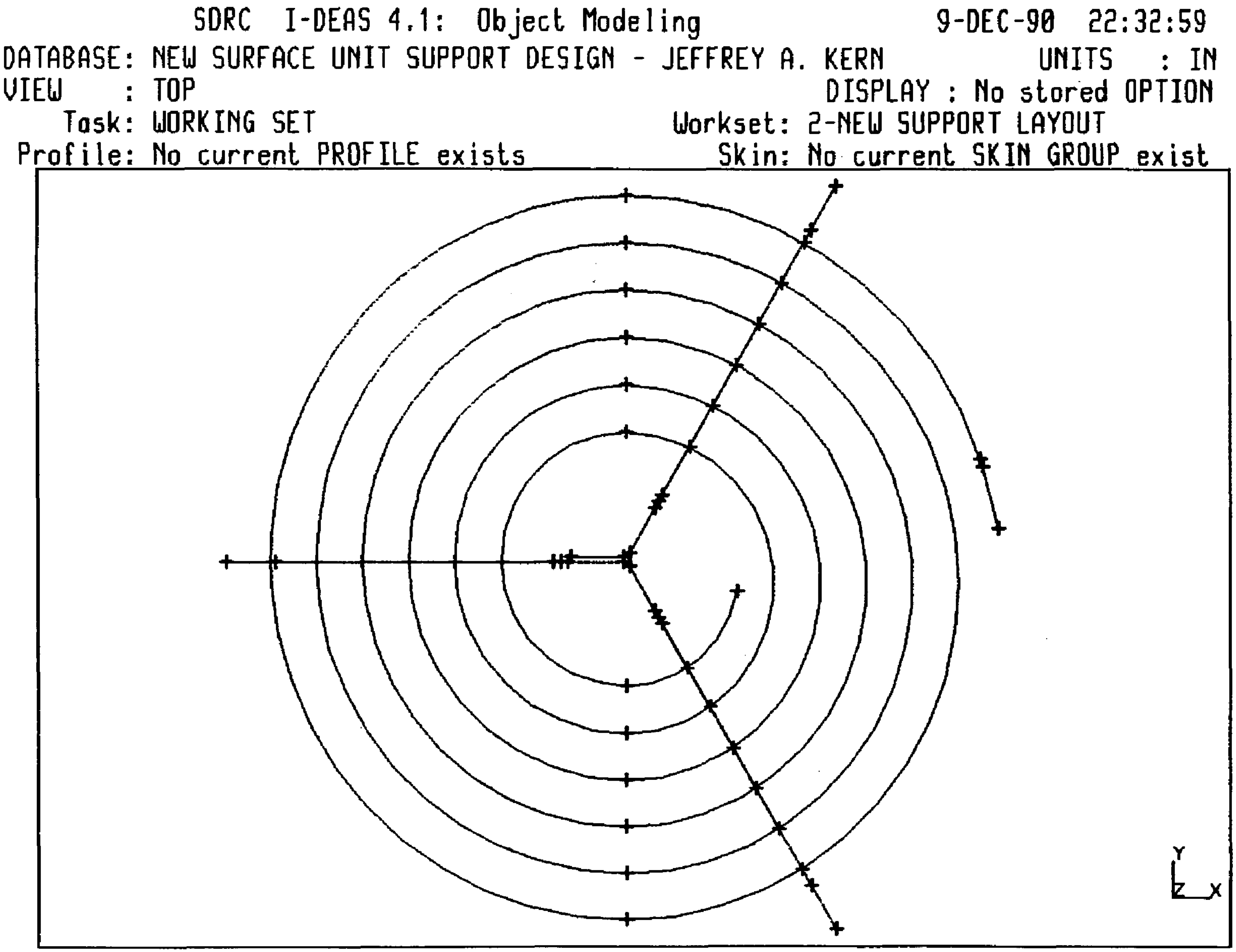
```
             SDRC  I-DEAS 4.1:  Object Modeling                 10-DEC-90  22:36:46
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : TOP                                             DISPLAY : No stored OPTION
    Task: WORKING SET                               Workset: 51-COIL LAYOUT
 Profile: No current PROFILE exists                    Skin: No current SKIN GROUP exist
```

FIGURE 3. - CENTER LINE OF THE SURFACE UNIT'S HEATING COIL



```
           SDRC  I-DEAS 4.1:  Object Modeling                    9-DEC-90  22:14:50
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN           UNITS   : IN
VIEW    : TOP                                             DISPLAY : No stored OPTION
    Task: WORKING SET                                Workset: 1-OLD SUPPORT LAYOUT
 Profile: No current PROFILE exists                     Skin: No current SKIN GROUP exist
```

FIGURE 4. - OLD COIL SUPPORT LAYOUT



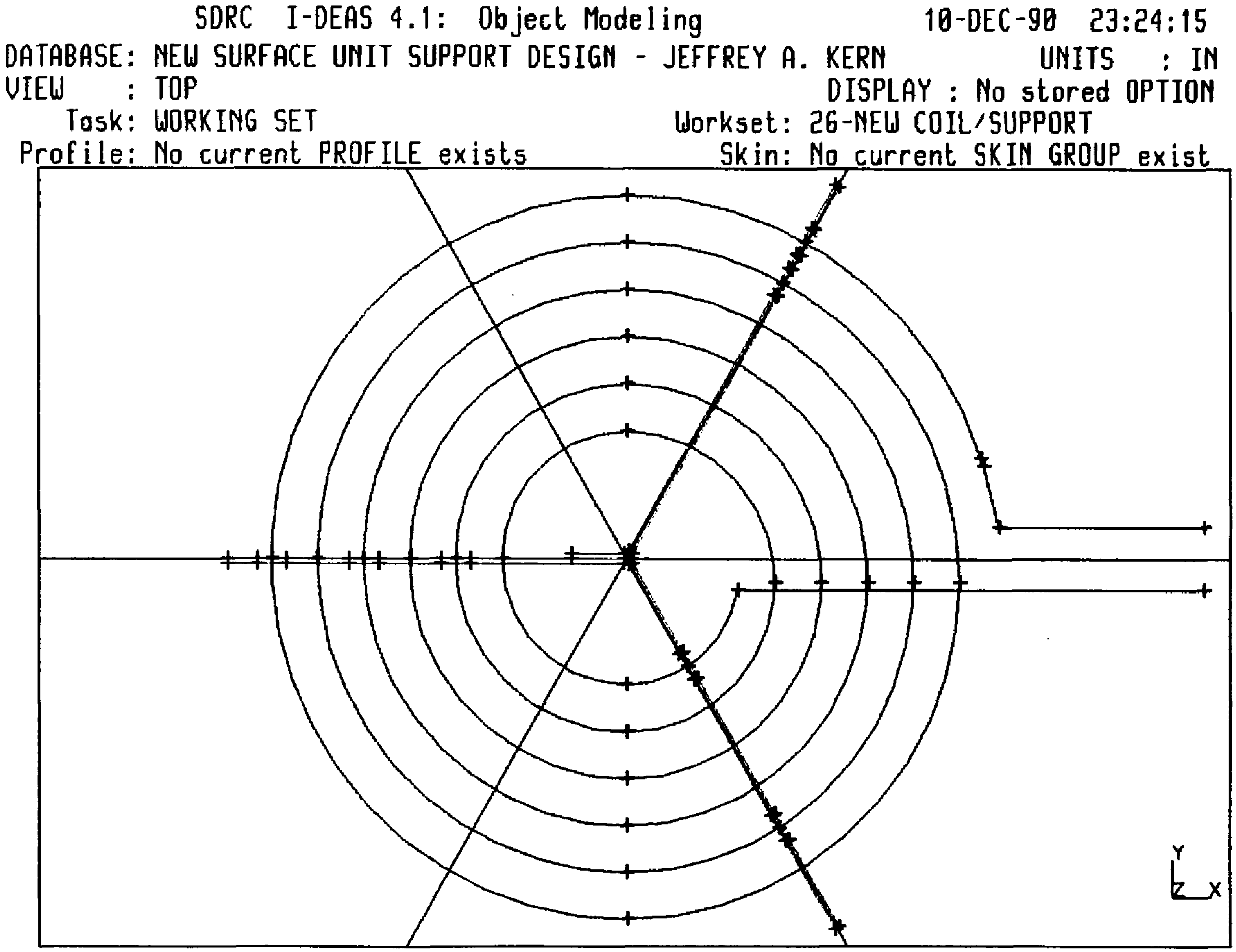
```
             SDRC  I-DEAS 4.1:  Object Modeling                  9-DEC-90  22:32:59
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : TOP                                               DISPLAY : No stored OPTION
    Task: WORKING SET                                  Workset: 2-NEW SUPPORT LAYOUT
 Profile: No current PROFILE exists                       Skin: No current SKIN GROUP exist
```

FIGURE 5. - NEW COIL SUPPORT LAYOUT

These working_sets were created without the need to recreate the heating coil for each of the other layouts. The "COIL LAYOUT" WORKING_-SET was stored. Then, it was copied into "NEW SUPPORT LAYOUT" and "OLD SUPPORT LAYOUT" WORKING_SETs. Once this was done, and the new sets were individually made active, and then the CREATE commands were used to create the geometry for the supports. This could have been done by creating each of the support layouts without the coil's geometry, and then it could have been added to them with an APPEND command.

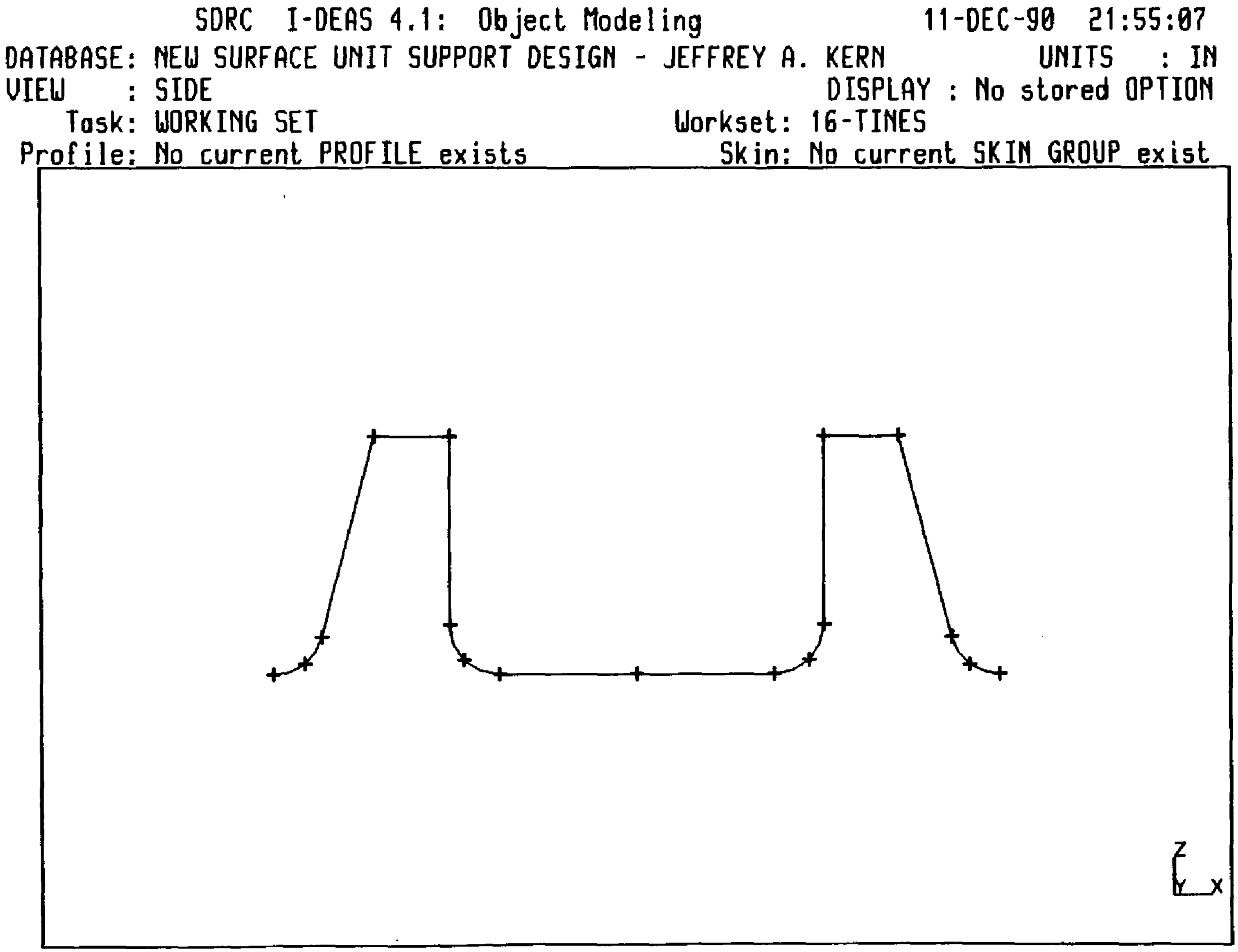
Several other WORKING_SETs were created. Complete top views of the supports were created. This allowed profiles to be created for each of the individual pieces of the supports, which were used to extrude their basic shapes. The complete process is explained in the following chapters. FIGURE 6. shows WORKING_SET which was created for the new support design.

Figure 7. shows the layout of a pair of tines used to attach the supports to the heating coil. They were added into the top edge of the supports by creating profiles of these tines, extruding them into solids, positioning then on the basic supports, and then using a BOOLEAN JOIN operation.



```
             SDRC  I-DEAS 4.1:  Object Modeling              10-DEC-90  23:24:15
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : TOP                                             DISPLAY : No stored OPTION
    Task: WORKING SET                                 Workset: 26-NEW COIL/SUPPORT
 Profile: No current PROFILE exists                      Skin: No current SKIN GROUP exist
```

FIGURE 6. - NEW SUPPORT'S WORKING_SET



```
              SDRC  I-DEAS 4.1:  Object Modeling              11-DEC-90  21:55:07
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : SIDE                                          DISPLAY : No stored OPTION
    Task: WORKING SET                              Workset: 16-TINES
 Profile: No current PROFILE exists                   Skin: No current SKIN GROUP exist
```

FIGURE 7. - ATTACHMENT TINES WORKING_SET

CHAPTER IV

OBJECT PROFILES

INTRODUCTION

I-DEAS Solid Modeling was used to construct 2D cross sectional outlines of the solids, called "profiles". Profiles can be used in the creation of solid models along with EXTRUDE, REVOLVE, or SKIN commands in object modeling. These profiles can include holes for the creation of hollow solids, or solids that represent sheet metal parts, etc. The use of profiles in the creation of the solids is explained in Chapter V.

PROFILE CREATION

A profile can contain up to 100 connected curves that lie in one 2D plane. Each curve will be given a different label which can be displayed and listed. These labels will be different than the labels of any corresponding working set curve, and are independent of the curve type. The CREATE_PROFILE command is used to create "profile points" through which the I-DEAS program then fits curves to create a profile. The profile points can be created by selecting working set points or by direct creation with options presented within the create command.

The profile points (like the profile curves) will each be given unique labels which are independent of the working set's points. Profile points can be modified and deleted without effecting any working set points used to create them. The AUTO_CHAINING command is used to create a profile through the selection of working set curves that are connected end-to-end in a closed loop.

CREATE_PROFILE COMMAND

If the CREATE_PROFILE command is used to construct a profile, the screen cursor is used to pick working geometry points. The cursor will remain active until the user types a "C" (CLOSE), "D" (DONE), or "$" (ABORT). The user first picks a point at the end of the initial curve. Then enough additional points are selected to define the curve. One additional point is required to define a line; two additional points define an arc; and three to twenty-four additional points define a spline. Once the curve is sufficiently defined, the "F" key is depressed or the menu is used to select FIT, and I-DEAS then fits a profile curve to the newly created profile points. This is repeated for each of the curves needed to create a closed profile. The last point in each curve automatically becomes the first point for the next. The profile can be closed either by picking the first profile point created as the last point in the final curve, or by pressing "C" or selecting CLOSE to complete the final curve.

Closed profiles are required for use in creating solids. But, the CREATE_PROFILE can be stopped with the return key or the DONE command, leaving the profile open. An open profile can be added to with the CREATE_PROFILE command, connecting new curves to the last curve created.

AUTO_CHAINING COMMAND

The AUTO_CHAINING command takes advantage of the curves that were created in the WORKING_SET task. It will automatically create as may profiles (up to a maximum of ten) in a given plane that will form closed loops of user selected curves.

After AUTO_CHAINING is initiated, this command will prompt the user to select the curves from which the search for loops should be limited. The screen cursor can be used to select only those curve that are desired (in any order). Or, SCREEN_AREAs can be used to select groups of curves.

If the later option is used, several closed profiles may result. This is not a problem. It will just require that they all be stored. Then the user will need to determine which are to be used, and the others can be deleted with the MANAGE menu's DELETE commands. If the number of lines and curves in the desired profile(s) are known, this can be simplified by using the LIST BOUNDARY ALL command to review all the profiles that were created.

Once all of the curves that are to be considered in the creation of the profile(s) have be selected, a <RETURN> or DONE (D) is input. Then I-DEAS will ask for the selection of a curve to start the profile(s). If a line is selected, I-DEAS will ask for another curve which connects to the last curve, so that it can determine the plane in which the profile(s) lie. When this is completed, I-DEAS will automatically create all profiles that can be constructed with the selected curves that lie in this plane. The total number of profiles will be displayed, and the user will be prompted to name each profile created and stored.

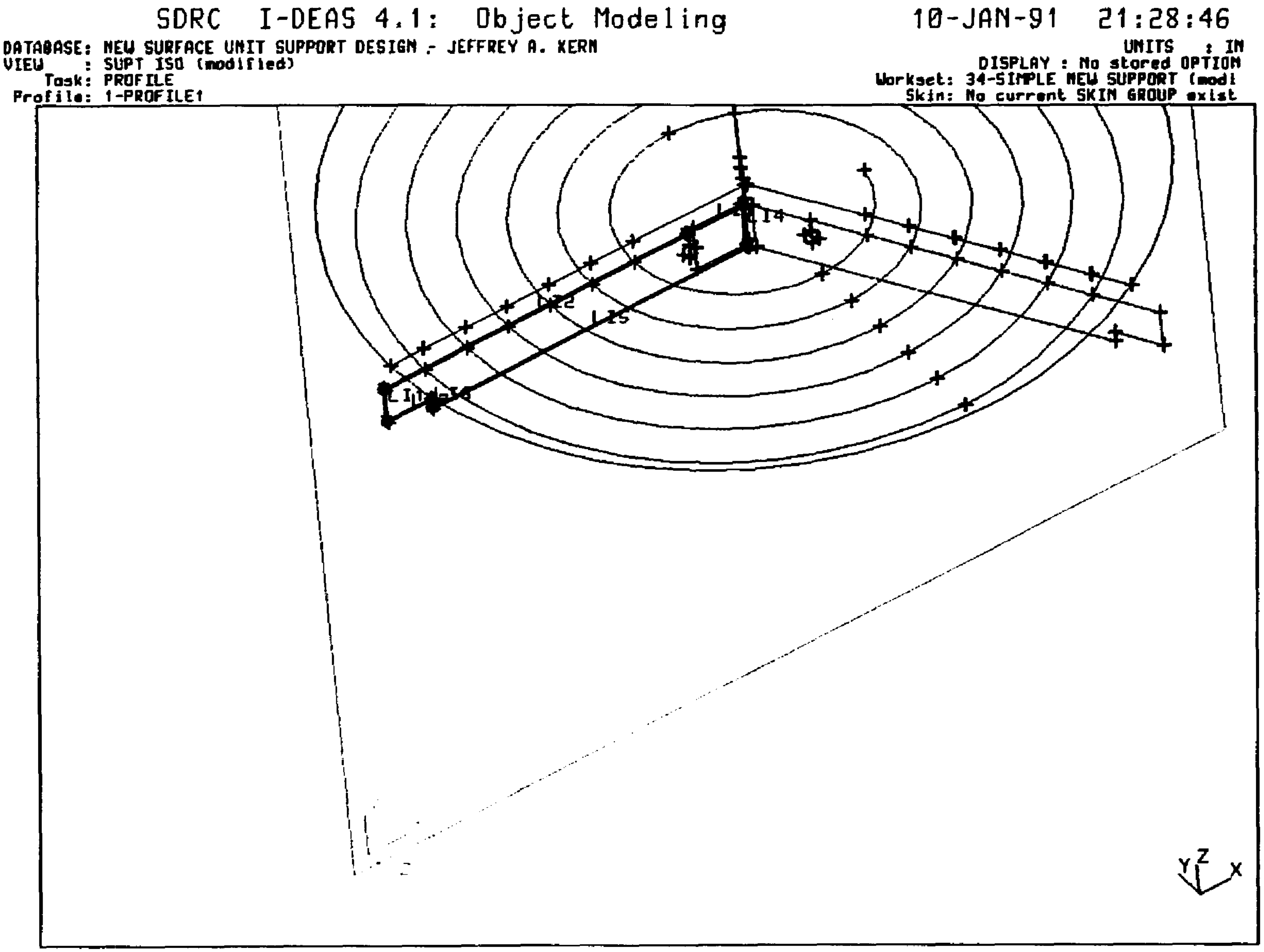
WORK_PLANE ORIENTATION

In I-DEAS Rev. 4.1, if profile(s) do not lie on the active "WORK_PLANE", I-DEAS will automatically position the profile(s) on the WORK_PLANE. Also, if the user creates a custom WORK_PLANE, which is in the same plane as the desired profile(s)', but that is not on the one of the global XY, XZ, or YZ planes, the resulting profile(s) may not be oriented on top of the working set's curves. In either case, the resulting profile(s) may need to be oriented on the WORK_PLANE before they are use to create solids in the OBJECT_3D task; or they may be used as is, and the solids created may need to be oriented before they are used in construction of a larger object. This is especially true about profiles used to create solids with the REVOLVE command.

EXAMPLE PROFILES

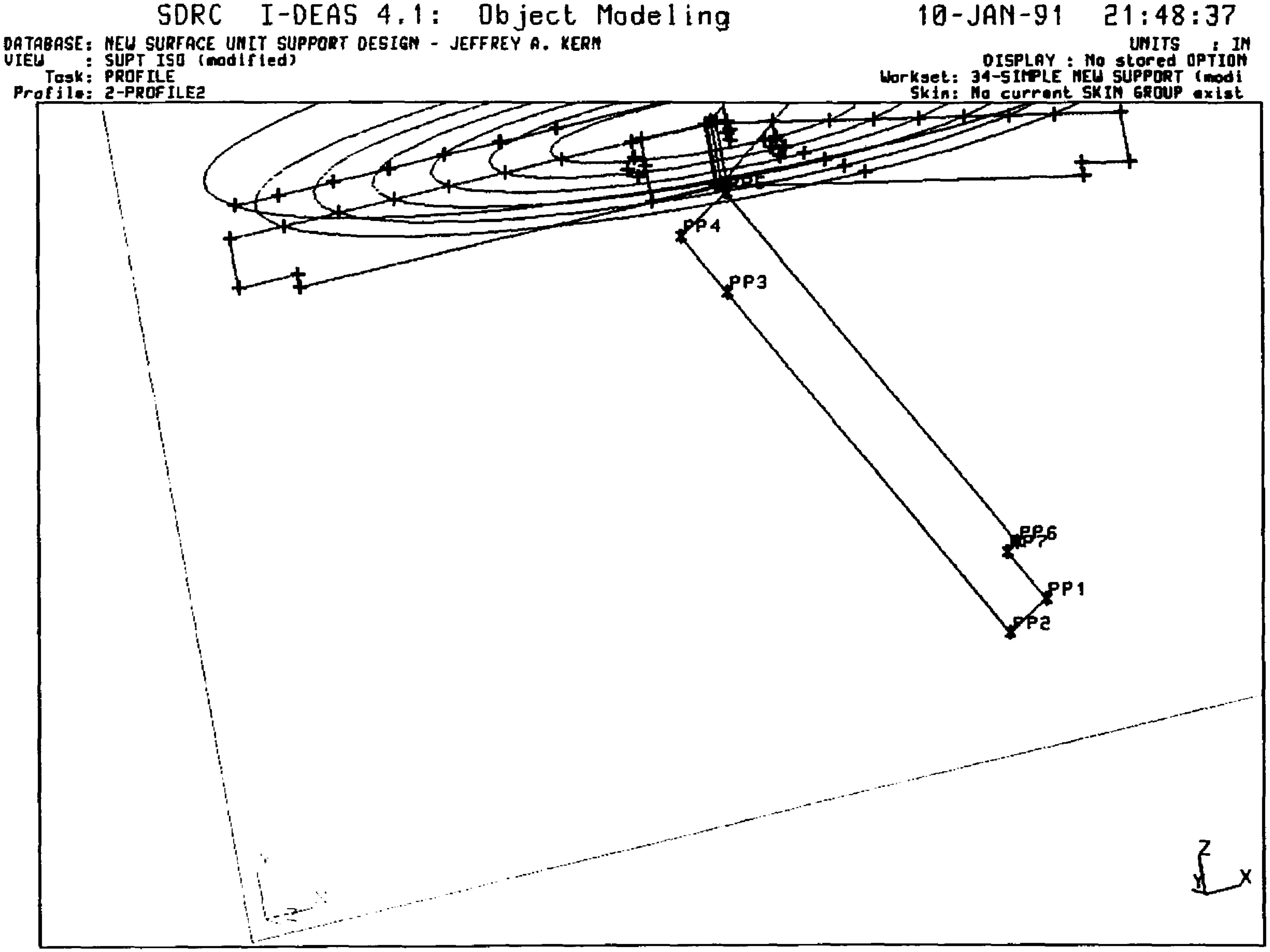
The first example of one of the profiles created for this thesis is shown in FIGURE 8. This profile was created with a WORK_PLANE which lies on the global XZ plane. The global system's axes triad is displayed in the lower right corner of the view port. The WORK_PLANE's triad is displayed in the lower right corner of its outline. This example shows that a profile created on a WORK_PLANE that lies on a global plane will be place on top of the curves used to create it.

The second example, shown in FIGURE 9, is of a profile created for the rear leg of the new support that extends into the negative Y half of the global 3D space. This shows that with the WORK_PLANE still laying on the global XZ plane, the resulting profile is not positioned on the geometry used to create it. It is correct. But, the solid that was created through the extrusion of this profile required orientation.



```
SDRC  I-DEAS 4.1:  Object Modeling              10-JAN-91  21:28:46
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN         UNITS   : IN
VIEW    : SUPT ISO (modified)                        DISPLAY : No stored OPTION
    Task: PROFILE                        Workset: 34-SIMPLE NEW SUPPORT (modi
 Profile: 1-PROFILE1                       Skin: No current SKIN GROUP exist
```

FIGURE 8. - FIRST SAMPLE 2D PROFILE



```
      SDRC  I-DEAS 4.1:  Object Modeling            10-JAN-91  21:48:37
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : SUPT ISO (modified)                            DISPLAY : No stored OPTION
    Task: PROFILE                                 Workset: 34-SIMPLE NEW SUPPORT (modi
 Profile: 2-PROFILE2                                 Skin: No current SKIN GROUP exist
```

FIGURE 9. - SECOND SAMPLE PROFILE

CHAPTER V

SOLID MODEL CREATION

INTRODUCTION

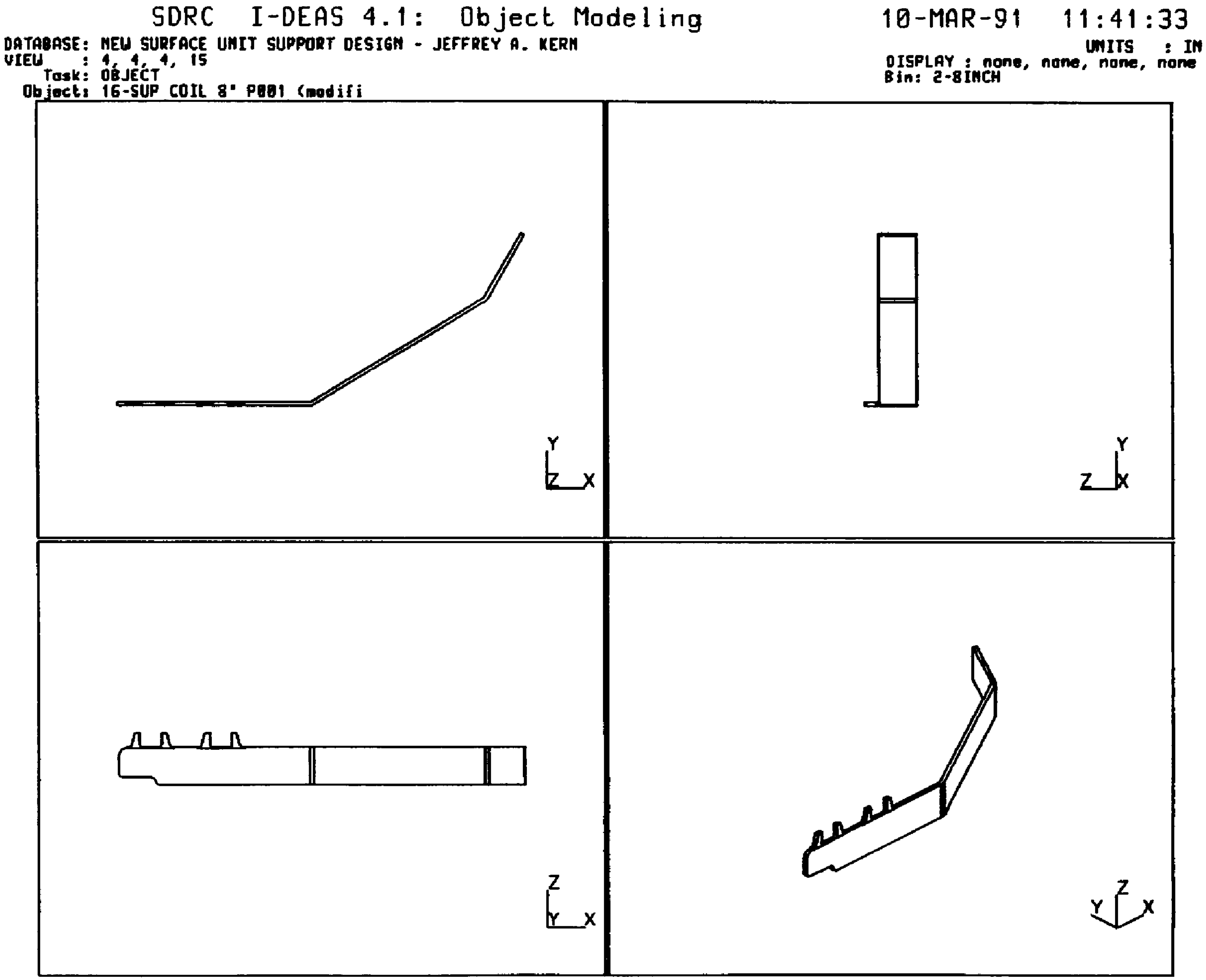
The methods for constructing solids were identified in Chapter 2 as: creation of basic solids, or "primitives" (blocks, cones, cylinders, spheres, and tubes); creation of user defined solids through the extrusion and revolution of 2D profiles; and the combination of these subparts with joining and cutting operations. These methods are described in Appendix VIII, and the solids that were created for this project will be discussed are discussed in this chapter.

SURFACE UNIT SUPPORT SOLID MODELS

Solid models were created for each of the pieces of the old support and new support designs. Profiles were created of the side views of each straight section of the support pieces, and of the top views of their bends. These profiles were extruded to create parts of each support piece. Then the parts where JOINed to form the complete pieces. The tines, which are used to crimp the supports to the heater, where created as standard solids. They where then oriented at the proper positions on the support pieces, and CUT from the basic supports, and then JOINed to them.

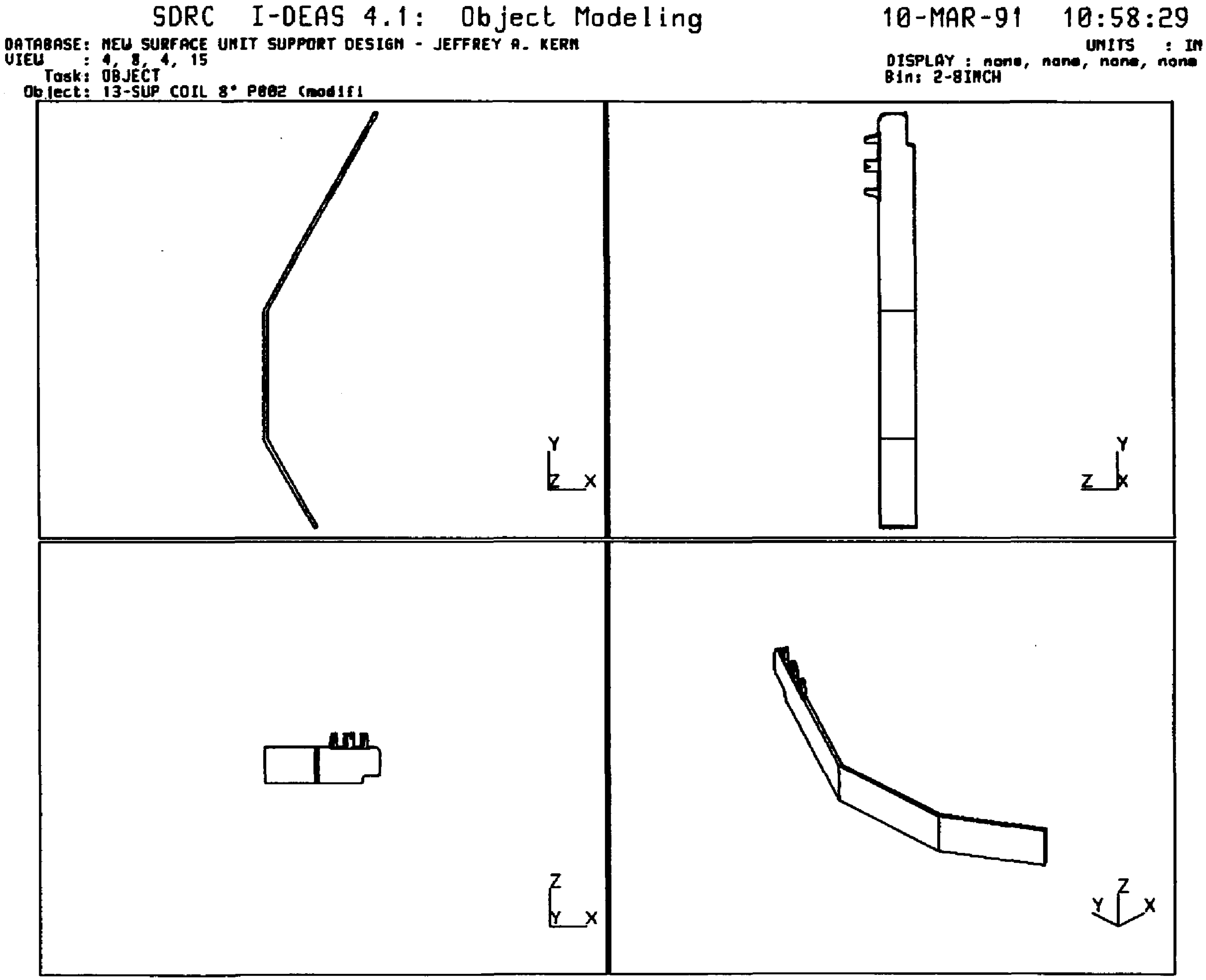
Figures 10. through 13. show the solids created of the for piece parts that make up the old support design. Figure 21. shows the old support assembly. It was created by orienting the three pieces properly in 3D space, and then JOINing them together. This was done to produce a complete assembly which could be used for the development of a finite element analysis (FEA) model for structural analysis of the old design. A system assembly could also have been created of this assembly to analyze is assembly and mechanical properties, as will be discussed in the chapter on *SYSTEM ASSEMBLY*.

Figures 14. and 15. show the solids that were created for the new support design, and Figure 16. shows them assembled.



```
         SDRC  I-DEAS 4.1:  Object Modeling                10-MAR-91  11:41:33
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN            UNITS   : IN
VIEW    : 4, 4, 4, 15                                        DISPLAY : none, none, none, none
   Task: OBJECT                                              Bin: 2-8INCH
  Object: 16-SUP COIL 8" P001 (modifi
```

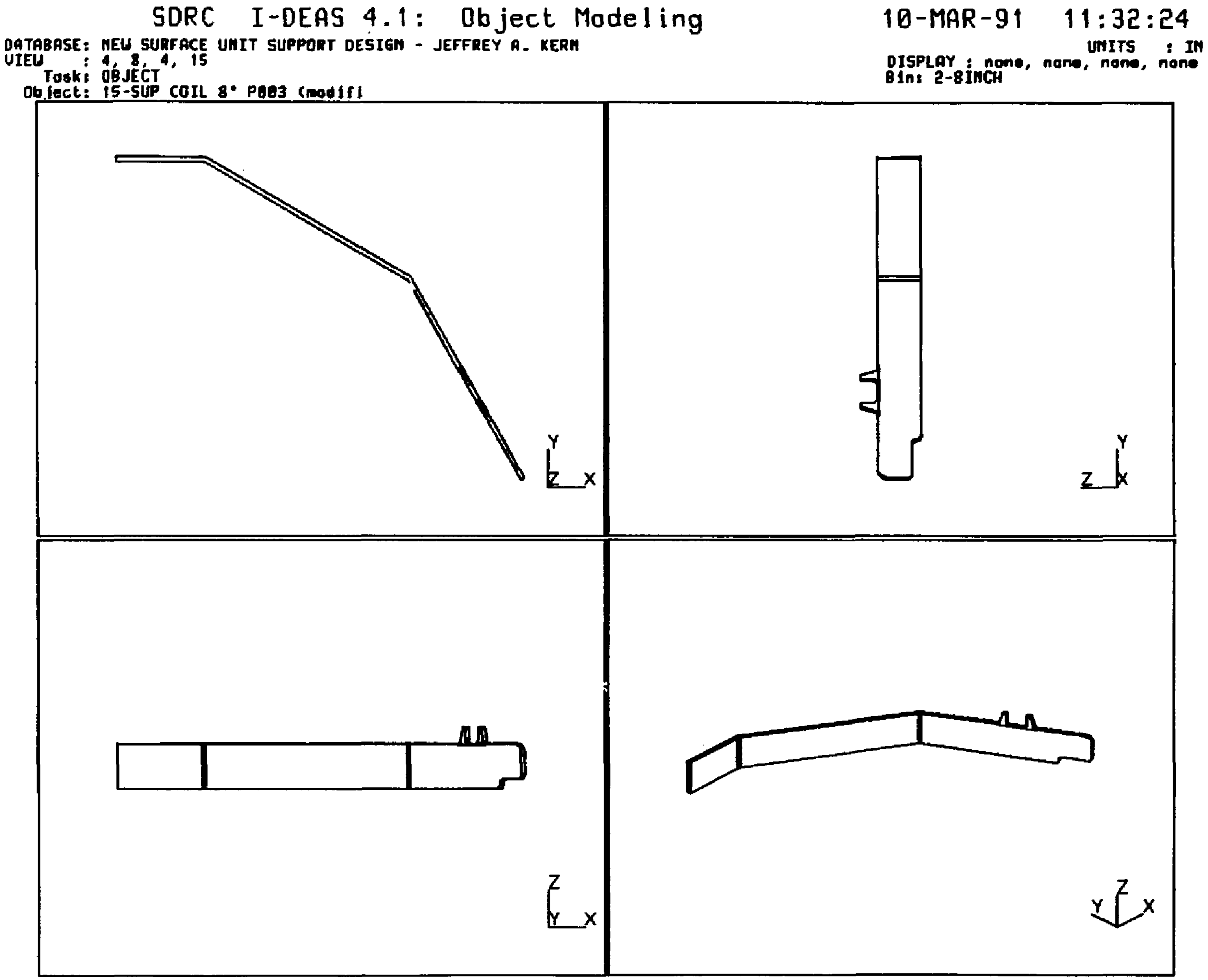
FIGURE 10. - OLD SUPPORT'S PART 1 SOLID



SDRC I-DEAS 4.1: Object Modeling 10-MAR-91 10:58:29

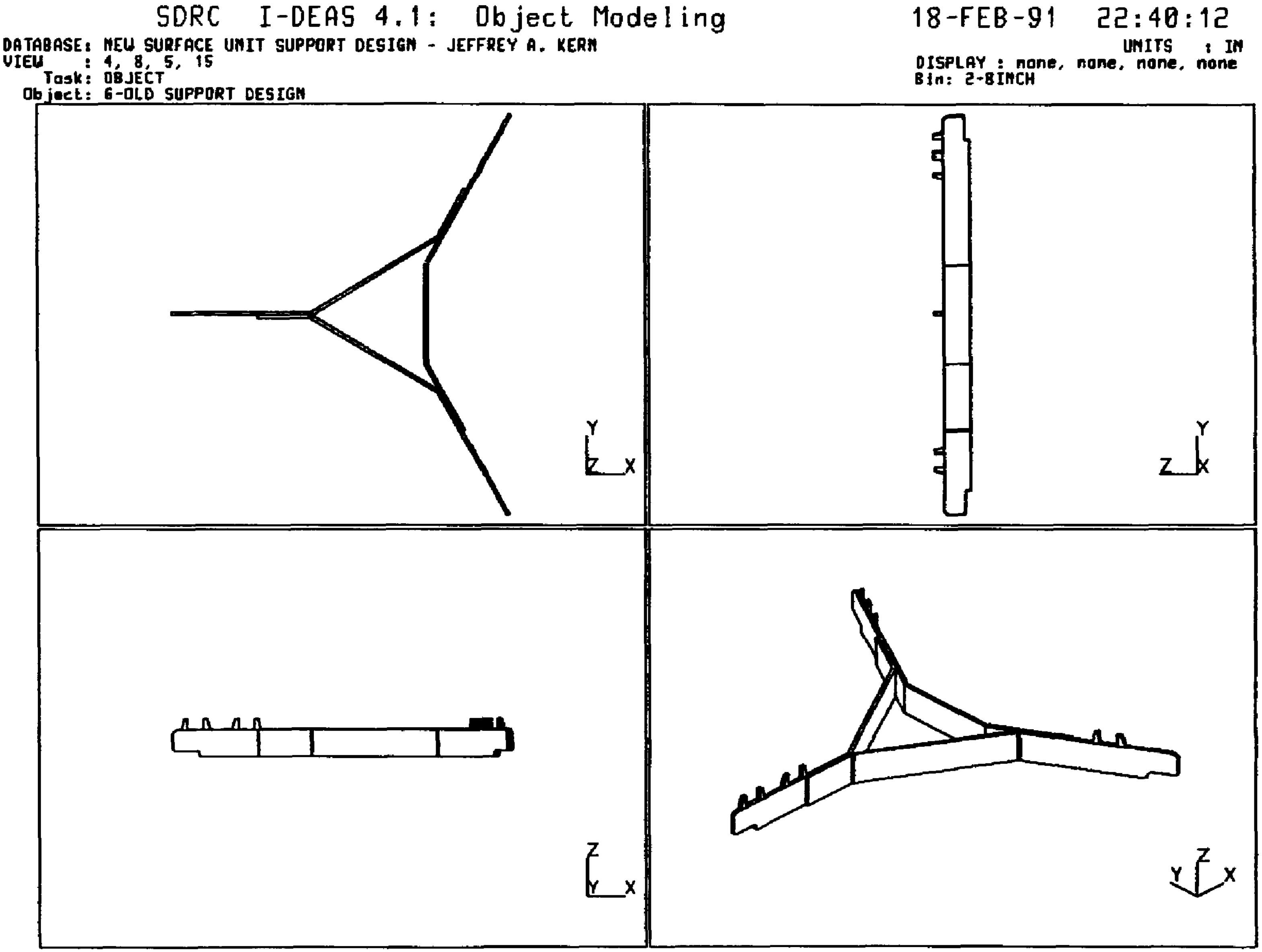
```
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : 4, 8, 4, 15                                                DISPLAY : none, none, none, none
    Task: OBJECT                                                     Bin: 2-8INCH
  Object: 13-SUP COIL 8" P002 (modifi
```

FIGURE 11. - OLD SUPPORT'S PART 2 SOLID



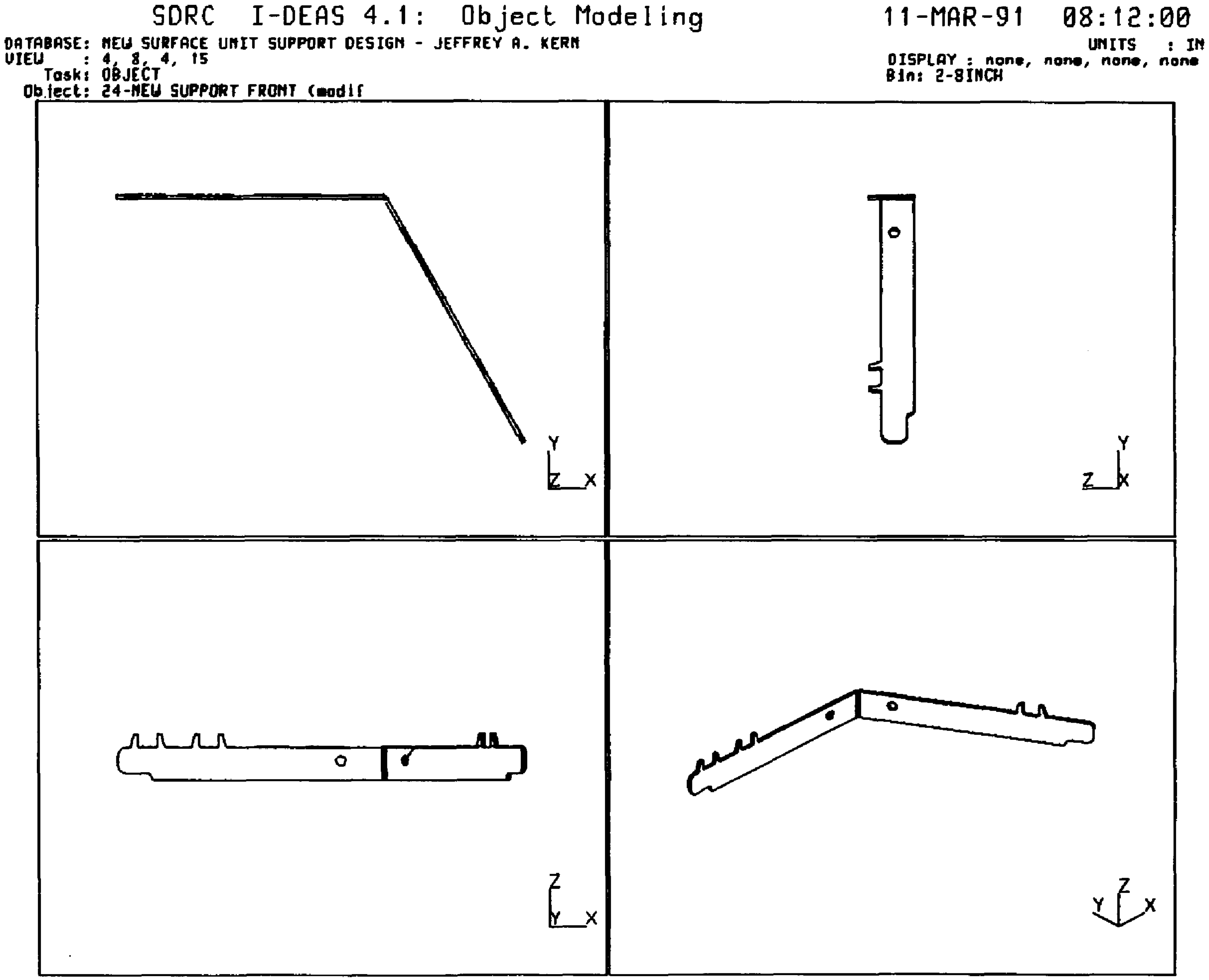
```
SDRC  I-DEAS 4.1:  Object Modeling                10-MAR-91  11:32:24
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : 4, 8, 4, 15                             DISPLAY : none, none, none, none
   Task: OBJECT                                   Bin: 2-8INCH
 Object: 15-SUP COIL 8" P003 (modif)
```

FIGURE 12 . - OLD SUPPORT'S PART 3 SOLID



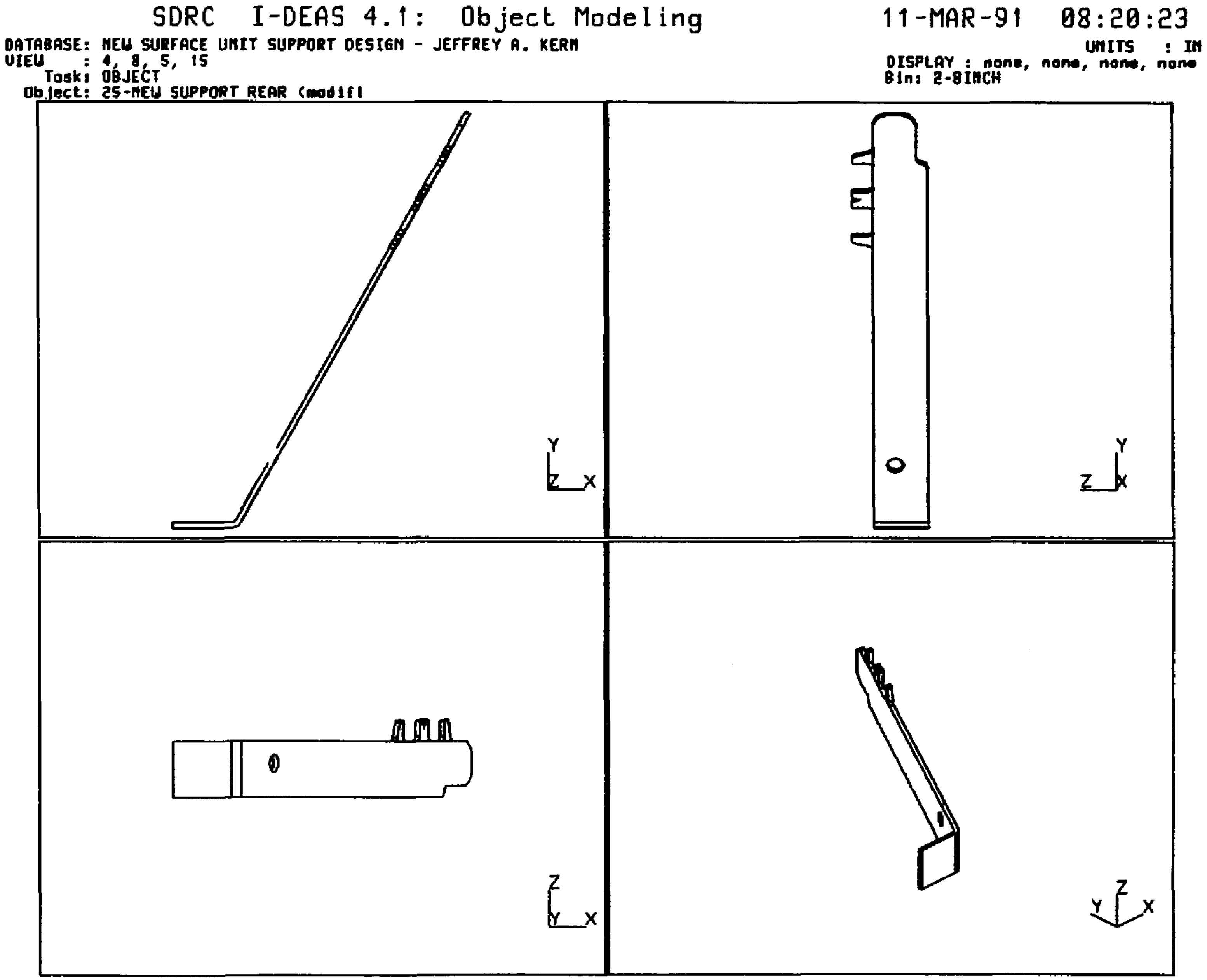
```
SDRC  I-DEAS 4.1:  Object Modeling                          18-FEB-91  22:40:12
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN              UNITS   : IN
VIEW    : 4, 8, 5, 15                                   DISPLAY : none, none, none, none
   Task: OBJECT                                         Bin: 2-8INCH
 Object: 6-OLD SUPPORT DESIGN
```

FIGURE 13. - OLD SUPPORT ASSEMBLY'S SOLID



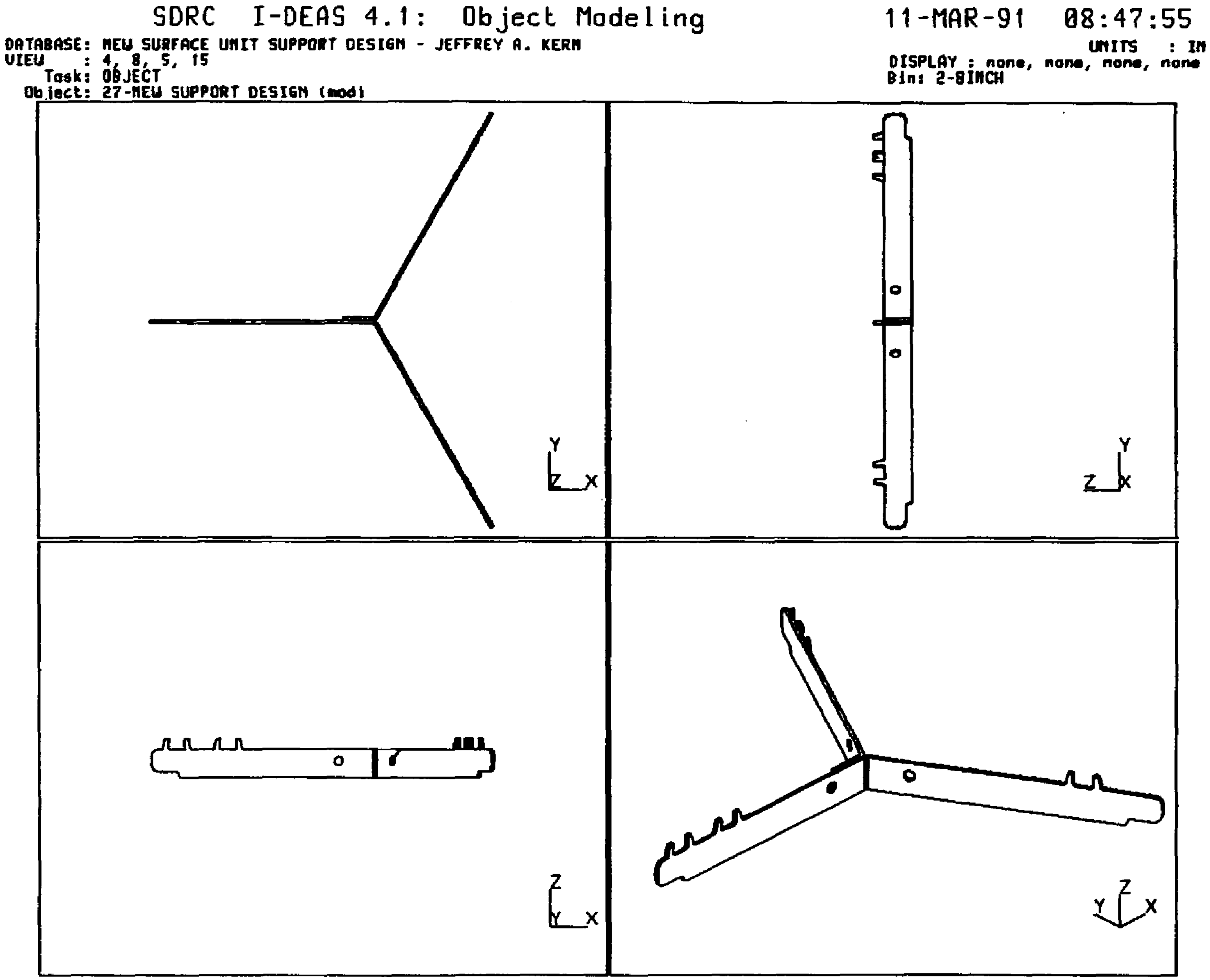
```
                SDRC  I-DEAS 4.1:  Object Modeling                 11-MAR-91  08:12:00
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                     UNITS   : IN
VIEW    : 4, 8, 4, 15                                   DISPLAY : none, none, none, none
    Task: OBJECT                                        Bin: 2-8INCH
  Object: 24-NEW SUPPORT FRONT (modif
```

FIGURE 14. - NEW SUPPORT'S FRONT PART SOLID



```
        SDRC  I-DEAS 4.1:  Object Modeling                 11-MAR-91  08:20:23
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN            UNITS   : IN
VIEW    : 4, 8, 5, 15                                   DISPLAY : none, none, none, none
    Task: OBJECT                                        Bin: 2-8INCH
  Object: 25-NEW SUPPORT REAR (modifi
```

FIGURE 15 . - NEW SUPPORT'S REAR PART SOLID



SDRC I-DEAS 4.1: Object Modeling 11-MAR-91 08:47:55

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN

VIEW : 4, 8, 5, 15

Task: OBJECT

Object: 27-NEW SUPPORT DESIGN (mod)

UNITS : IN

DISPLAY : none, none, none, none

Bin: 2-8INCH

FIGURE 16 . - NEW SUPPORT'S ASSEMBLY SOLID

CHAPTER VI

ASSEMBLY MODELING

INTRODUCTION

Chapter V discussed the solid models created for this thesis. In that discussion, several objects were combined, using the JOIN command, to form larger objects. For this thesis, each of the surface unit supports' individual pieces were joined into complete support assemblies in this manner. Thus, the surface unit supports can now be treated as single objects. Therefore, these solid models represent the complete support assemblies' weight, material volume, and center of gravity, etc.

But, in general, when a system is being designed the logical progression is to first design individual components, then combine the components into subassemblies, and finally, combine the subassemblies into complete system assemblies, and investigate their properties. This is the methodology applied in the structure of the I-DEAS Solid Modeling family of program modules. The previous chapters in this text have covered the use of I-DEAS Solid Modeling's Object Modeling to create individual components, as solid models. Appendix IX is devoted to the description of work done with I-DEAS Object Modeling's Assembly Modeling commands. With these commands, the individual piece of the surface unit supports, and complete heater assemblies were assembled as that would be if they were not need as combined objects for the FEA analyses.

SURFACE UNIT SYSTEM ASSEMBLIES

System assemblies were create for the surface unit assemblies, using the old and new support assemblies. Tables 1. and 2. show the hierarchy listing for these system assemblies.

TABLE 1. - SYSTEM ASSEMBLY HIERARCHY FOR THE OLD SURFACE UNIT ASM

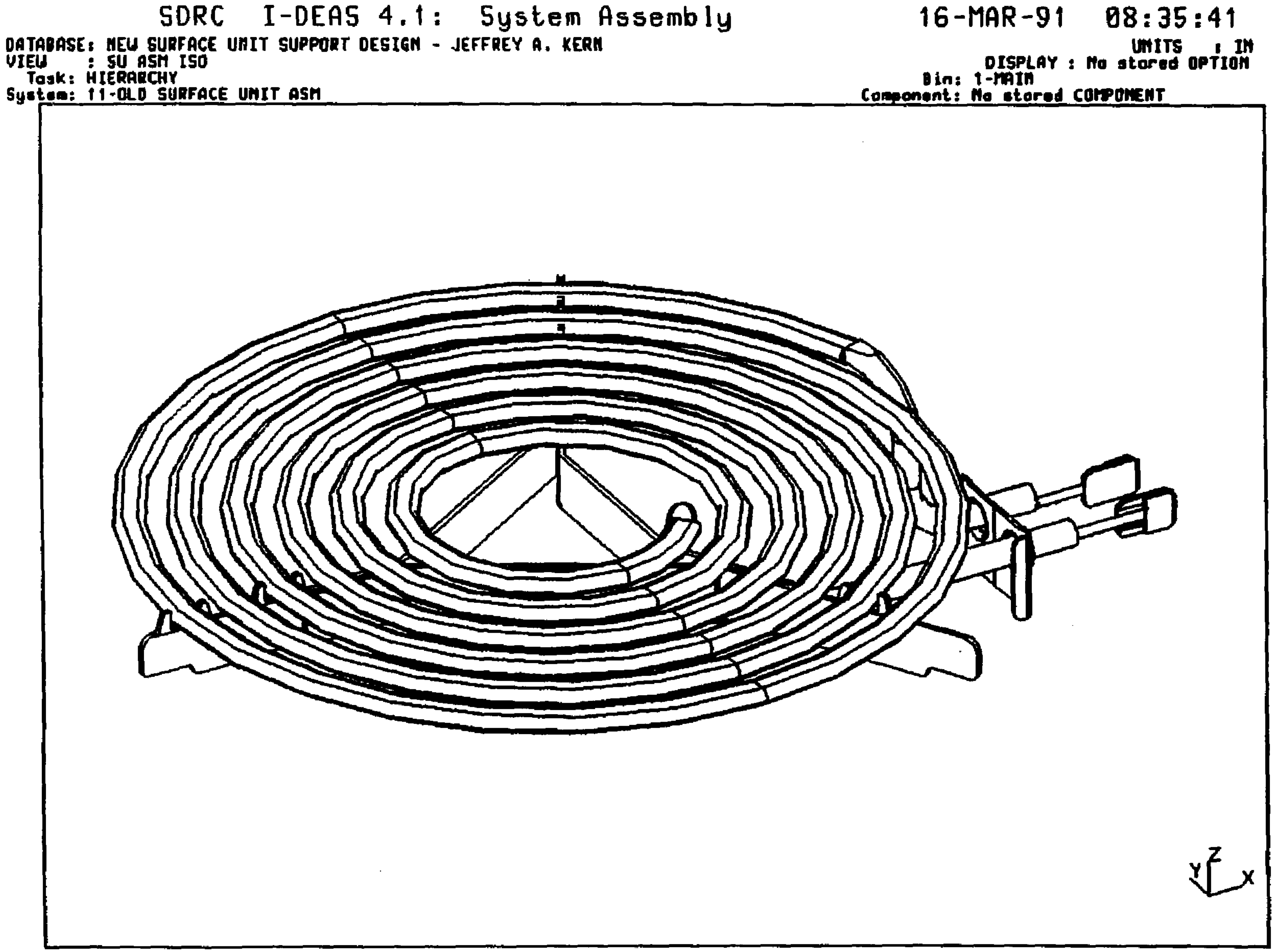
```
System: 11-OLD SURFACE UNIT ASM, Bin1
 :...29-System: OLD SUPPORT ASM, Bin1
 :    4-IN1
 :      :...26-SUP COIL 8" P001, Bin1
 :      :    5-IN1
 :      :...27-SUP COIL 8" P002, Bin1
 :      :    6-IN1
 :      :...28-SUP COIL 8" P003, Bin1
 :           7-IN1
 :....8-CLIP GROUNDING, Bin1
 :    3-GROUND CLIP
 :....9-COIL ASM 8" P.I., Bin1
      2-COIL ASM
```

TABLE 2. - SYSTEM ASSEMBLY HIERARCHY FOR THE NEW SURFACE UNIT ASM

```
System: 21-NEW SURFACE UNIT ASM, Bin1
 :...24-System: NEW SUPPORT ASM, Bin1
 :    2-IN1
 :      :...22-NEW SUPPORT FRONT, Bin1
 :      :    3-IN1
 :      :...23-NEW SUPPORT REAR, Bin1
 :      :    4-IN1
 :      :...25-TRIM CENTER 8", Bin1
 :           5-IN1
 :....9-COIL ASM 8" P.I., Bin1
 :    6-HEATING COIL
 :....8-CLIP GROUNDING, Bin1
      7-IN1
```

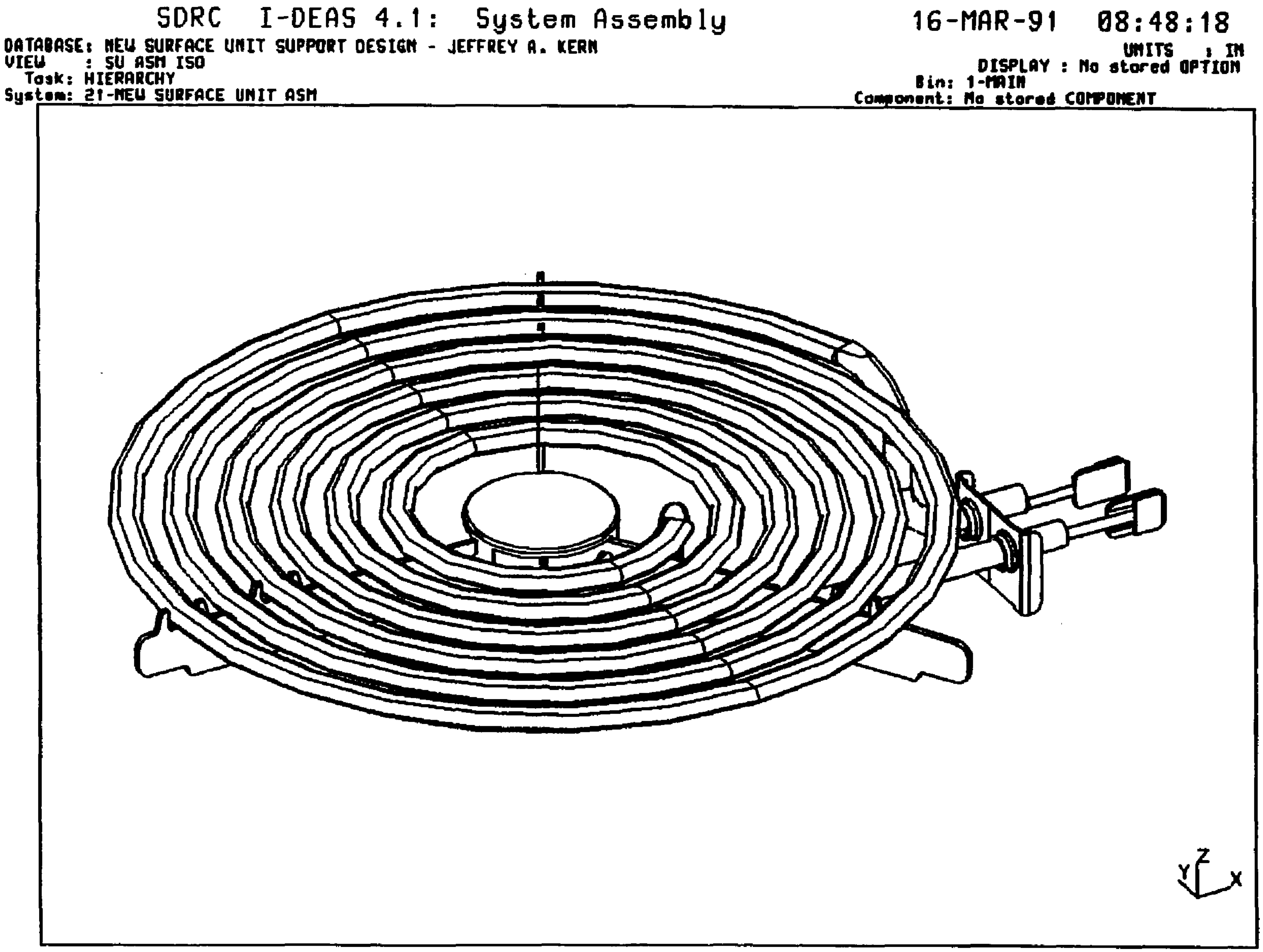
Figure 17. shows the old surface unit assembly that was created with the system assembly hierarchy listed on Table. 4. The new surface unit assembly was constructed with the hierarchy listed in Table 5. It is displayed in Figure 18.

Figure 19. shows an example of an exploded view created with the ORIENT commands. It displays the new surface unit assembly exploded along the Z-axis.



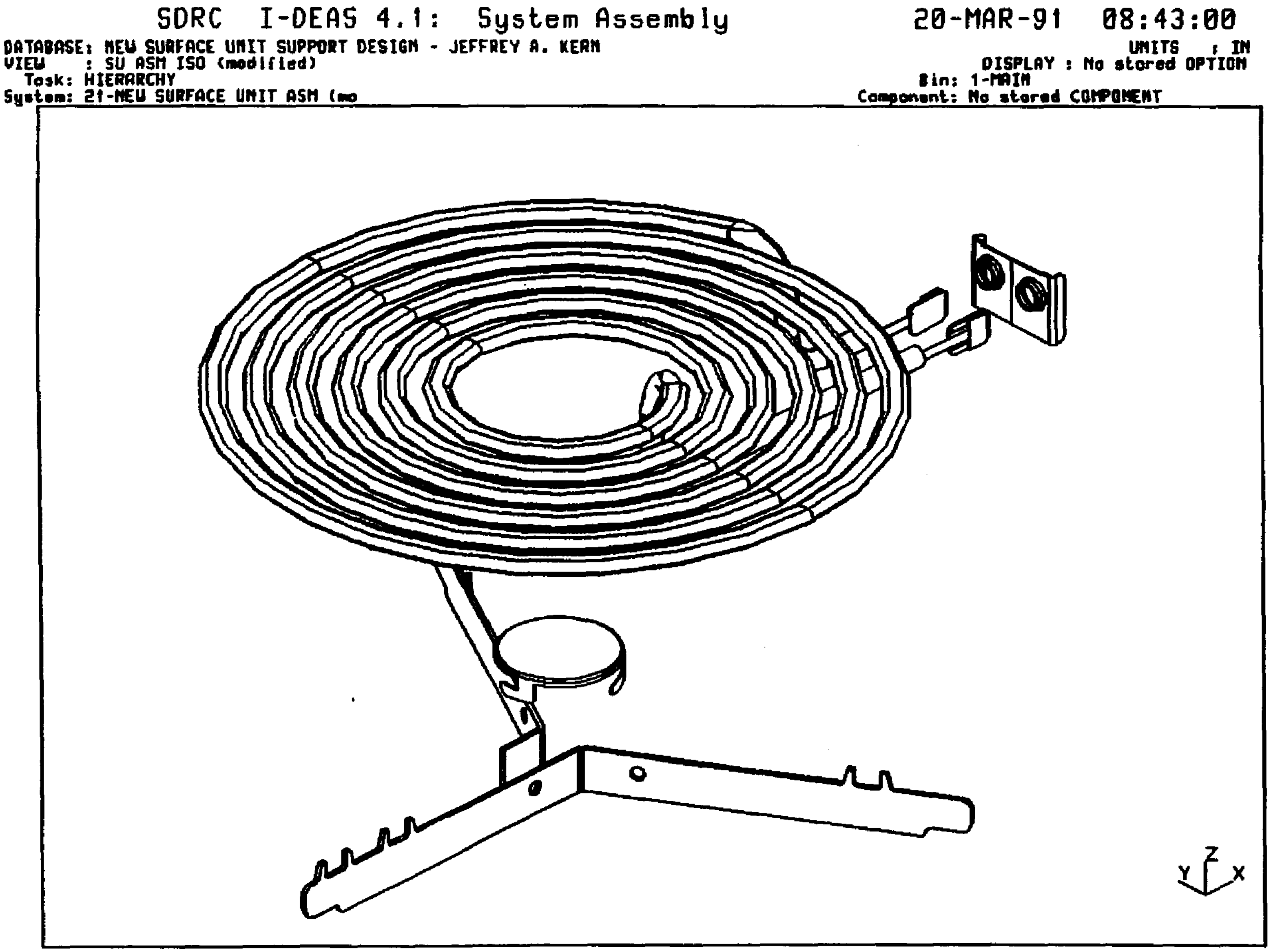
```
        SDRC  I-DEAS 4.1:  System Assembly                   16-MAR-91  08:35:41
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                   UNITS   : IN
VIEW    : SU ASM ISO                                            DISPLAY : No stored OPTION
  Task: HIERARCHY                                                  Bin: 1-MAIN
System: 11-OLD SURFACE UNIT ASM                          Component: No stored COMPONENT
```

FIGURE 17. - SYSTEM ASSEMBLY OF THE OLD SURFACE UNIT ASSEMBLY



```
         SDRC  I-DEAS 4.1:  System Assembly              16-MAR-91  08:48:18
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN              UNITS   : IN
VIEW    : SU ASM ISO                                       DISPLAY : No stored OPTION
  Task: HIERARCHY                                        Bin: 1-MAIN
System: 21-NEW SURFACE UNIT ASM                    Component: No stored COMPONENT
```

FIGURE 18 . - SYSTEM ASSEMBLY OF THE NEW SURFACE UNIT ASSEMBLY



```
SDRC  I-DEAS 4.1:  System Assembly                    20-MAR-91  08:43:00
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN            UNITS    : IN
VIEW     : SU ASM ISO (modified)                            DISPLAY : No stored OPTION
  Task: HIERARCHY                                           Bin: 1-MAIN
System: 21-NEW SURFACE UNIT ASM (mo                         Component: No stored COMPONENT
```

FIGURE 19'. - EXPLODED VIEW OF THE NEW SURFACE UNIT ASSEMBLY

CHAPTER VII

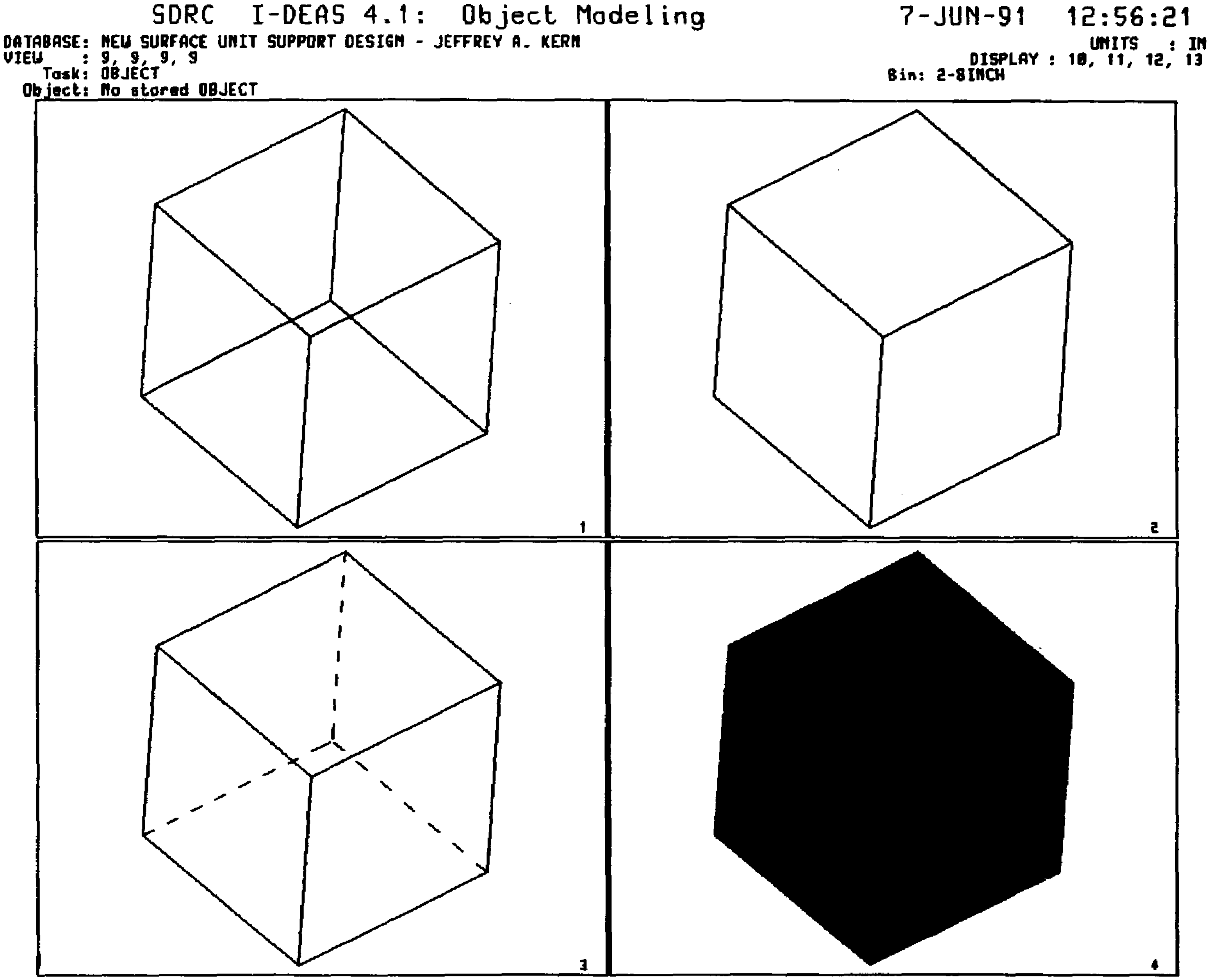
SOLID MODEL DISPLAY

INTRODUCTION

Appendix VII discusses the I-DEAS View commands. With the use of those functions the author was able to customize the display of entities created with the I-DEAS system for viewing on a terminal's graphics display screen. But, I-DEAS offers even greater flexibility in the ways in which entities can be displayed via the commands that are presented in the DISPLAY_OPTIONS and DISPLAY_ATTRIBUTES (or Surface Appearance) menus that are located in each Task of the software.

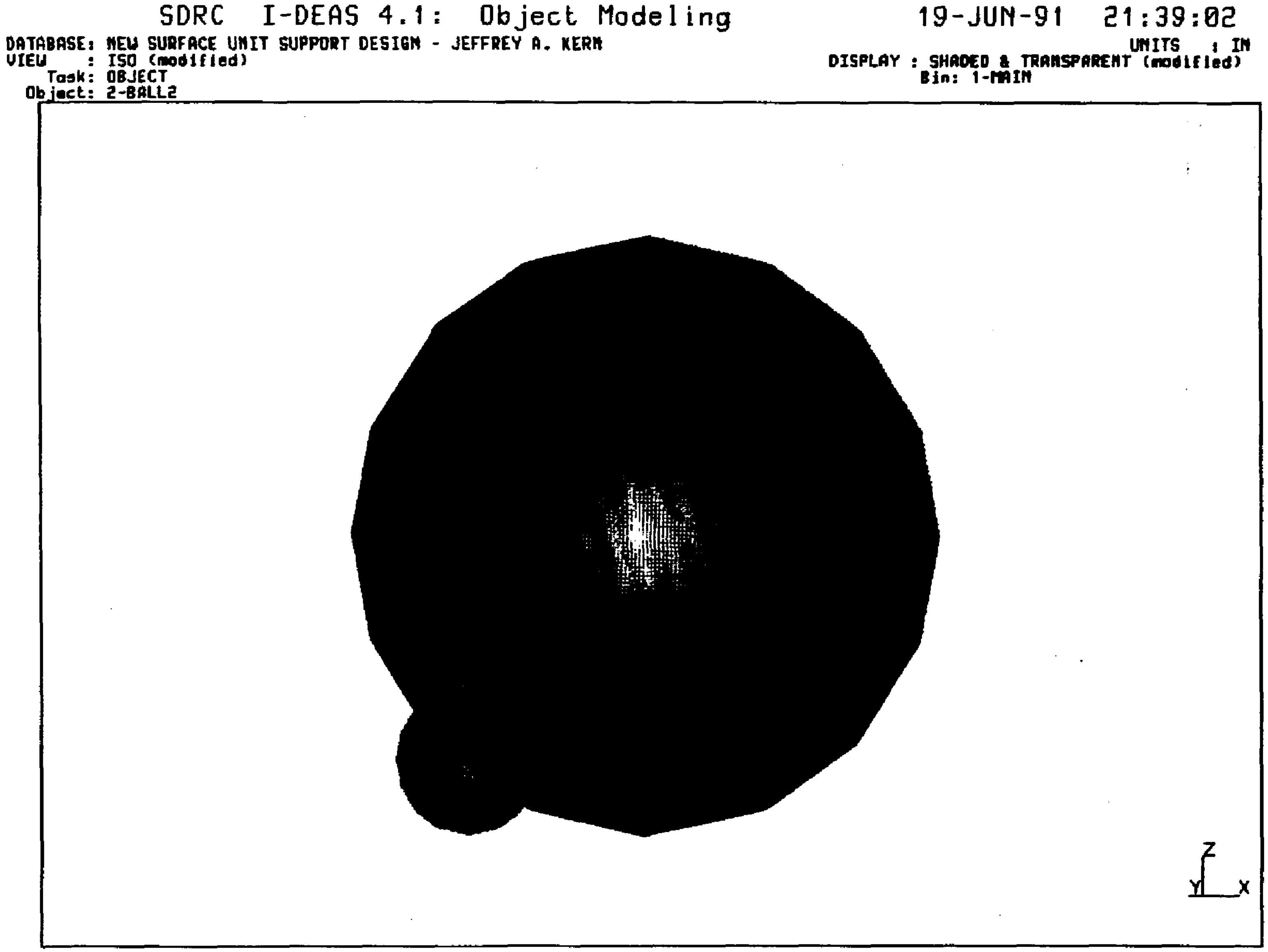
DISPLAY OPTIONS COMMANDS

Appendix X contains discussions covering the display commands that were used for this thesis. There are four major display types that can be produced in I-DEAS, which are called: LINE, HIDDEN LINE, BOUNDARY, and SHADED displays. Each are discussed in Appendix X. Figure 20. shows a cube displayed using one of each of these display types in its four viewports. The most complex of the four display types is shaded images. Among the many options described in Appendix X for shaded images, one of the most interesting the display of solid models that appear to be translucent. Figure 21. shows the use of TRANSLUCENCY.



```
SDRC  I-DEAS 4.1:  Object Modeling                    7-JUN-91  12:56:21
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN        UNITS   : IN
VIEW    : 9, 9, 9, 9                                   DISPLAY : 10, 11, 12, 13
    Task: OBJECT                                     Bin: 2-8INCH
  Object: No stored OBJECT
```

FIGURE 20 . - DISPLAY OPTIONS: LINE, HIDDEN, BOUNDARY & SHADED



```
          SDRC  I-DEAS 4.1:  Object Modeling                 19-JUN-91  21:39:02
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                  UNITS   : IN
VIEW    : ISO (modified)                            DISPLAY : SHADED & TRANSPARENT (modified)
    Task: OBJECT                                              Bin: 1-MAIN
  Object: 2-BALL2
```

FIGURE 21 . - SHADED IMAGES WITH TRANSLUCENCY

CHAPTER VIII

2D CAD DRAWINGS

INTRODUCTION

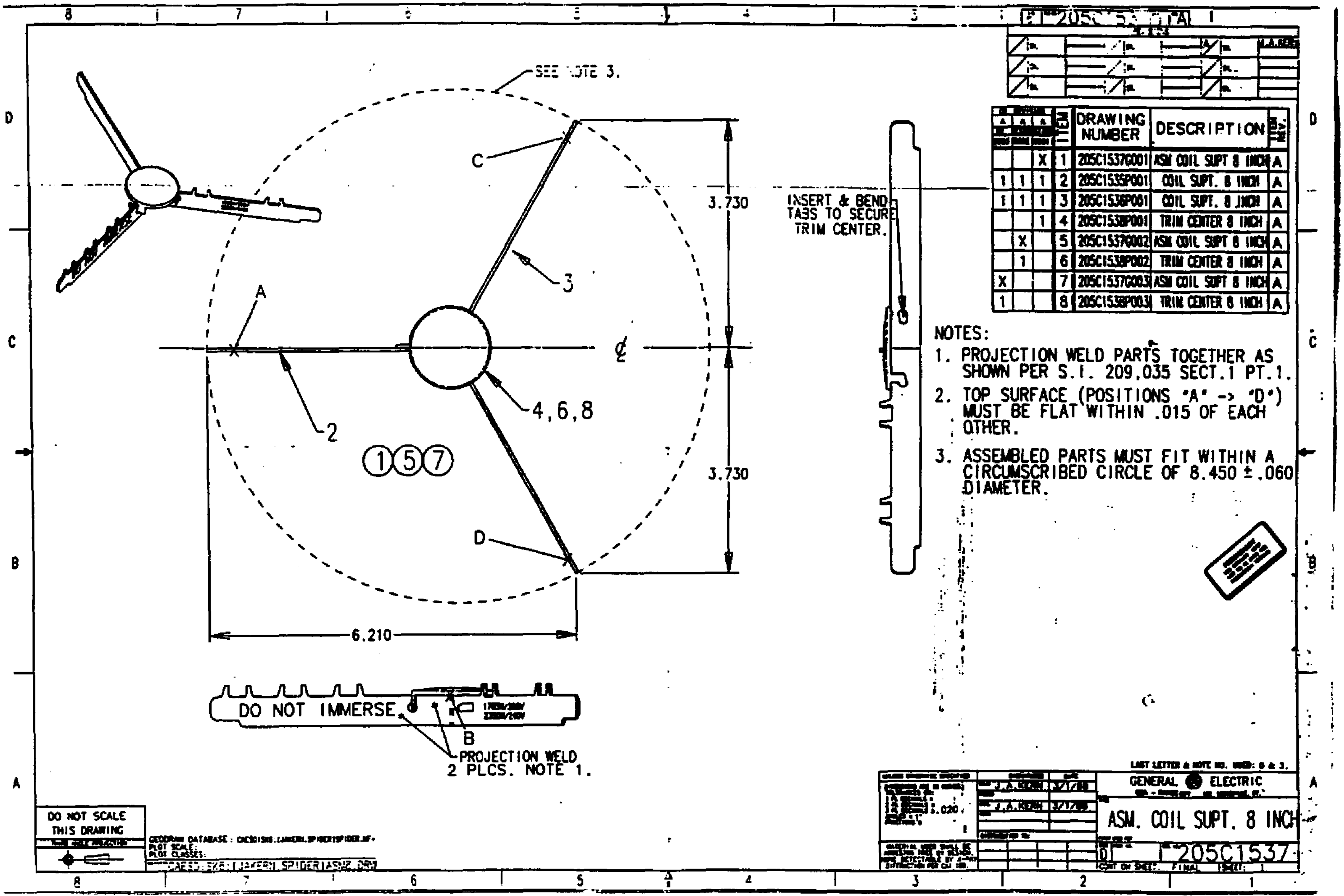
I-DEAS provides a software Family within the main program called DRAFTING. As introduced in CHAPTER 1, DRAFTING can utilize the 3D geometry from solid models for the creation of 2D CAD drawings. It can also be used to create drafting style drawings directly, in the same manner as with any good 2D CAD package.

DRAWING OF THE NEW SURFACE UNIT SUPPORT

In general, at GEA a Designer in the CAD department is assigned to create the production drawings for products and components manufactured by GEA. But, in cases where parts are sourced from outside suppliers, the suppliers are usually responsible for producing the detailed drawings. In these instances, GEA will usually then take the suppliers' drawings, and after they have been approved, they will be duplicated on a GEA drawing format and released for production. These drawings may only display the dimensions that are critical to the parts usage and quality control.

Four drawings were created for this thesis using I-DEAS DRAFTING. This was done because the parts were being submitted to outside suppliers for quotation. This required "sketches" of the two pieces of the support, for the center trims, and for the complete support assembly. Because the parts had been completely designed using the I-DEAS Solid Modeling system, it was decided that it would be easy to quickly detail the solid models on full drawing formats to provided these "sketches". In parallel, the geometry from the solids were transferred to the CAD department (as discussed in the next chapter) for detailing of actual production drawings, so that GEA's Calrod Heater Manufacturing Operation could also prepare quotations for their manufacture within GEA.

Figures 22. through 25. display the I-DEAS DRAFTING drawing or "sketches" that were created.



			ITEM	DRAWING NUMBER	DESCRIPTION	REV
		X	1	205C1537G001	ASM COIL SUPT 8 INCH	A
1	1	1	2	205C1535P001	COIL SUPT. 8 INCH	A
1	1	1	3	205C1536P001	COIL SUPT. 8 INCH	A
		1	4	205C1538P001	TRIM CENTER 8 INCH	A
	X		5	205C1537G002	ASM COIL SUPT 8 INCH	A
	1		6	205C1538P002	TRIM CENTER 8 INCH	A
X			7	205C1537G003	ASM COIL SUPT 8 INCH	A
1			8	205C1538P003	TRIM CENTER 8 INCH	A

NOTES:

1. PROJECTION WELD PARTS TOGETHER AS SHOWN PER S.I. 209,035 SECT.1 PT.1.
2. TOP SURFACE (POSITIONS "A" -> "D") MUST BE FLAT WITHIN .015 OF EACH OTHER.
3. ASSEMBLED PARTS MUST FIT WITHIN A CIRCUMSCRIBED CIRCLE OF 8.450 ± .060 DIAMETER.

FIGURE 22. - I-DEAS DRAFTING SKETCH OF THE NEW SUPPORT ASSEMBLY

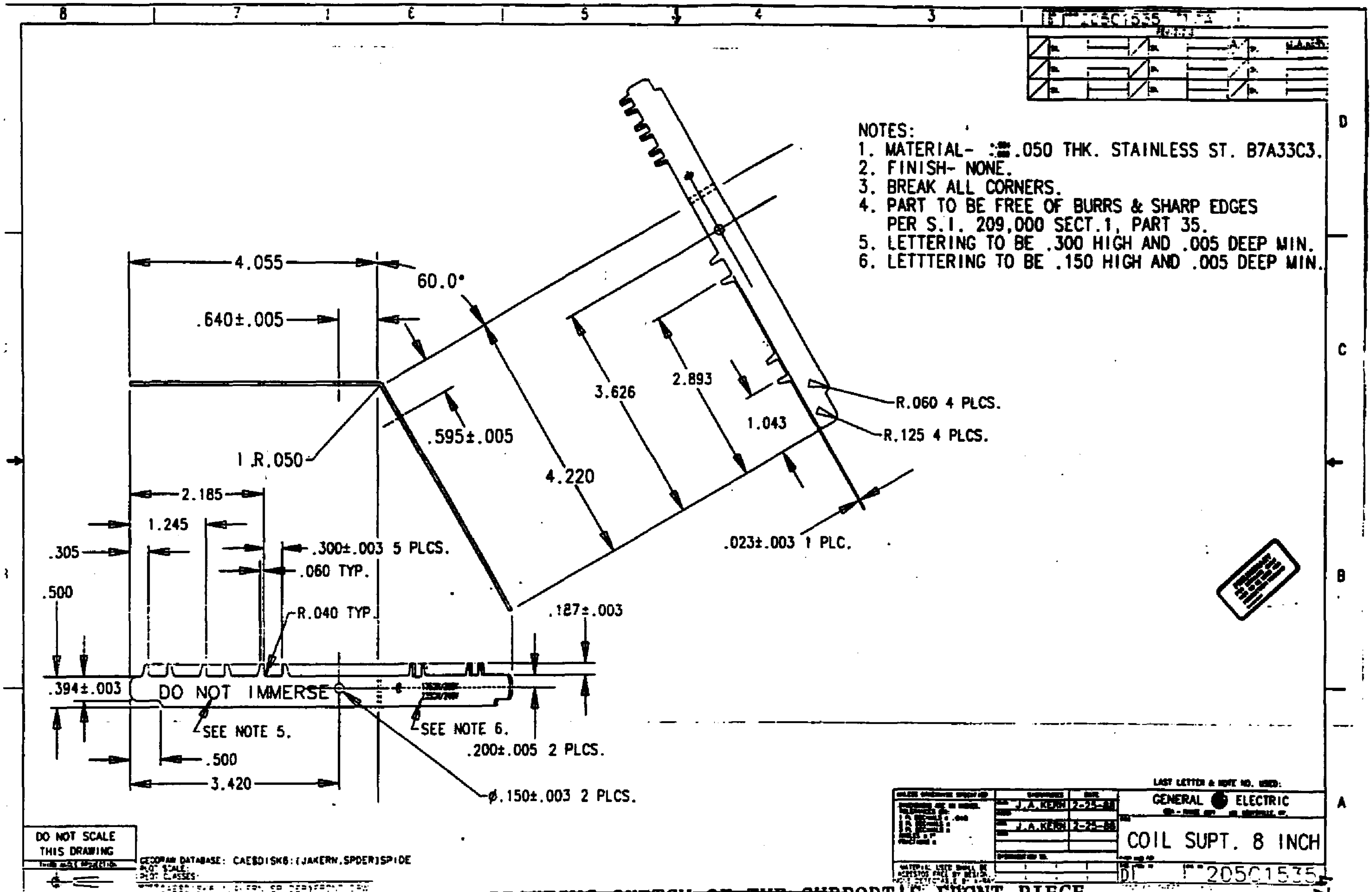


FIGURE 23.. - I-DEAS DRAFTING SKETCH OF THE SUPPORT'S FRONT PIECE

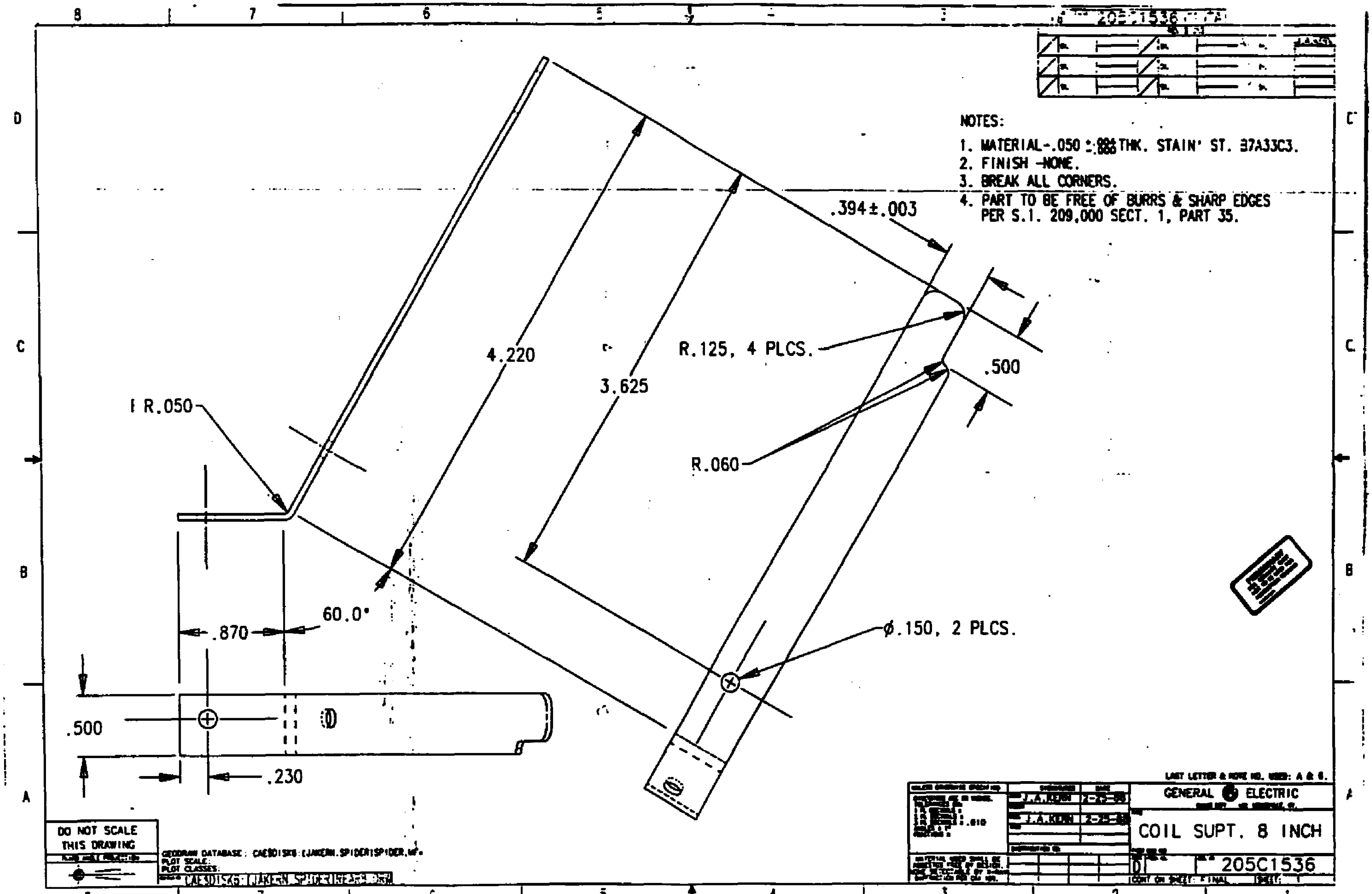

NOTES:
1. MATERIAL-.050 ± THK. STAIN' ST. 37A33C3.
2. FINISH -NONE.
3. BREAK ALL CORNERS.
4. PART TO BE FREE OF BURRS & SHARP EDGES PER S.I. 209,000 SECT. 1, PART 35.
.394±.003
R.125, 4 PLCS.
.500
4.220
3.625
R.050
R.060
60.0°
.870
ϕ.150, 2 PLCS.
.500
.230
DO NOT SCALE THIS DRAWING
GEODRAW DATABASE: CAE$DISK6:[JAKERN.SPIDER]SPIDER.MF
PLOT SCALE:
PLOT CLASSES:
J.A.KERN 2-25-88
J.A.KERN 2-25-88
GENERAL ELECTRIC
COIL SUPT. 8 INCH
205C1536
LAST LETTER & NOTE NO. USED: A & 6.
D
C
B
A
8
7
6
5
4
3
2
1

FIGURE 24 . - I-DEAS DRAFTING SKETCH OF THE SUPPORT'S REAR PIECE

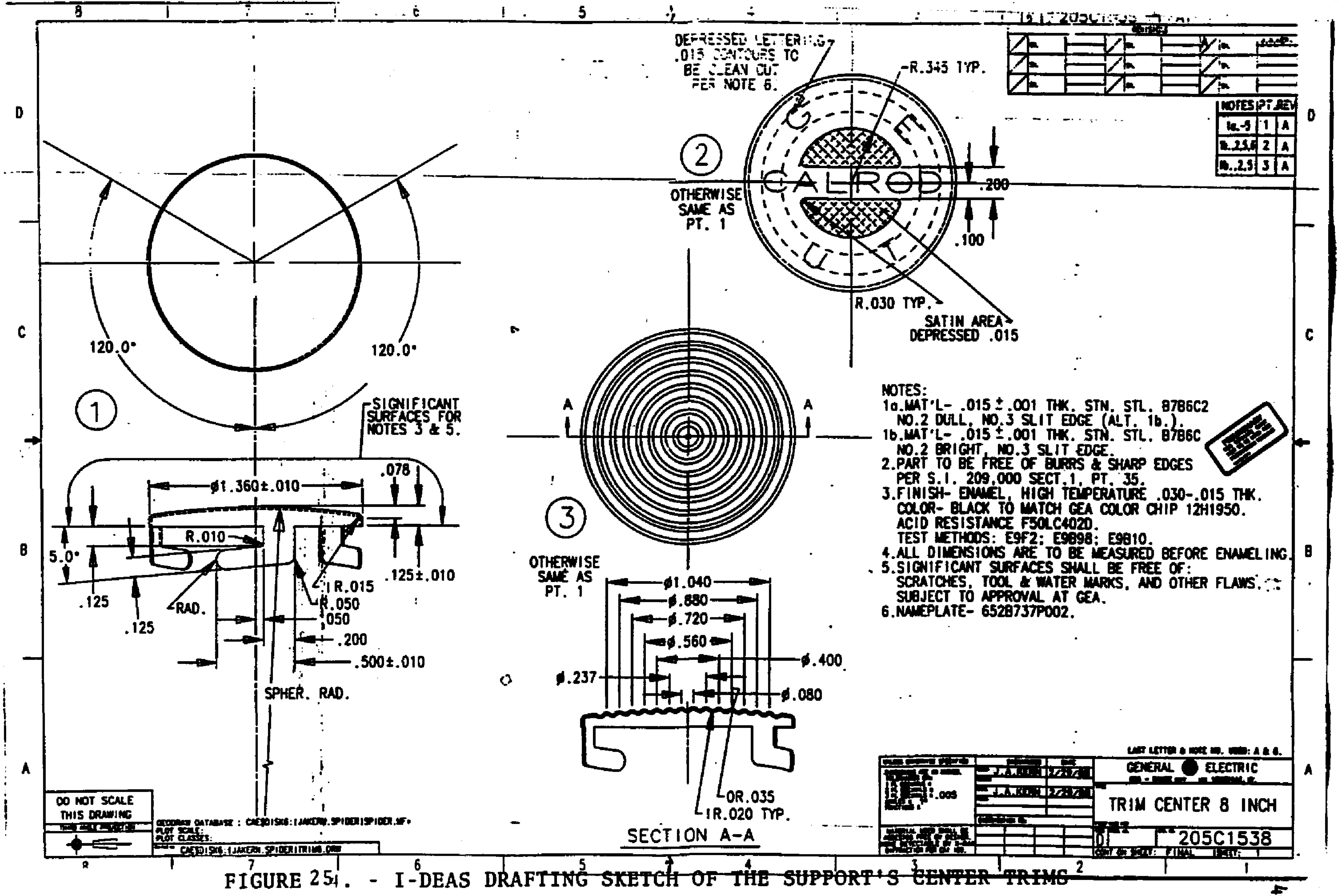

DEPRESSED LETTERING
.015 CONTOURS TO
BE CLEAN CUT
PER NOTE 6.
R.345 TYP.
2
OTHERWISE
SAME AS
PT. 1
CALROD
.200
.100
R.030 TYP.
SATIN AREA
DEPRESSED .015
NOTES PT REV
1a.-5 1 A
1b.,2,5,6 2 A
1b.,2,5 3 A
120.0°
120.0°
1
SIGNIFICANT
SURFACES FOR
NOTES 3 & 5.
ø1.360±.010
.078
5.0°
R.010
.125
.125
RAD.
1R.015
R.050
.050
.200
.500±.010
.125±.010
SPHER. RAD.
A
A
3
OTHERWISE
SAME AS
PT. 1
ø1.040
ø.880
ø.720
ø.560
ø.400
ø.237
ø.080
OR.035
1R.020 TYP.
SECTION A-A
NOTES:
1a.MAT'L- .015 ± .001 THK. STN. STL. B7B6C2
NO.2 DULL, NO.3 SLIT EDGE (ALT. 1b.).
1b.MAT'L- .015 ± .001 THK. STN. STL. B7B6C
NO.2 BRIGHT, NO.3 SLIT EDGE.
2.PART TO BE FREE OF BURRS & SHARP EDGES
PER S.I. 209,000 SECT.1, PT. 35.
3.FINISH- ENAMEL, HIGH TEMPERATURE .030-.015 THK.
COLOR- BLACK TO MATCH GEA COLOR CHIP 12H1950.
ACID RESISTANCE F50LC4020.
TEST METHODS: E9F2; E9B98; E9B10.
4.ALL DIMENSIONS ARE TO BE MEASURED BEFORE ENAMELING.
5.SIGNIFICANT SURFACES SHALL BE FREE OF:
SCRATCHES, TOOL & WATER MARKS, AND OTHER FLAWS.
SUBJECT TO APPROVAL AT GEA.
6.NAMEPLATE- 652B737P002.
DO NOT SCALE
THIS DRAWING
GEOMOD DATABASE : CAE$DISK6:[JAKERN.SPIDER]SPIDER.MF;
PLOT SCALE:
PLOT CLASSES:
CAE$DISK6:[JAKERN.SPIDER]TRIMS.DRW
.005
J.A.KERN
J.A.KERN
GENERAL ELECTRIC
TRIM CENTER 8 INCH
D
205C1538
FINAL
1
D
C
B
A
8
7
6
5
4
3
2
1

FIGURE 25. - I-DEAS DRAFTING SKETCH OF THE SUPPORT'S CENTER TRIMS

CHAPTER IX

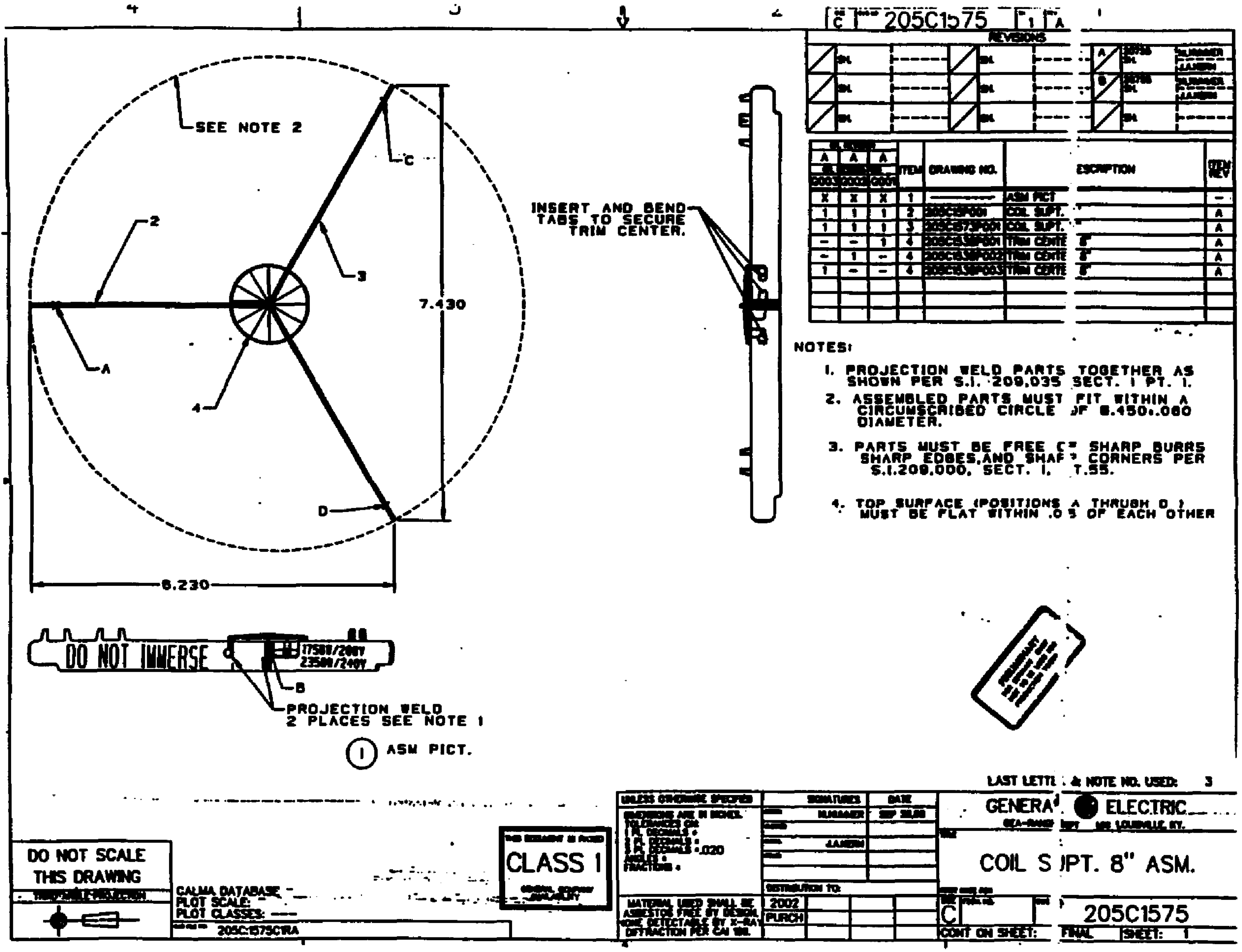
PRODUCTION (CALMA) DRAWINGS

INTRODUCTION

Chapter VIII discussed the preliminary engineering drawings for the new surface unit supports that were produced with I_DEAS DRAFTING. But, it is also possible to transfer solid models' geometry from I-DEAS Solid Modeling to other CAD software packages for detailing. At GEA, production drawings are created using Calma DDM CAD software by the CAD Department's Designers. Therefore, in cases where GEA engineers have designed parts using I-DEAS Solid Modeling, it can be advantageous to transfer the geometry from I-DEAS to Calma. This allows a CAD Designer to bring the engineer's part geometry into Calma drawings. The Designer can then complete any unfinished designing, and then detail production drawings of the parts transferred. The operations required to transfer solid models' geometry to Calma are discussed in Appendix XI.

DRAWING OF THE NEW SURFACE UNIT SUPPORT

The geometry from the new surface unit support's solid models were transferred to Calma. Then GEA'S CAD Department created four detailed production drawings for this work using Calma DDM and transferred geometry. Figures 26. through 29. display these Calma drawings.



A	A	A	ITEM	DRAWING NO.		DESCRIPTION	ITEM REV
G003	G002	G001					
X	X	X	1	————	ASM PICT		–
1	1	1	2	205C15P001	COIL SUPT.		A
1	1	1	3	205C1573P001	COIL SUPT.		A
–	–	1	4	205C1538P001	TRIM CENTE	8"	A
–	1	–	4	205C1538P002	TRIM CENTE	8"	A
1	–	–	4	205C1538P003	TRIM CENTE	8"	A

NOTES:

1. PROJECTION WELD PARTS TOGETHER AS SHOWN PER S.I. 209.035 SECT. 1 PT. 1.
2. ASSEMBLED PARTS MUST FIT WITHIN A CIRCUMSCRIBED CIRCLE OF 8.450±.060 DIAMETER.
3. PARTS MUST BE FREE C* SHARP BURRS SHARP EDGES,AND SHAF ? CORNERS PER S.I.209.000. SECT. I. T.55.
4. TOP SURFACE (POSITIONS A THRUGH D) MUST BE FLAT WITHIN .0 5 OF EACH OTHER

FIGURE 26. - CALMA DRAWING OF THE NEW SUPPORT ASSEMBLY

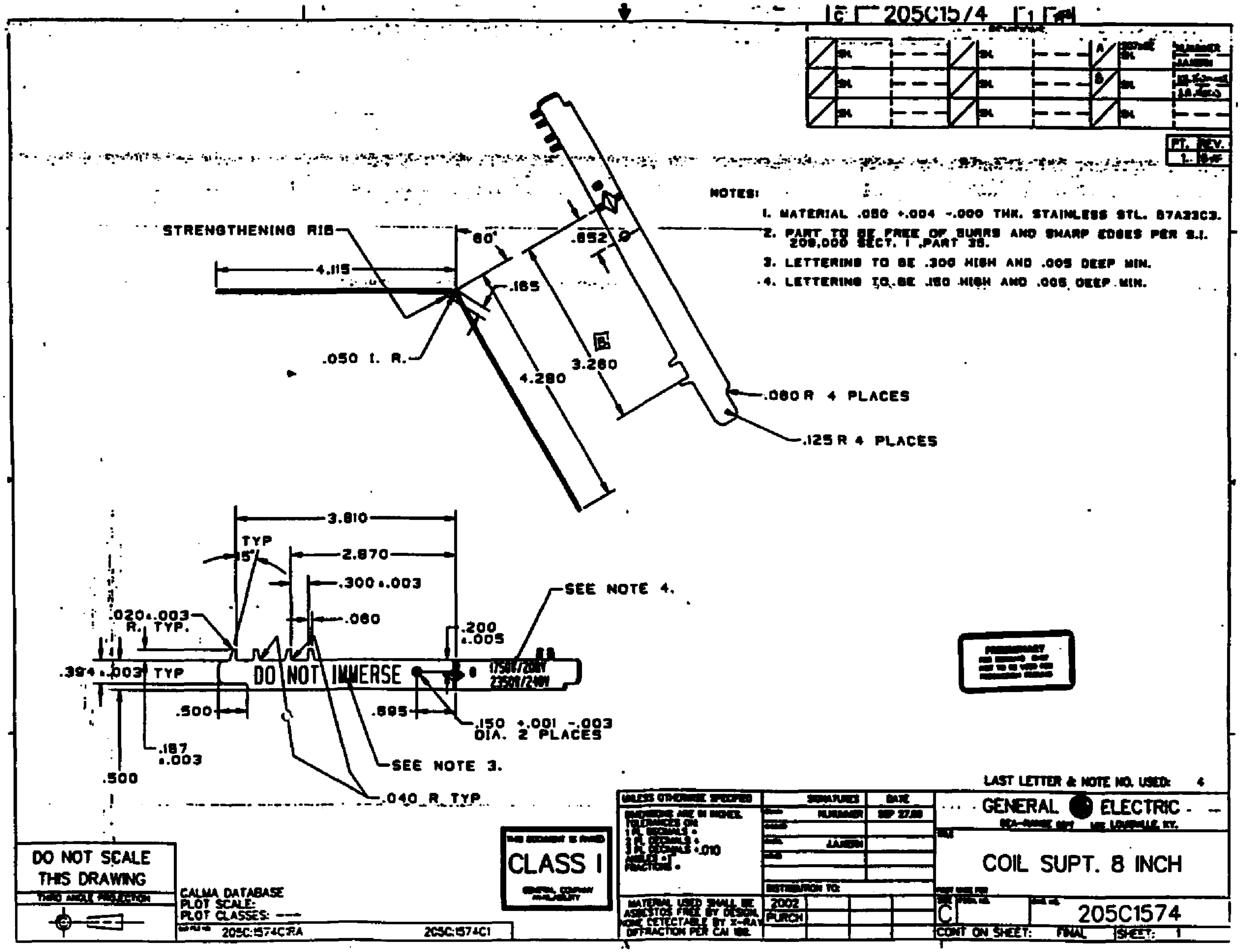


FIGURE 27. - CALMA DRAWING OF THE SUPPORT'S FRONT PIECE

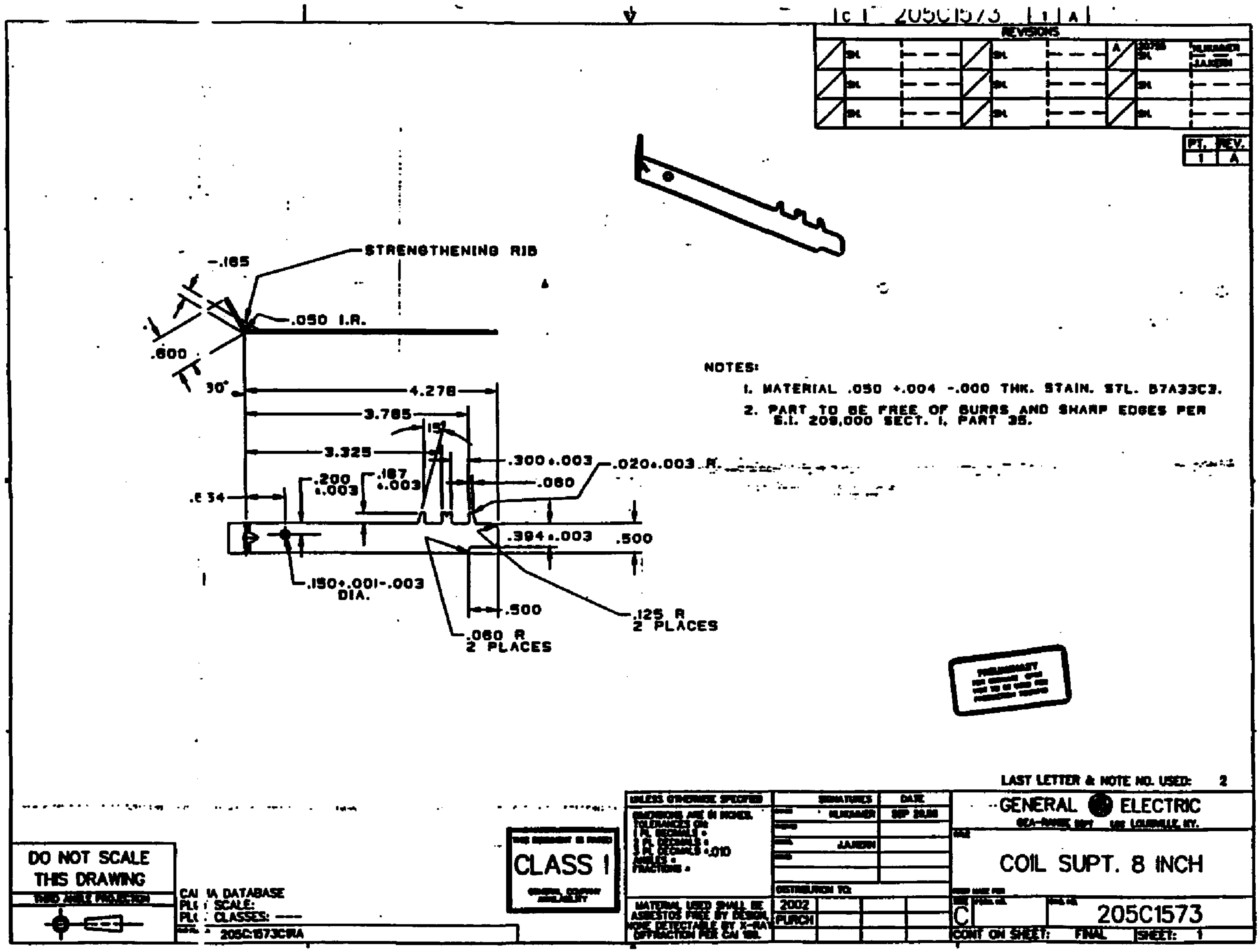


FIGURE 28. - CALMA DRAWING OF THE SUPPORT'S REAR PIECE

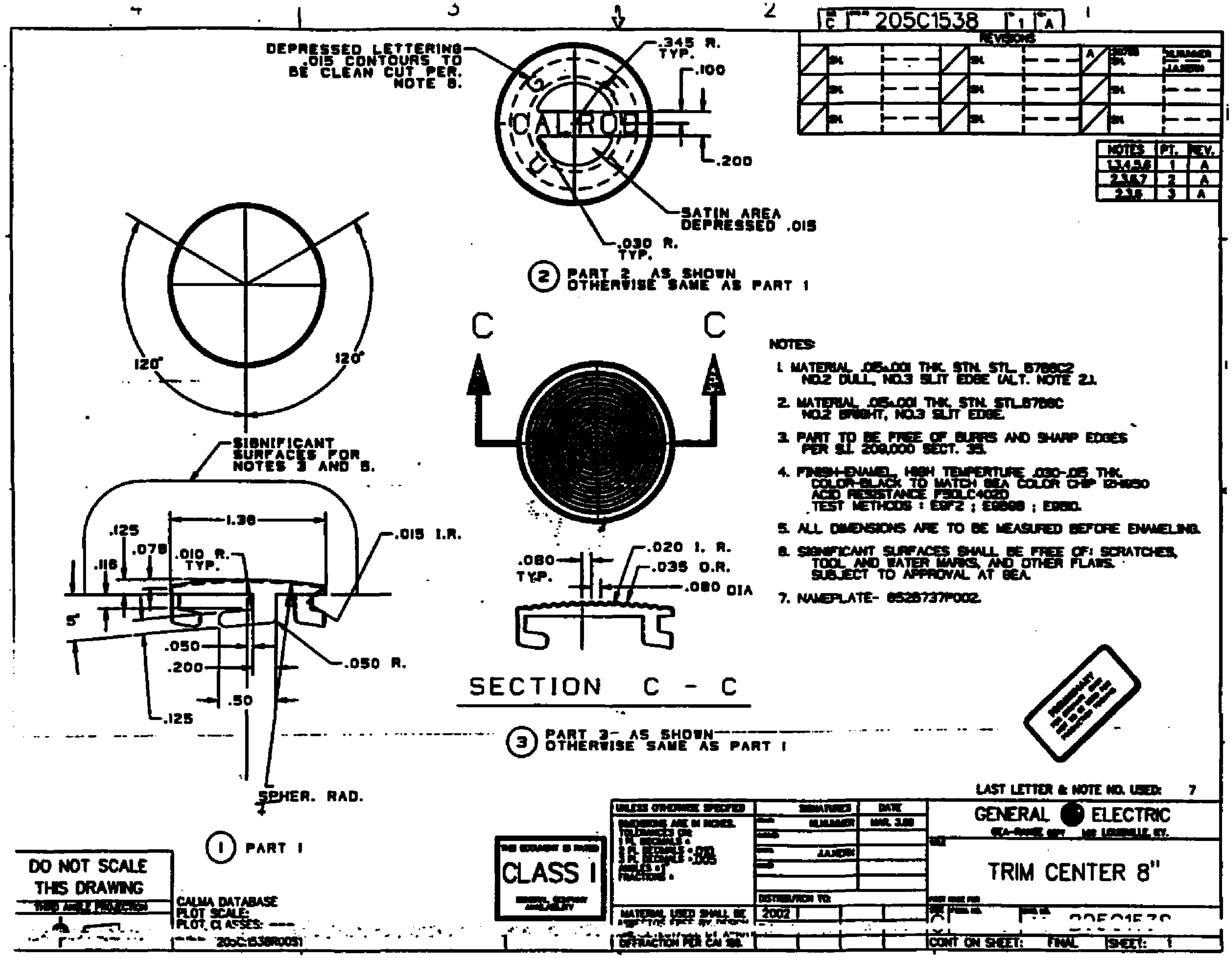

205C1538
REVISIONS
NOTES PT. REV.
1,3,4,5,6 1 A
2,3,5,7 2 A
2,3,6 3 A
DEPRESSED LETTERING .015 CONTOURS TO BE CLEAN CUT PER NOTE 8.
.345 R. TYP.
.100
.200
CALROD
SATIN AREA DEPRESSED .015
.030 R. TYP.
2 PART 2 AS SHOWN OTHERWISE SAME AS PART 1
120°
120°
C
C
SIGNIFICANT SURFACES FOR NOTES 3 AND 6.
.125
1.38
.015 I.R.
.078
.010 R. TYP.
.116
5°
.050
.200
.050 R.
.50
.125
SPHER. RAD.
1 PART 1
.080 TYP.
.020 I. R.
.035 O.R.
.080 DIA
SECTION C - C
3 PART 3 AS SHOWN OTHERWISE SAME AS PART 1
NOTES:
1. MATERIAL .05±.001 THK. STN. STL. B788C2 NO.2 DULL, NO.3 SLIT EDGE (ALT. NOTE 2).
2. MATERIAL .05±.001 THK. STN. STL.B788C NO.2 BRIGHT, NO.3 SLIT EDGE.
3. PART TO BE FREE OF BURRS AND SHARP EDGES PER S.I. 209,000 SECT. 35.
4. FINISH-ENAMEL, HIGH TEMPERTURE .030-.05 THK. COLOR-BLACK TO MATCH GEA COLOR CHIP 12-1050 ACID RESISTANCE P30LC4020 TEST METHODS : E9F2 ; E9808 ; E980.
5. ALL DIMENSIONS ARE TO BE MEASURED BEFORE ENAMELING.
6. SIGNIFICANT SURFACES SHALL BE FREE OF: SCRATCHES, TOOL AND WATER MARKS, AND OTHER FLAWS. SUBJECT TO APPROVAL AT GEA.
7. NAMEPLATE- 8528737P002.
LAST LETTER & NOTE NO. USED: 7
DO NOT SCALE THIS DRAWING
THIRD ANGLE PROJECTION
CALMA DATABASE
PLOT SCALE:
PLOT CLASSES:
205C1538R005
CLASS I
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES.
SIGNATURES
DATE
MAR. 3,88
DISTRIBUTION TO:
2002
GENERAL ELECTRIC
TRIM CENTER 8"
CONT ON SHEET: FINAL
SHEET: 1

FIGURE 29. - CALMA DRAWING OF THE SUPPORT'S CENTER TRIMS

CHAPTER X

I-DEAS FEA PREPROCESSING

INTRODUCTION

The primary purpose for this thesis was to determine the relative strength of the new surface unit support compared to the old support, through the application of finite element analyses. The analyses performed were static loads applied to FEA models of the supports, simulating a standard "load test" which GEA has historically used to compare the relative strengths of surface unit supports.

GEA's standard surface unit load test subjects units to cyclic application of electrical of power, while loaded statically with twenty-five pound (25 lbs) pots of water. The electrical power (260 volts) is cycled 60 minutes on and 20 minutes off, for 1000 on hours with one pot, and then for an additional 1000 hours with a second pot. One pot has a concave bottom and the other has a convex bottom. The static loads are comprised of the pots' and water's weights plus enough lead and steel weights added to yield the total 25 pounds.

The load test was designed to simulate a "life time" of use by an avid "home canner". The use of surface units for canning is the "worst case" application of the units, involving heavy static loads, which is condoned by GEA. The use of surface (and oven) heaters without loads for space heating of a room is worst for the heating coils, and users are warned not to do this in the Use & Care instructions included in the electric ranges sold by GEA.

The pots consumers use in the preparation of canned fruits, meats, and vegetables often have corrugated bottoms which make poor contact with the surface units' heating coils. This creates thermal variations in the unit's heater coil and can cause distortion of the coil. This, combined with the fact that few household cooking utensils have truly flat bottoms, is the reason for the use of concave and convex pots during for the load test. But, since the positions of the new support's "tines" (used to crimp the supports to the coils) are the same as the old support's, this effect was not analyzed. Thus, the final FEA analyses were performed with uniformly loaded flat plates. These plates were coupled to the supports' models with gap elements, which transferred the plate loads to the supports in the same way that utensils' weights are transferred to surface unit supports through the heating coils, which make point contacts with the supports.

Also, the material used for the supports is stainless steel. The alloy's properties do not vary significantly within the range of operating temperatures, and the supports' temperatures are reasonably uniform during this test. Therefore, no thermal loading was modeled.

FEA models were constructed for both support designs using I-DEAS' Engineering Analysis Family. This is discussed in this chapter. Once models were constructed, they were transferred to ANSYS for final preprocessing, and for FEA solutions, as described in Chapter XI. Several finite element mesh sizes were applied to the models under simple static loading to optimize the number of elements used versus the size of the model files, computation times, and accuracy of the results.

This is discussed in Chapter XII. Simple experiments were performed on sample supports of both designs to verify the accuracy of the FEA results, as detailed in Chapter XIII. Finally, FEA was used to simulate the load test, as discussed in Chapter XIV; and the complete projects' results were summarized in Chapter XV.

I-DEAS FINITE ELEMENT ANALYSIS

I-DEAS Preprocessing was used for building FEA models of the new and old supports based on the solid models produced of them in I-DEAS Solid Modeling, Object Modeling Module. Also, I-DEAS Post Processing Module was used for display of the model solutions which were run with ANSYS programs.

The I-DEAS Engineering Analysis finite elements Model Solution Module was not used to produce the solutions to the static analyses performed. ANSYS was used instead, because of its extensive power in solving complex problems, and because ANSYS is the primary FEA tool used by GEA to perform finite element analyses. But, I-DEAS Engineering Analysis Model Solution is capable of solving linear static, dynamic, heat transfer, forced response, and potential flow FEA models. Therefore, although the I-DEAS Model Solutions Module was not used for this thesis, and will not be discussed further, in the absence of ANSYS, or in cases which can be handled with I-DEAS' some what limited element library and solution types, it could be used to solve many FEA models.

TYPE OF FEA MODELS AND ANALYSES TO BE CONSTRUCTED AND PERFORMED

Before the finite element models were constructed for this thesis, it was necessary to decide what type of FEA models and what type of finite elements would be best suited for the analyses that were to be performed. For this thesis, "shell" FEA models were used. Their finite element meshes (planes of nodes and elements) were placed at the center lines of the material thickness, or "web", of each piece of the surface unit supports. Shell models require the application of "area" elements, which are planner, and contain information about the thickness of the elements (i.e., the thickness of the supports).

For this thesis, only static analyses were performed. The specification of the analysis type, done in ANSYS with the "KAN" command, is discussed in Appendix XIII.

Because I-DEAS was used for pre/post processing, the analysis type did not need to be specified in I-DEAS when the models were constructed. But, the analysis type did need to be kept in mind when the element types were determined.

For this thesis, QUADRILATERAL SHELL elements were selected as the element type which were best suited for the analyses to be performed. These were selected because they are AREA 3D elements which can be used to analyze the stresses, strains, and displacements which result from bending of a flat objects caused by in-plane loading. In I-DEAS, in order to specify QUADRILATERAL SHELL elements, the THIN_SHELL element family's, LINEAR element order, QUADRILATERAL element topology was established as the default element type for the surface unit supports.

MESH AREAS

The models created for this thesis were constructed through automatic meshing of FREE_MESH areas.

Some difficulty was encountered with fatal errors in the solutions of the initial models created. These errors were labeled as "POSITIVE PIVOTS". This would generally indicate that the models were improperly constrained, resulting in their attempting to translate or rotate in free body motion through space.

After many changes were tried in attempting to eliminate these errors, wasting countless hours of analysis time, it was discovered that the actual cause for the errors was elements with interior angles that were not close enough to 90°. These where created by the automatic meshing of mesh areas which spanned the entire straight length of each piece of the surface unit supports; including their ends, which have 0.106" steps at the bottom (to rest in the surface unit trim rings on the electric range tops). Once this was determined, the mesh areas were subdivided to create separate mesh areas for the 0.500" and 0.394" wide sections of the supports, and in the areas of the support pieces that are overlapped by other pieces (which is displayed later in this chapter). This forced the automatic mesh generator to create regularly shaped elements, and aided in the merging of "welded" nodes (which is described later in this chapter). This resulted in the elimination of the errors.

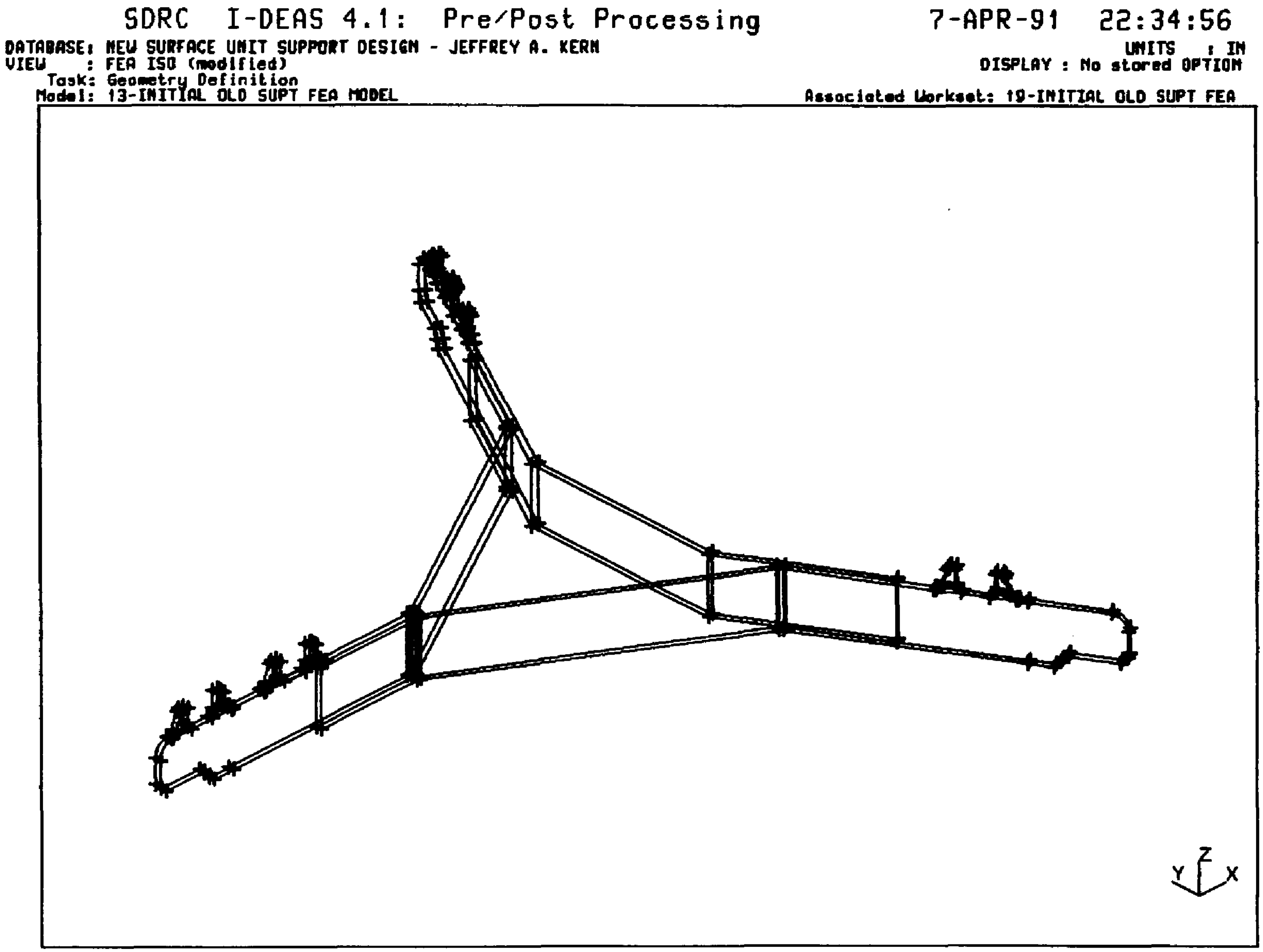
This emphasizes the care that must be exercised in "planning the mesh areas to prevent the creation of overly distorted elements", as mentioned above. I-DEAS automatic mesh generator is very useful for meshing FEA models, but it can't prevent the user from specifying mesh areas which will result in models which can not be properly solved.

GEOMETRY DEFINITION

Since mesh areas are constructed through the selection of closed-loops of geometric curves, it was necessary to define the geometry from which these curves were selected. The geometry of the I-DEAS solid models were too complex for direct use in the construction of FEA models. The solids' corners that were rounded to represent the physical parts, were not rounded in the FEA models. Also, in construction of the shell FEA models, mesh areas were needed on web of the parts assembled in the solid models, rather that on the models' exterior surfaces.

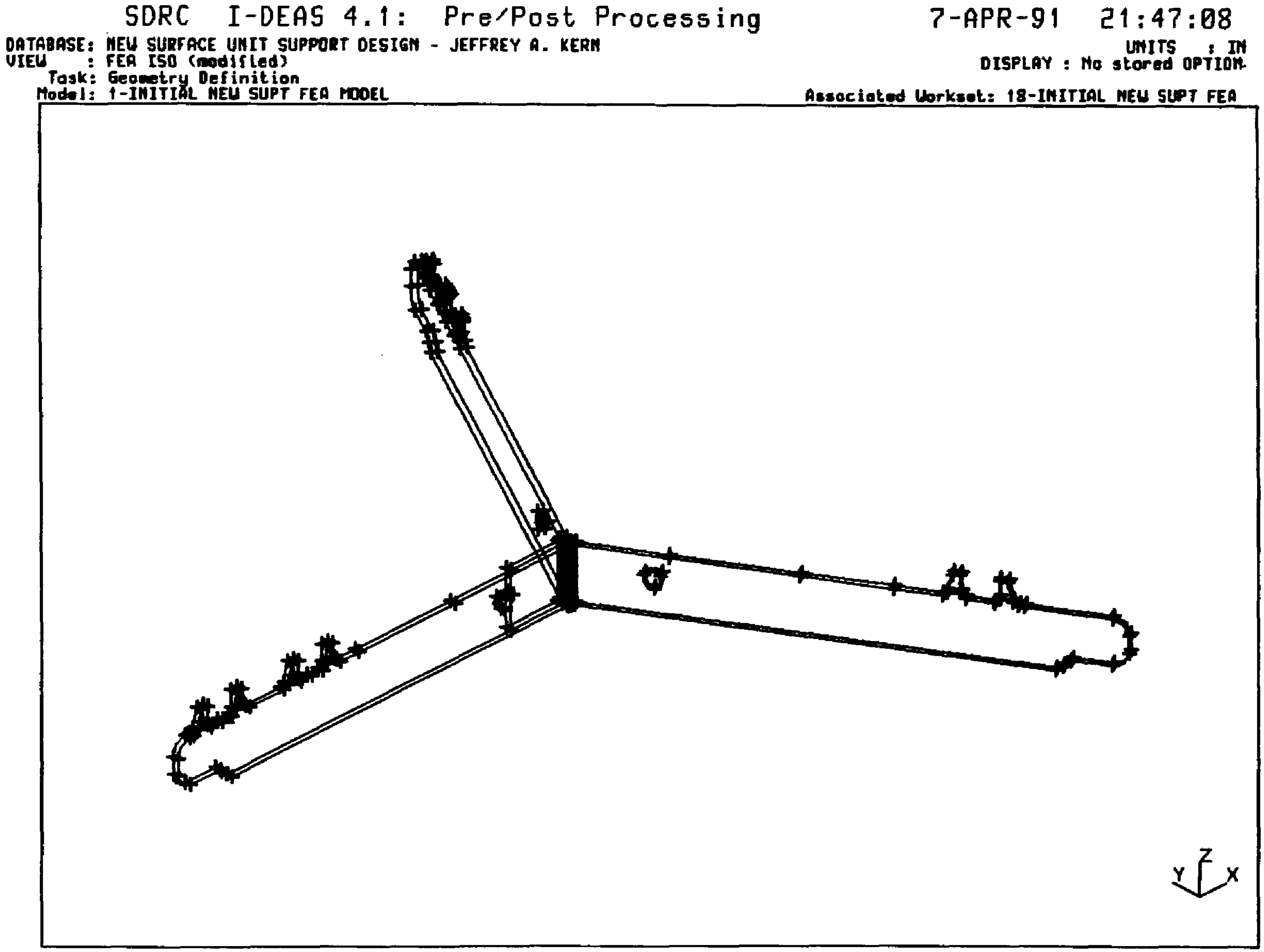
Therefore, two basic working sets were constructed for use in this thesis. One for the new support and one for the old support, including the geometry for a plate for use in loading each. I-DEAS Engineering Analysis' commands were used to create working sets from the solid models that were created of the entire supports (see Chapter V and Appendix XIII). Figures 30. and 31. show the working sets that resulted from these operations.

Once these initial working sets were created, points were created, by interpolation halfway between existing pairs of points, for the end points of the mesh areas needed at the web of each support piece. All the new points created were made a different color than those I-DEAS generated from the solids. In this process, the tines, used to crimp the supports to the heating coils, were eliminated. This was done based on an assumption that the tines would have a negligible effect on the strength of the supports. Similarly, the holes that were designed into the new support and slit ribs that exist in the old support, for the attachment of their center trims, were ignored when theses working sets were refined. Also, the rounded corners were squared off to 90° angles.



```
SDRC  I-DEAS 4.1:  Pre/Post Processing                 7-APR-91  22:34:56
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : FEA ISO (modified)                               DISPLAY : No stored OPTION
   Task: Geometry Definition
  Model: 13-INITIAL OLD SUPT FEA MODEL          Associated Workset: 19-INITIAL OLD SUPT FEA
```

FIGURE 30. - INITIAL OLD SUPPORT'S FEA WORKING SET



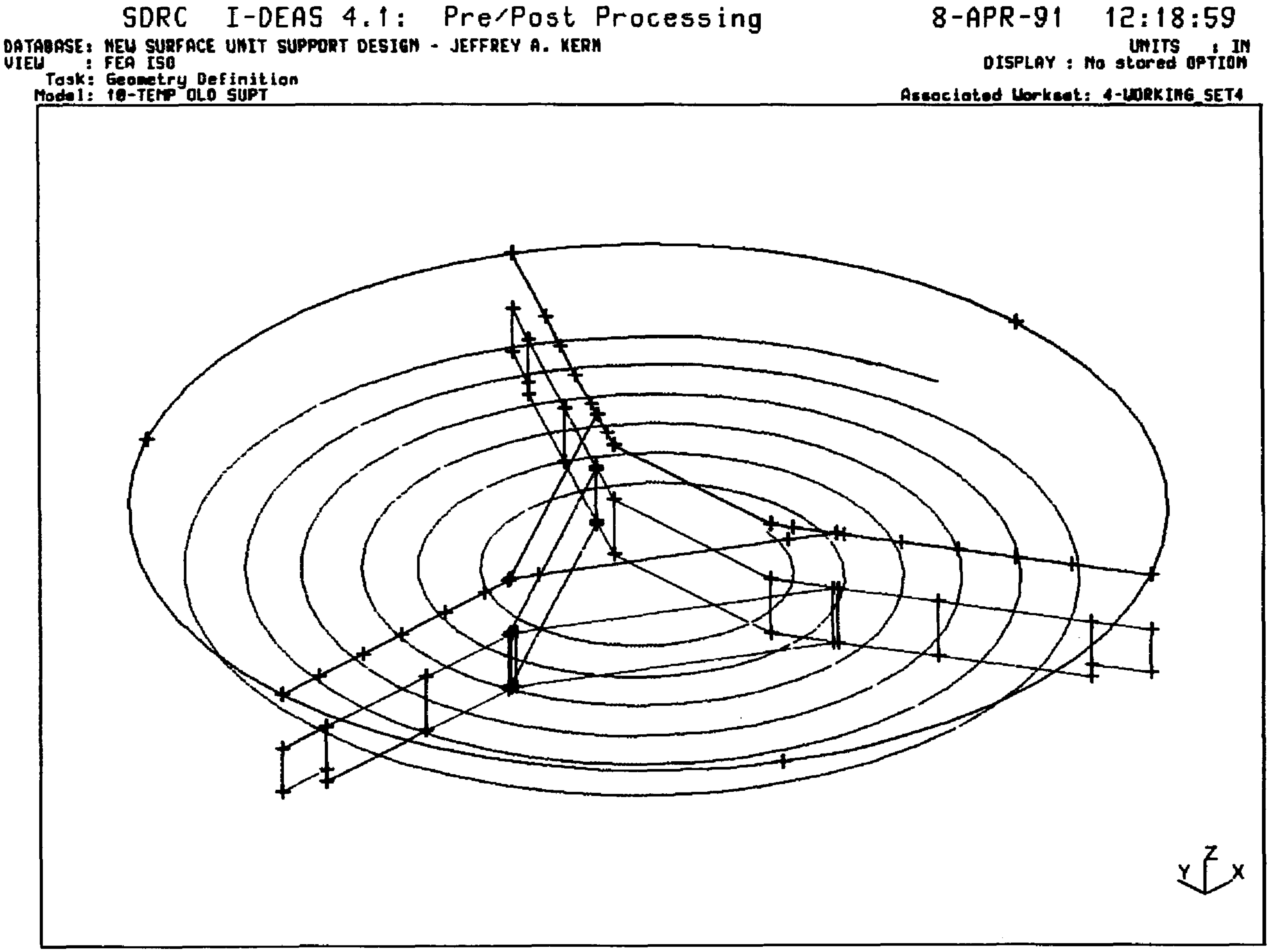
```
SDRC  I-DEAS 4.1:  Pre/Post Processing              7-APR-91  21:47:08
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                UNITS   : IN
VIEW    : FEA ISO (modified)                                      DISPLAY : No stored OPTION
   Task: Geometry Definition
  Model: 1-INITIAL NEW SUPT FEA MODEL                  Associated Workset: 18-INITIAL NEW SUPT FEA
```

FIGURE 31. - INITIAL NEW SUPPORT'S FEA WORKING SET

After the points had been created at the web of each support piece, lines were created connecting them. These lines were eventually selected when the boundaries of the mesh area were created. A different color was used for each piece on the supports (i.e., three colors for the old support and two colors for the new support), to aid in seperation of the individual pieces in later steps in preprocessing of the FEA models.

After the lines were created, the geometry for the web of each support piece was complete. The supports' exterior geometry (from their solid models) was deleted from their working sets. Then, each piece's web geometry was translated so that each piece touches the adjoining the piece it is welded to in the assembly of the supports. This allowed the pieces to be "welded" together in the FEA models by combining overlapping nodes at the weld points, which is describe further, later in this chapter.

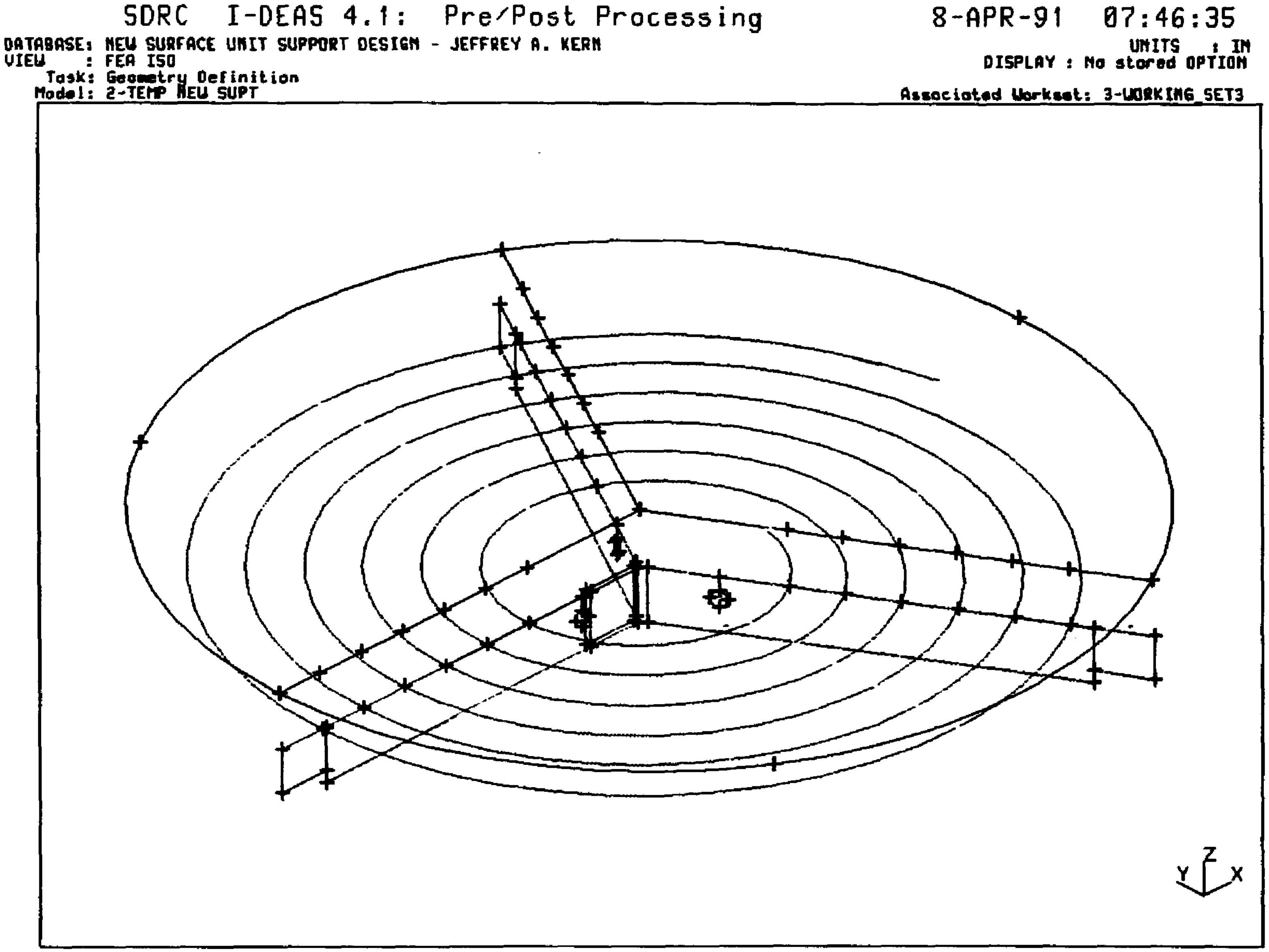
The next item needed in each of these working sets was the plate with which the supports' FEA models were loaded. These plates were constructed by creating points (using a new color) above the ends of the supports, the center of the new support, and intersection of the old support's delta section supports. These points were then connected with lines. Next, the perimeter of the plates was constructed by creating one continuous circle around their perimeters through the three points above the ends of the supports. Then three arcs were created which span between the lines above the supports' legs, and the complete circles were then deleted. This completed the construction of the geometry needed for the supports' and plates' FEA models. These are displayed in Figures 32. and 33.



SDRC I-DEAS 4.1: Pre/Post Processing 8-APR-91 12:18:59

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN UNITS : IN
VIEW : FEA ISO DISPLAY : No stored OPTION
Task: Geometry Definition
Model: 10-TEMP OLD SUPT Associated Workset: 4-WORKING_SET4

FIGURE 32 - TEMPORARY OLD SUPPORT'S FEA WORKING SET



```
      SDRC  I-DEAS 4.1:  Pre/Post Processing              8-APR-91  07:46:35
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN            UNITS   : IN
VIEW    : FEA ISO                                       DISPLAY : No stored OPTION
   Task: Geometry Definition
  Model: 2-TEMP NEW SUPT                          Associated Workset: 3-WORKING_SET3
```

FIGURE 33. - TEMPORARY NEW SUPPORT'S FEA WORKING SET

PHYSICAL AND MATERIAL PROPERTIES TABLES

Before the mesh areas and finite elements could be created, Material and Physical properties tables needed to be constructed for the supports and the loading plates. This involved the specification of the desired properties for the supports' stainless steel, and of a material for the loading plates. It was decided that an aluminum alloy, typically used for household kitchen cooking utensils, and more specifically for the load test's pots, would be used for these analyses.

Both the supports' stainless steel and load plates' aluminum materials were specified as ISOTROPIC. Then two material properties tables were created, one for the supports and one for the load plates. The material properties used for these analyses were taken from the material properties tabulated in the Materials Engineering, Materials Selector 1990.[11]

The surface unit supports are made with a stainless steel alloy. A material properties table was created to store the properties for this material. The CREATE command was used to specify that this material is ISOTROPIC, and it was named "STAINLESS STEEL ALLOY". The properties used were: 28E06 psi, for its modulus of elasticity; and 0.29, for its Poisson's ratio.

The material selected for the load plates was aluminum, alloy 3003. This was chosen based on its typical use for cooking utensils, as listed in the Materials Selector.[12] A material properties table was created to store the properties for this material. The CREATE command was used

[11] Materials Selector, Materials Engineering magazine, Penton Publishing, Inc., December, 1989, p. 48.

[12] Ibid., p. 78.

to specify that this material is ISOTROPIC, and it was named "ALUMINUM ALLOY 3003". The properties used were: 10E06 psi, for its modulus of elasticity; and 0.334, for its Poisson's ratio.

After the materials tables were created, the physical properties tables needed to be created. For the shell elements that were applied in this thesis, the only physical property that the elements required was their thickness. The PHYSICAL_PROPERTIES, CREATE command was used to create two physical properties tables, one of the supports and one for the load plate. The supports' table was named "SU SUPT THICKNESS", and their thickness was specified as 0.050" with the "1st value for thickness". The load plates' table was named "LOAD PLATE THICKNESS", and 0.100" was specified for the "1st value for thickness".

CREATION OF THE MESH AREA

Once the material properties were defined, the MESH_GENERATION Task's commands were used to create the mesh areas at the support pieces' webs. All of the mesh areas for an individual support pieces were created, using a different color for each piece of the supports. This aided in the separation of the support pieces during later stages on the analyses.

When the mesh areas were created, care was taken that the loops of curves selected were always made in a consistent direction of rotation around the mesh areas' boundaries. "Counter clockwise" rotations were used. Although this direction depended on the way the models' geometry was being viewed, it was the consistency that was important. This effects the direction of the positive normal surface vector which the program defines for each element. The normal vector determines which sides of the elements are considered the top and bottom surfaces. This is important for the support models' elements, because it effects the positive and negative signs placed on the stresses and strains that are computed in the solution of the FEA models under load. It also effects the side and direction of the surface loads which were applied to the plates' elements during the simulation of the load tests (Chapter XIV).

Figures 34. and 35. display the mesh areas that were created for the temporary FEA models of the old and new surface unit supports.

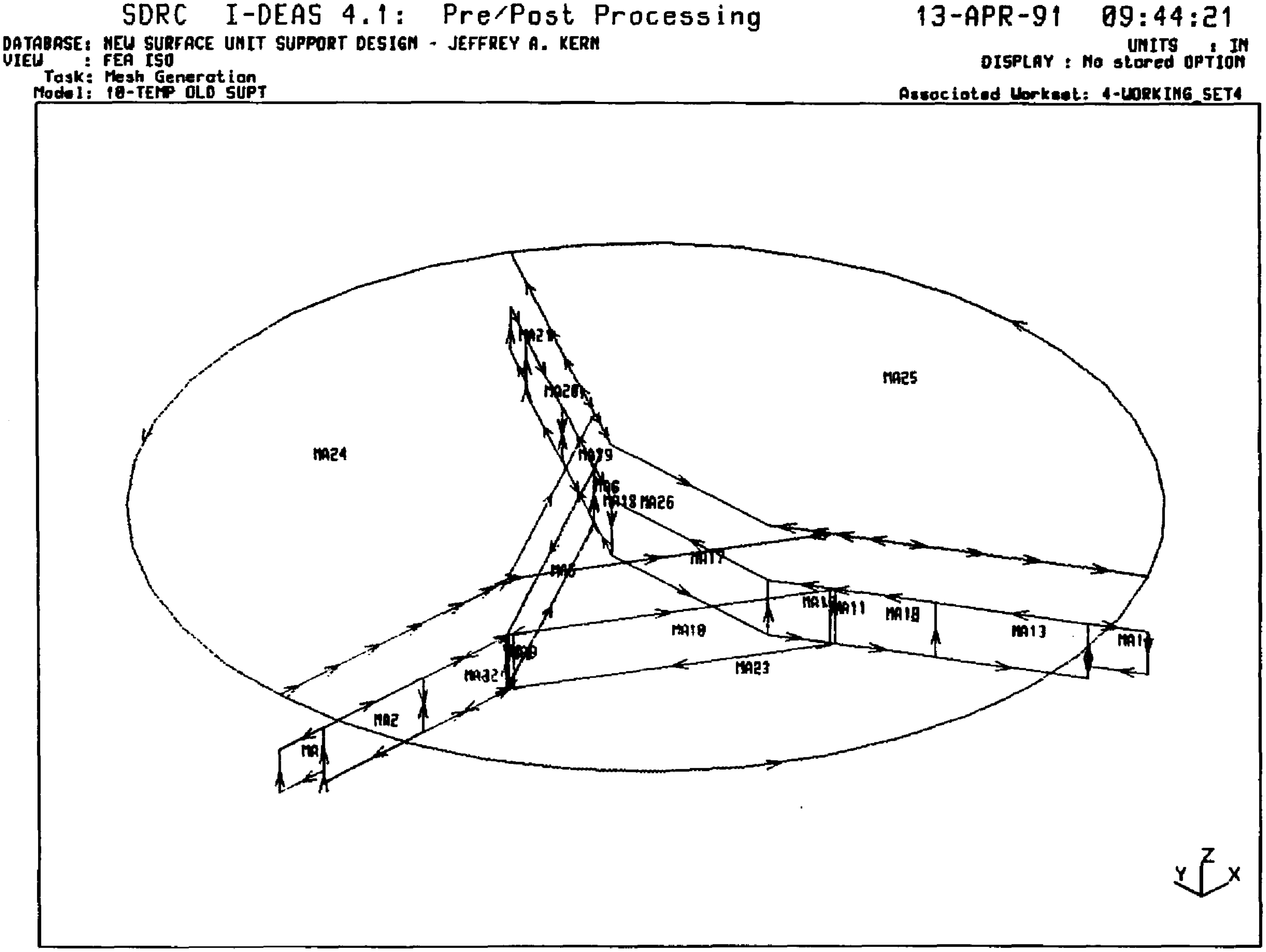
MULTIPLE FEA MODELS

Once the supports' and load plates' mesh areas had been added to the basic FEA models, I-DEAS automatic mesh generator was used to create the nodes and elements on the mesh areas. Several mesh sizes (i.e., several sizes of elements) were applied to each model to optimize the mesh used for the load test simulation. This is discussed in Chapter XII, and the load test simulation is discussed in Chapter XIV. In order to create several models from the same mesh areas, the two temporary models were copied into other (new) models.

In order to be able to distinguish the nodes and elements associ--ated with each of the supports' pieces and the load plates from each other, different colors were used for each piece.

AUTOMATIC MESH GENERATION

Having made all of the preparations for the automatic generation of the nodes and elements, the model's mesh areas were meshed. The mesh areas were selected by color, meshing each piece of the model individually. I-DEAS then meshed the specified mesh areas.



```
        SDRC  I-DEAS 4.1:  Pre/Post Processing            13-APR-91  09:44:21
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN           UNITS   : IN
VIEW    : FEA ISO                                         DISPLAY : No stored OPTION
   Task: Mesh Generation
  Model: 10-TEMP OLD SUPT                       Associated Workset: 4-WORKING SET4
```

FIGURE 34 . - TEMPORARY OLD SUPPORT'S FEA WORKING SET

```
SDRC  I-DEAS 4.1:  Pre/Post Processing          13-APR-91   11:05:40
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS    : IN
VIEW    : FEA ISO                                     DISPLAY : No stored OPTION
    Task: Mesh Generation
   Model: 2-TEMP NEW SUPT                      Associated Workset: 3-WORKING_SET3
```

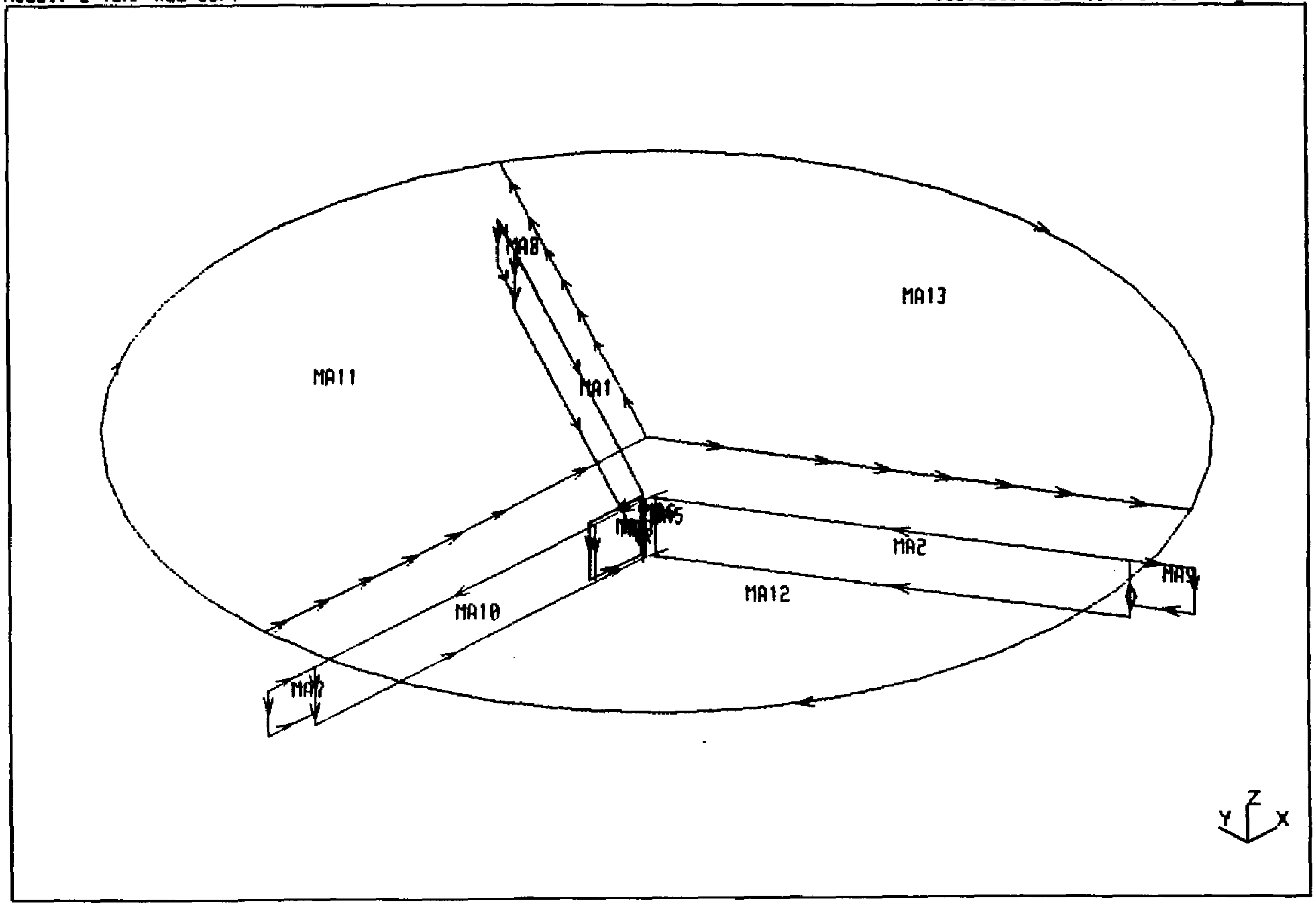


FIGURE 35. - TEMPORARY NEW SUPPORT'S FEA MESH AREAS

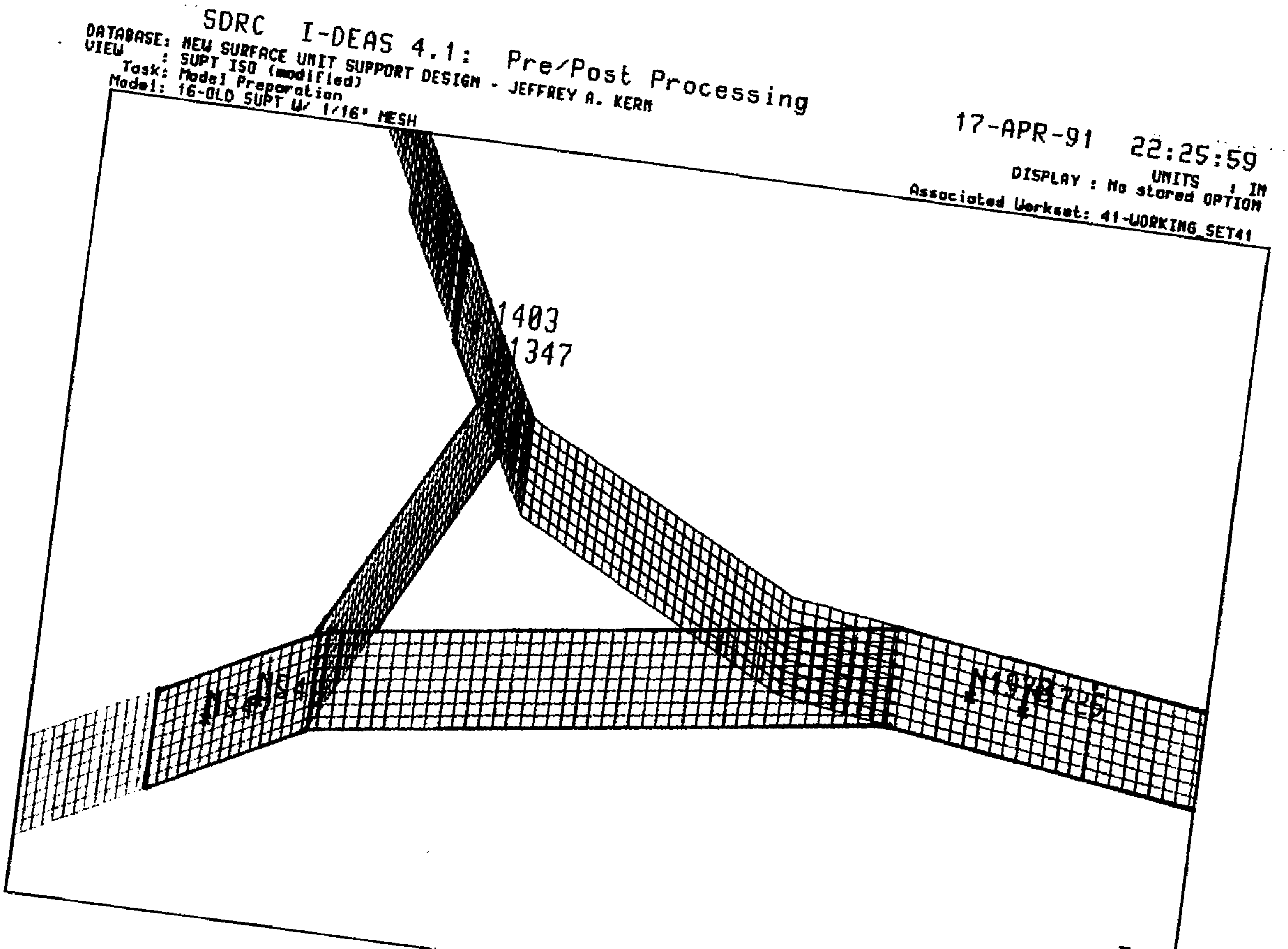
"WELDING" THE SUPPORT PIECES TOGETHER

The shell FEA models of the individual pieces of the surface unit supports could have been left separated by one material thickness (i.e., one half material thickness per piece the each pair of overlapping weld joint). If this was done, pairs of nodes on the joining pieces could have been coupled together. This would have forced nodes at the supports physical welds to translate and rotate together. This was initially tried. In attempting to eliminate the "positive pivot" errors discussed earlier in this chapter, it was thought that some twisting of the loaded models was resulting from this type of weld simulation. An assumption was then made that the shell models of the individual pieces could be moved together (i.e., by 0.050"), and the pairs of nodes at the weld locations could be merged to simulate the welds, with negligible effects on the displacement and stress/strain solutions to the models. This is why the mesh area were moved together. Therefore, at this point in the FEA modeling the "welded" nodes were merged.

Figures 36. and 37. show the nodes that were "welded" for each of the surface unit supports, as described above. The number of welds was identical in each of the duplicated models created of the two supports for the investigation of various mesh sizes. Also, the location of the welds were maintain as closely as the mesh sizes tried would allow.

RESTRAINING THE FEA MODELS

Once the FEA models had been constructed, it was necessary to restrain them so that they could be loaded. The restraints needed to model the manner in which the surface unit supports would be restrained by the trim rings on a range top. These trim rings restrain the surface



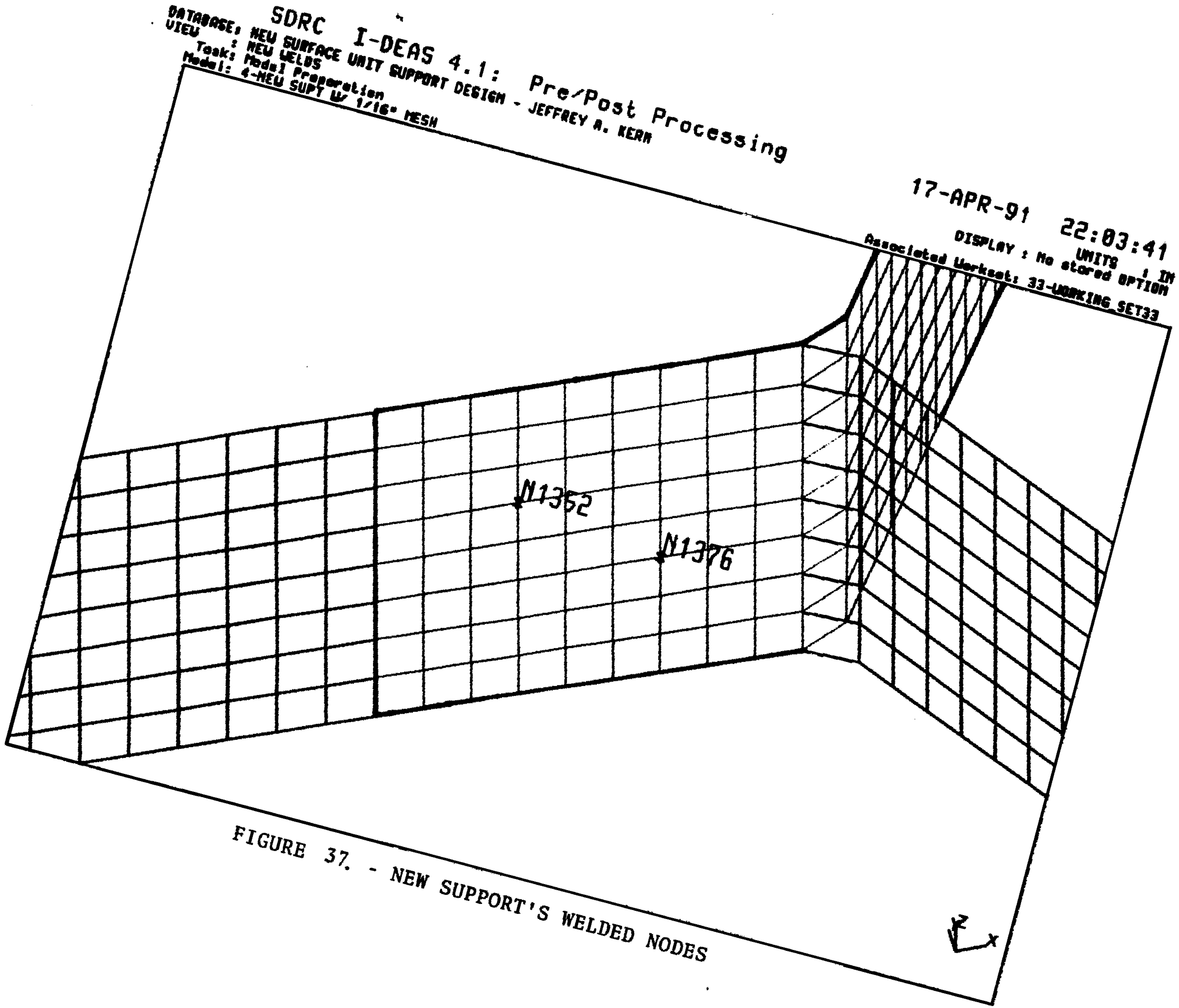
SDRC I-DEAS 4.1: Pre/Post Processing

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW   : SUPT ISO (modified)
   Task: Model Preparation
  Model: 16-OLD SUPT W/ 1/16" MESH

17-APR-91 22:25:59

UNITS   : IN
DISPLAY : No stored OPTION
Associated Workset: 41-WORKING_SET41

FIGURE 36. - OLD SUPPORT'S WELDED NODES



SDRC I-DEAS 4.1: Pre/Post Processing

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW : NEW WELDS
Task: Model Preparation
Model: 4-NEW SUPT W/ 1/16" MESH

17-APR-91 22:03:41
UNITS : IN
DISPLAY : No stored OPTION
Associated Workset: 33-WORKING SET33

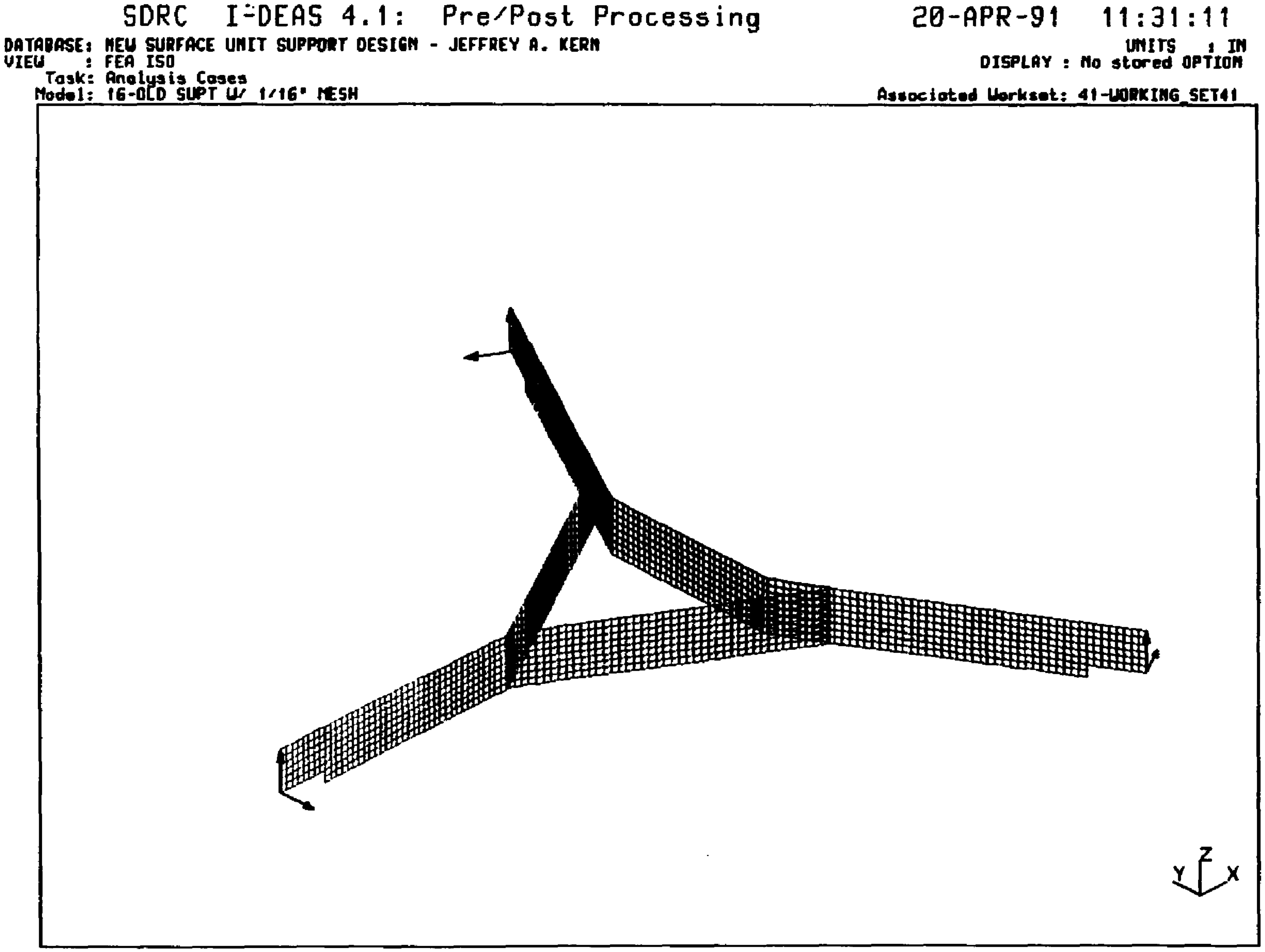
FIGURE 37. - NEW SUPPORT'S WELDED NODES

unit supports from moving in the vertical, Z- direction. They also prevent them from rotating, due to the friction between the supports knife like edges and the trim rings supporting surfaces. Therefore, since the supports were to be loaded in the vertical direction only, and because the trim rings restrain them from rotating, it was decided that the FEA model's nodes' definition and displacement coordinate systems should be converted from Cartesian to Cylindrical systems; allowing the supports' restraints to be created in Global Cylindrical coordinate directions.

After the nodes coordinate systems had been modified, the supports restraints were created. For the surface unit supports, it was decided that they should be restrained at the extreme ends of the supports. In reality, the supports are rounded at the ends, and the trim rings will make more than point contact with the bottom edges of the supports' ends. But, the worst case was modeled, and this was done consistently for both support designs. Therefore, an "END RESTRAINTS" restraint set was created for each of the FEA models of the supports. The restraints in these sets were placed at the support legs' end nodes on their bottom edge. The restraints were nodal displacements with values of zero in the radial and vertical cylindrical coordinate directions, and all other translations and rotations were set free. Figures 38. and 39. display the restrained models.

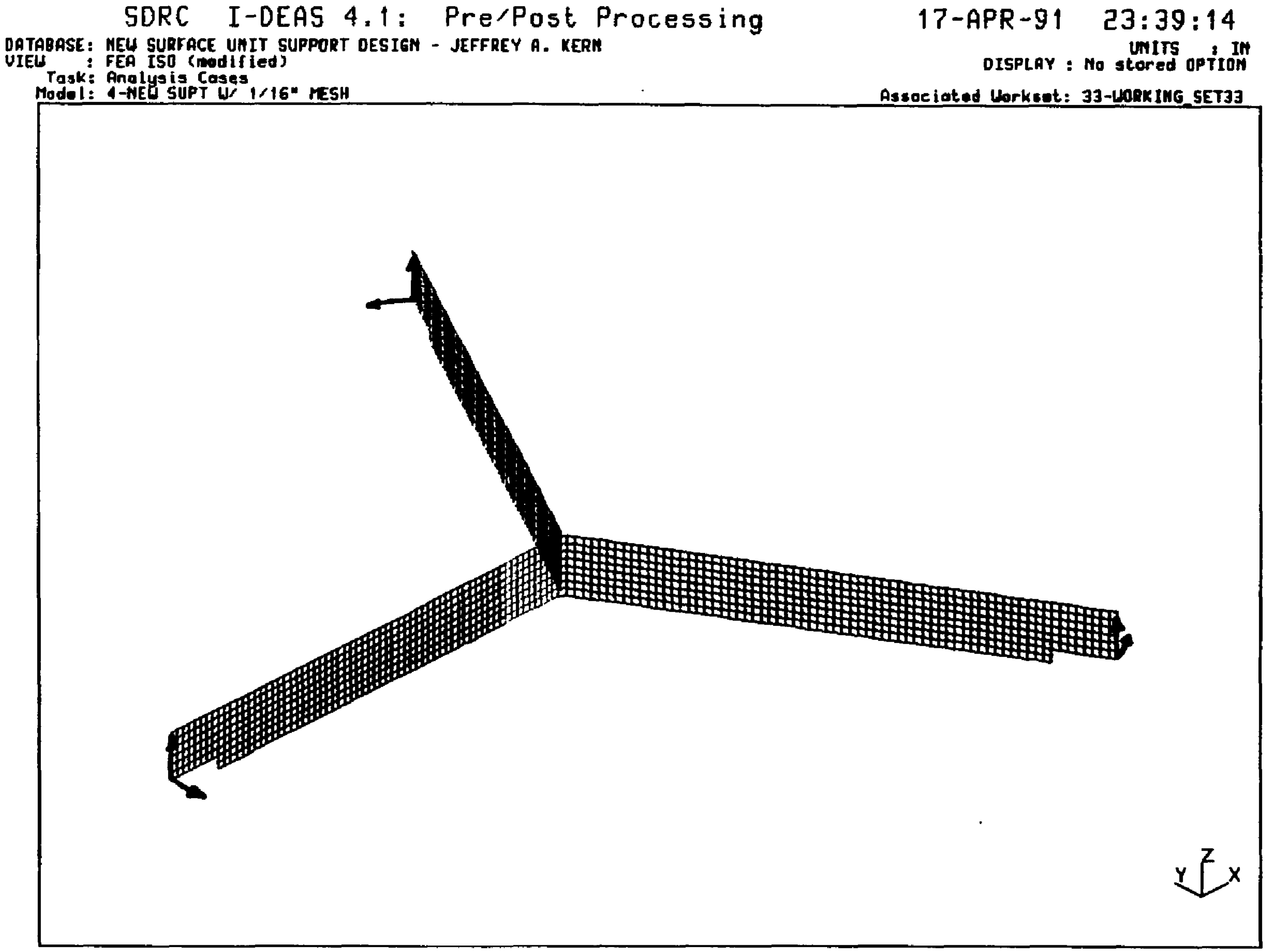
LOADING OF THE FEA MODELS

After the models' were restrained, the next step in the I-DEAS preprocessing of the supports' FEA models was the application of static loads. The support models were loaded with NODAL_FORCEs which totaled



```
      SDRC  I-DEAS 4.1:  Pre/Post Processing            20-APR-91  11:31:11
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN               UNITS   : IN
VIEW    : FEA ISO                                              DISPLAY : No stored OPTION
   Task: Analysis Cases
  Model: 16-OLD SUPT W/ 1/16" MESH                      Associated Workset: 41-WORKING_SET41
```

FIGURE 38 . - OLD SUPPORT'S FEA MODEL RESTRAINED



SDRC I-DEAS 4.1: Pre/Post Processing 17-APR-91 23:39:14

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW : FEA ISO (modified)
Task: Analysis Cases
Model: 4-NEW SUPT W/ 1/16" MESH

UNITS : IN
DISPLAY : No stored OPTION
Associated Workset: 33-WORKING_SET33

FIGURE 39'. - NEW SUPPORT'S FEA MODEL RESTRAINED fifteen pounds (15 lbs) in the negative vertical (Z) direction, for the FEA solution runs that were used to determine what mesh size would be used in the load test simulation. This is discussed in Chapter XII. The support models were loaded via their connection to the load plates with gap elements, for the FEA simulation of the load test. The plates were loaded with elemental FACE_PRESSUREs, which yielded a net twenty-five pound (25 lbs) vertical static load on the supports. This is discussed in Chapter XV.

ANALYSIS CASE SETS

The next step in I-DEAS preprocessing of the FEA models was the creation of analysis case sets. For this thesis, each case set created contained an "END RESTRAINTS" restrain set, and one load set, which was either named "CENTRAL 15# LOAD", "25# PLATE LOAD (.448944575 PSI)", or "25# PLATE LOAD (.3432142 PSI)". The twenty-five pound plate loads required different pressure loads applied to the plates for the new and old supports. This was because the diameters of the plates matched the radius of circles that circumscribed the points at the ends of the supports, and the new support was designed to be slightly larger than the old support, to better fill the trim rings in which they rest on electric range tops.

WAVE FRONT OPTIMIZATION

In an attempt to reduce the computer memory required for the solution of the FEA models created for this thesis, the "wavefront size" was optimized for each model. The was done with I-DEAS' OPTIMIZE_WAVE-FRONT command, which automatically optimizes the model's wavefront size.

To verify that this has been executed properly, the CHECK_WAVEFRONT_SAT command can be executed on each model.

TRANSFER OF THE MODELS TO ANSYS

Once the each FEA models was constructed, a case set was created for each, and its frontwidth was optimized, it was transferred to ANSYS for final preprocessing, and submittal to the ANSYS solution program.

For this thesis, the ANSYS input files were given the names: N112, N116, N120, NPLATEG, O112, O116, O120, and OPLATEG. These names helped to distinguish between the models that were created of the new and old (i.e., either N or O) surface unit support designs. The numbers in these names distinguished them base on their mesh size (e.g., 116 was for the models with a 1/16" mesh size); and the files with PLATEG in their names represented the FEA modes containing the load plate models coupled to the supports' models with gap elements.

Having written out ANSYS input files to the VAX computer disk directory, this work was ready for the next step, which was file editing and ANSYS preprocessing. This is discussed in the next chapter.

CHAPTER XI

ANSYS FEA PREPROCESSING

INTRODUCTION

The proceeding chapter discussed the preprocessing work that was done in I-DEAS to produce FEA models of the new and old surface unit supports. At the end of Chapter X the models had been written out from I-DEAS as "ANSYS PREP7 input files". In this chapter, ANSYS preprocessing is discussed, and the steps taken to make use of these files is described. A detailed discussion on the use of ANSYS preprocessing operations that were used is provided in Appendix XIII.

PREPROCESSING

For this thesis, the FEA models had already been constructed, and preprocessed in I-DEAS Engineering Analysis; and at this stage in the analyses, they could have been transferred into ANSYS for final PREProcessing and solution. Before this was done, all of the ANSYS preprocessing procedures and commands were reviewed (see Appendix XIII) to determine what needed to be present in the transfer files written by I-DEAS. This was done so that they could be edited before being brought into ANSYS, to insure that the desired results would be accomplished.

TRANSFER OF PREP7 INPUT FILES FROM I-DEAS

After the input files were edited to make the changes needed to have then conform to the steps outlined in Appendix XIII, they were ready to be transferred into ANSYS. This was done in batch mode, with the aid of a command file as discussed in Appendix XIII.

FINAL ANSYS PREPROCESSING

Once the models had been transferred into ANSYS, they were interrogated to be sure that they had been received as desired. This was accomplished with use of ANSYS' entity listing, graphical display and plotting commands. Once the author was sure that they models were acceptable, ANSYS Analysis Files were written out for batch solution of the FEA models. See Appendix XIII for details on the operations performed.

BATCH SOLUTION OF THE ANSYS ANALYSIS FILES

The analysis files were submitted to ANSYS' solution routine in the batch mode. When the ANSYS analysis solution is complete, the results are available in ANSYS FILE12.DAT (or MODEL_NAME.12 when the batch command file is used). Once this file was obtained for each of the models solved, the results were post processed in ANSYS and then transferred into I-DEAS for final postprocessing and graphical display of the results. This is discussed in the next chapter.

CHAPTER XII

POST PROCESSING OF THE FEA RESULTS

INTRODUCTION

When the ANSYS FEA models had been solved, ANSYS preprocessing was used to initially investigate the results. After the results had been reviewed to be sure they looked appropriate, the data were transferred back into I-DEAS for final post processing and display of the results. This last step was done because of the superior graphics display options presented by I-DEAS.

ANSYS POST PROCESSING OF THE SOLUTION RESULTS

Once the analysis files had been solved, and the results were available in ANSYS FILE12 (i.e., MODEL_NAME.F12). ANSYS' post processing routine was used to review and display the results graphically. ANSYS' "POST1" post processor was used to list and graphically display the analyses results.

FAILURE CRITERION FOR THIS THESIS

For this thesis the failure criterion [13] applied to determining if the new surface unit support would perform acceptably in comparison to

[13] Arthur P. Boresi, Omar M. Sidebottom, Advanced Mechanics of Materials, Fourth Edition, John Wiley & Sons, NY, 1985, pp. 97 - 104.

the old support design was "Excessive Yielding at Ordinary Temperatures", with the yield criterion of the "maximum distribution energy theory". This required the examination of the maximum Von Mesus stresses resulting from the loading of the two surface unit support designs.

RESULTS REVIEWED

For this thesis, the nodes' maximum vertical displacements, and the elements' maximum component, principal, and Von Mesus stresses where desired, to determine which mesh size was best suited for these analyses, and to determine if the new support's strength is acceptably in comparison to the old support design. In each case, the data was sorted to place the maximum values of the desired displacements and stress at the top of each listing. The stress data that were reviewed were the shell elements' middle (average) stresses.

Although excessive yielding was the criterion used to determine if the new support design was acceptable, it was also important to examine the maximum deflection of the new support in comparison to the old support. If surface unit supports defect too much, even though they might not yield excessively, this can lead to poor contact with cooking utensils, which can effect heating performance. When ANSYS reads the analysis results into memory for post processing, the nodal displacements are automatically read. So, no declaration needed to be made prior to initiating ANSYS' reading of the results.

In addition to printing out the vertical displacements and stresses, the reaction forces were also printed out to verify that their totals were equal to the loads applied in each analysis case solved.

I-DEAS POST PROCESSING OF THE SOLUTION RESULTS

In addition to using ANSYS to process the analysis results, I-DEAS post processing was also applied. This required the results to be transferred back into I-DEAS. But, this allowed I-DEAS' superb color graphics display capabilities to be used. This required the use of I-DEAS Data Loader software, as discussed in Appendix XIV.

ANALYSES' RESULTS

The ANSYS and I-DEAS post processing procedures outlined in this chapter (see Appendix XIV for detailed operations performed) were used for this thesis. The plots and results reports produced are displayed and discussed in the next three chapters.

CHAPTER XIII

FEA MESH SIZE OPTIMIZATION

INTRODUCTION

As was introduced in Chapter X, the FEA models of the old and new surface unit supports were loaded with a static fifteen pound force, with various mesh sizes applied to the models in order to determine the best mesh size to use when simulating the surface unit load test. The models were prepared in I-DEAS as discussed in Chapter 10. Then they were transferred to ANSYS for final preprocessing and solution, as described in Chapter XI. The results were then post processed with ANSYS, as described in Chapter XII.

MESH SIZES TRIAL LOADED

The supports are 1/2 inch wide. It was desirable to choose element sizes that would span this width with a whole number of elements, in order to use automatic mesh generation to produce good square elements. Therefore, the new surface unit support's basic FEA model (mesh area model), was meshed with elements of the following sizes: 1/8", 1/12", 1/16", and 1/20".

Once the four meshed models of the new support had been constructed, each was loaded with a fifteen pound (15 lbs) static load, under the end constant conditions discussed in Chapter X. Figure 40.

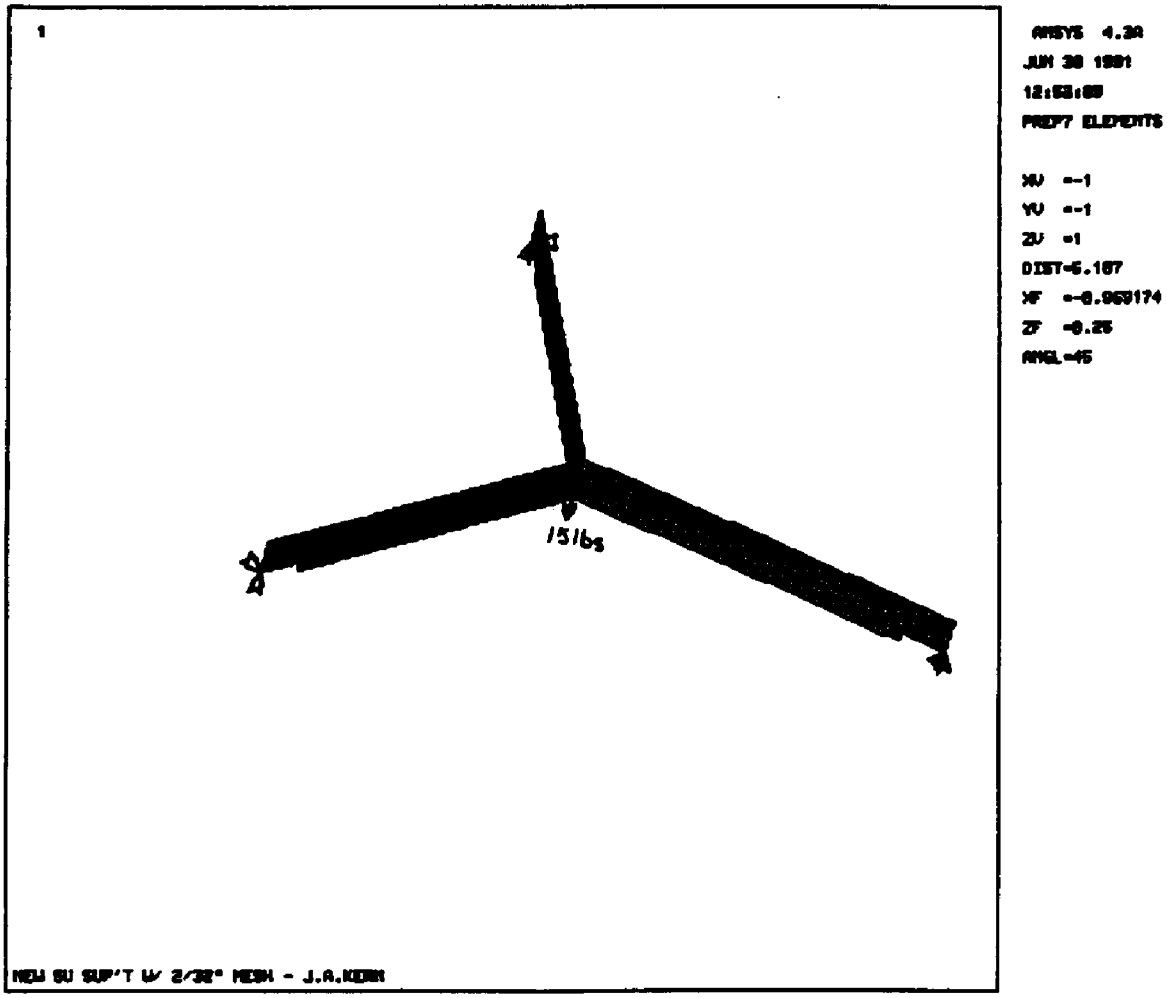


FIGURE 40.- NEW SUPPORT WITH CENTRAL FIFTEEN POUND STATIC LOAD

shows one of the FEA models of the new support with the fifteen pound for solution in batch runs. The results are listed in Table 3.

TABLE 3.

FEA MESH SIZE TRIAL RESULTS

MESH SIZE (INCH)	ELEMENTS PER WIDTH	MAX. VERTICAL DISPLACEMENT (INCH)
1/8 (.1250)	4	0.0403
1/12 (.0833)	6	0.0459
1/16 (.0625)	8	0.0505
1/20 (.0500)	10	0.0502

MESH SIZE SELECTION

It was known from FEA theory that the finer the FEA mesh, the closer the results should be to reality. But, finer meshes yield larger FEA models (i.e., greater numbers of nodes and elements), which will require more disk storage space, and longer computation times for solution. In addition, at some point, the computational errors introduced with larger FEA matrices will cause the results to become less realistic. Therefore, based on the results of this study, it was decided that 1/16" should be used as the mesh size for the load test simulations.

VERIFICATION OF MESH SIZE FOR OLD SUPPORT

To be sure that these same results would hold true for the FEA model of the old support, the basic model for the old support was meshed using 1/12" and 1/16" mesh sizes. These two models were loaded with a central fifteen pounds (15 lbs) load, applied with three five pound static loads through nodes at the split between each welded section of the support. This is displayed in Figure 41.

These two FEA models yield solution with maximum vertical displacements of 0.0212" and 0.0224" respectively. Therefore, it was decided that 1/16" elements would be acceptable for the old supports' load test simulation also.

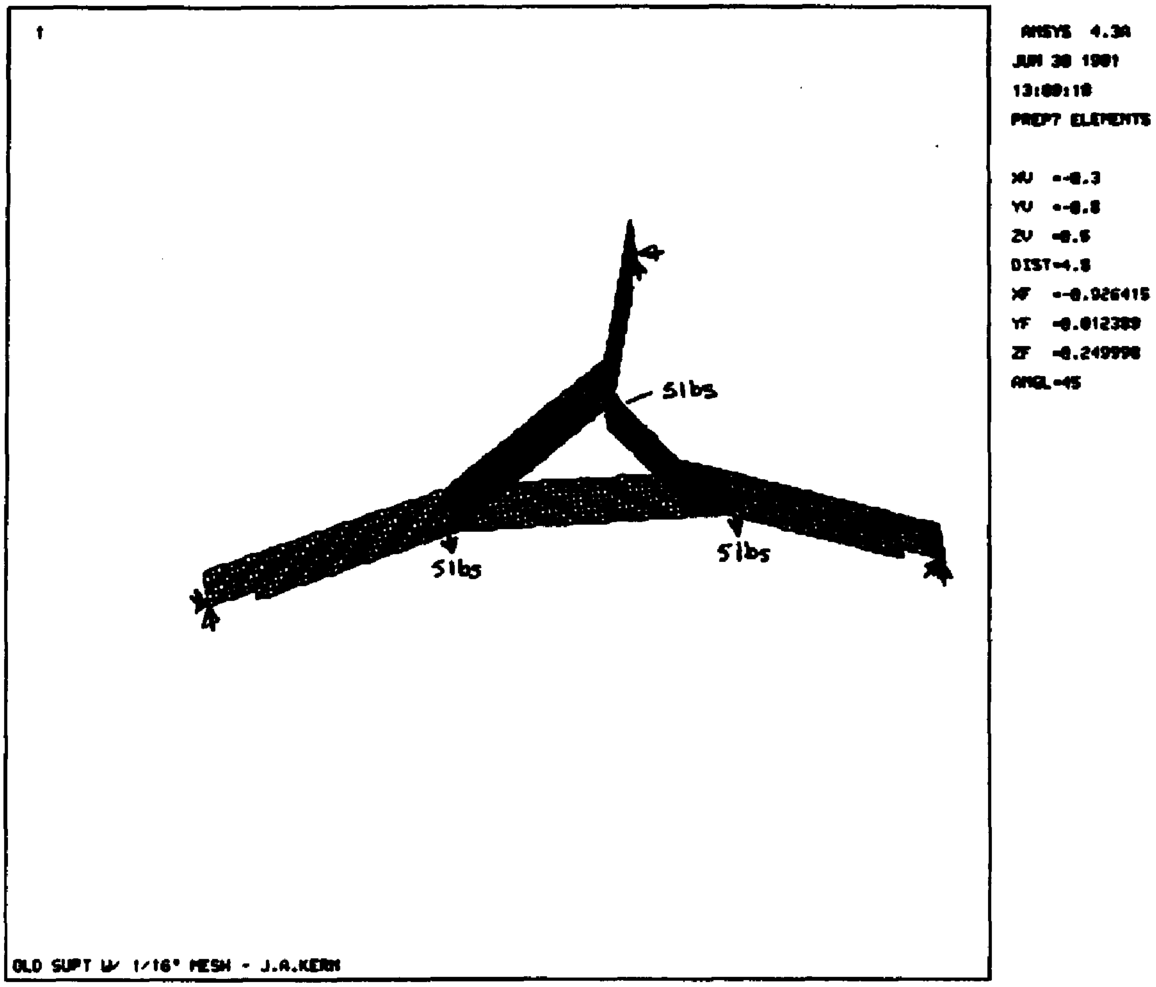


FIGURE 41. - OLD SUPPORT WITH CENTRAL FIFTEEN POUND STATIC LOAD

CHAPTER XIV

PHYSICAL MODEL VERIFICATION

INTRODUCTION

The FEA models that had been constructed and used to determine the optimum mesh size seemed to yield reasonable results. But, as with any FEA analyses, it was advisable to prepare simple physical experiments to verify that the model's were yielding results that were realistic of the displacements that actual surface unit supports of these designs would experience with this type of static loading. Therefore, sample supports were constructed, and tested as outlined in this chapter.

TEST SETUP

In order to duplicate the fifteen pound static loads that were applied to the FEA models of the surface unit supports, discussed in Chapter XIII, simple bench top tests were setup and conducted. Three large round headed, straight blade, wood screws were driven into the top of a hardwood work bench, with their slots aligned with the legs of the surface unit supports (see Figures 42. and 43.). These were placed at the ends of the supports. But, they could not be placed at the extreme ends of the supports exactly like the FEA models' supports, and still expect the supports to be loaded without falling between the screws. Therefore, the results of the physical tests were expected to yield

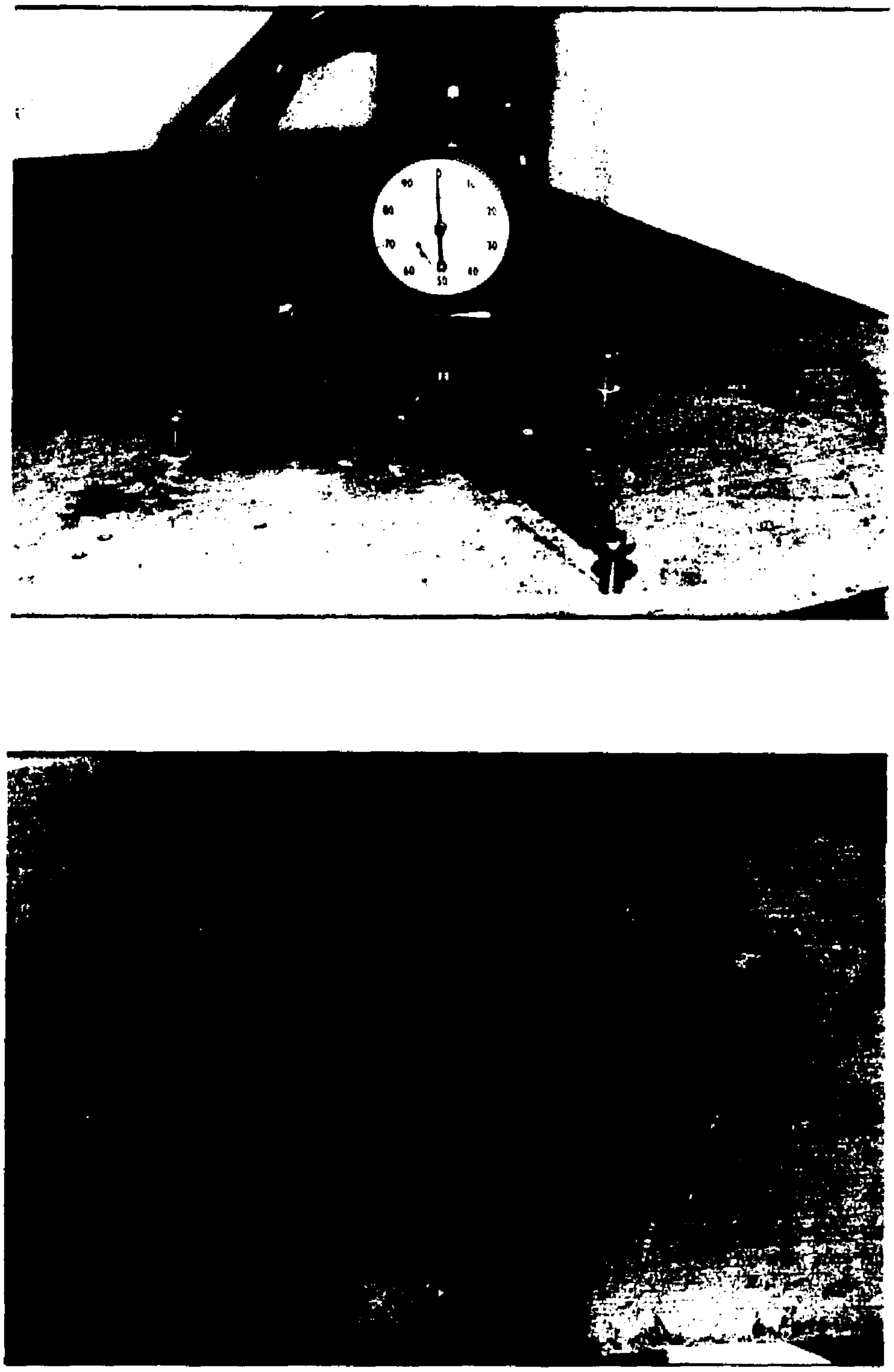


FIGURE 42. - OLD SUPPORT'S VERIFICATION TEST SETUP

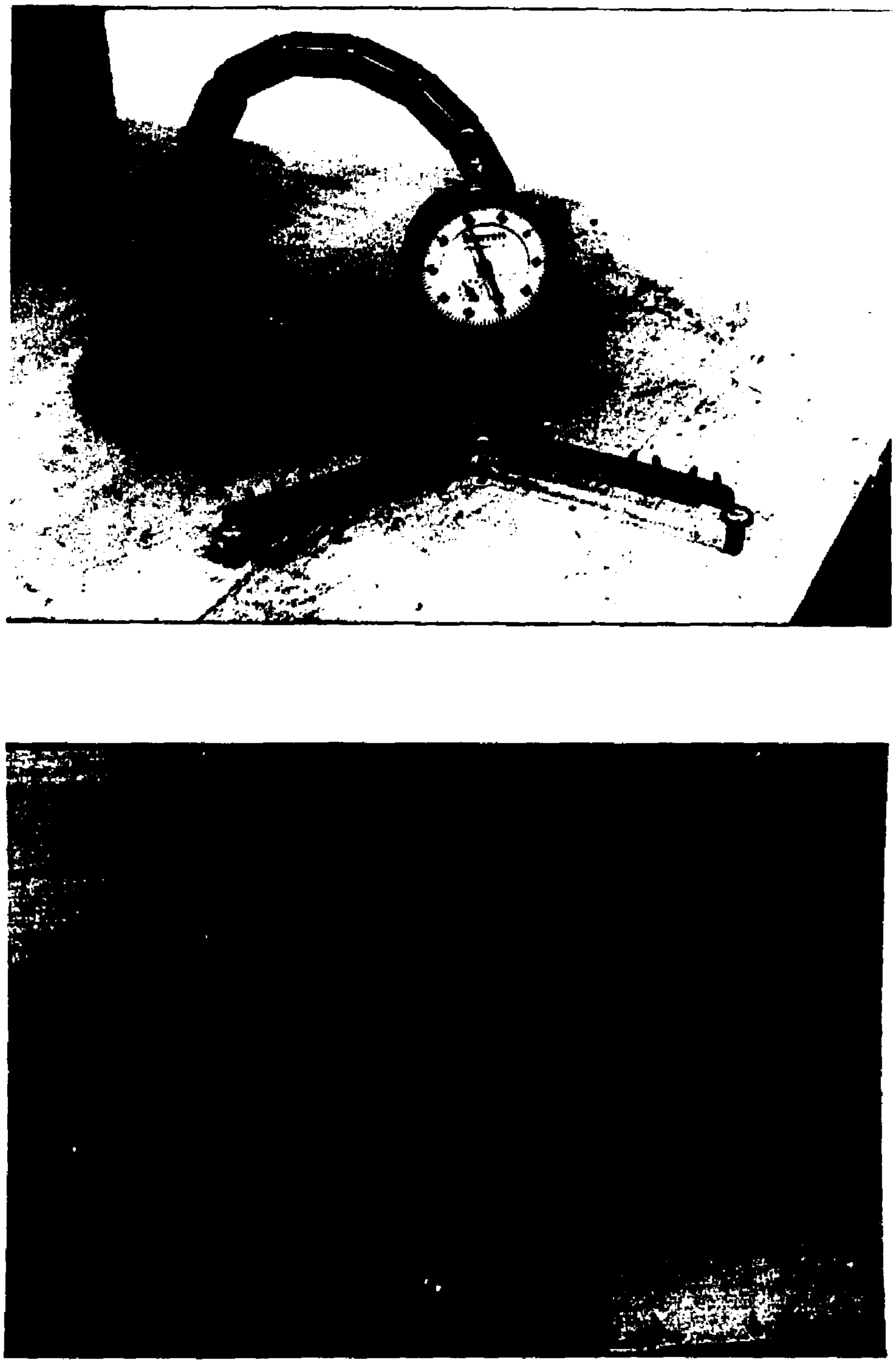


FIGURE 43.. - NEW SUPPORT'S VERIFICATION TEST SETUP

slightly smaller displacements then the FEA models, due to the screws support of the surface unit supports at points a little closer to the center of the supports.

For the old surface unit support, three holes were drilled through the work bench directly under the points on the support where its individual pieces straight sections end and the radii begin the bends into the delta shaped portion of the support (see Figure 42.). For the new surface unit support, one hole was drilled through the work bench at the center of the support's "Y" (see Figure 43.). "Piano" wires were run through these holes to load the supports. Loops were made at the top ends of the wires, and shaped to pull straight down on the top of the surface unit supports. The bottom ends of the wires were attached to the cap of a plastic gallon bottle, and the bottle was filled with weight and screwed onto the cap to load the supports (see Figure 44.).

A dial linear gauge with a gooseneck support and magnetic base was used to measure the supports' displacements when placed under static load. The dial gauge can be seen in Figures 42. and 43. The magnetic base was attached to a heavy flat steel plate and placed on the top of the work bench. The gooseneck was adjusted, and locked into position over the supports at the point at which the deflections were measured. The old supports' deflections were measured at a point at the center of the rear section of the delta portion of the supports between their rear legs. The new supports' deflections were measured at the center of the supports' "Y".

FIGURE 44.- BOTTLE AND WIRES USED TO LOAD THE SUPPORTS OF TESTS

TESTS PERFORMED

Two samples of each surface unit support design were tested to be sure that there was nothing unusual about the samples' results (e.g., a support with weak welds, etc.). Each support was loaded statically with the weight filled bottle once with each of the following loads: 5, 10, 15, 20, and 25 pounds. The bottle was loaded by removing it from the cap, filling it with weights until it weighed five pounds. Then the bottle was screwed onto the cap, and then gently allowed to hang on the wire(s) attached to the support being tested. After the deflection was recorded, and sufficient time had passed to be sure that the support was not yielding under the applied load, the bottle was removed. This process was repeated with the next heavier weight until all five loads had been applied to each of the four supports tested.

TEST RESULTS

Table 4. lists the results of the tests that were performed. From these, it was noted that the FEA models' results were indeed in line with reality. The tests indicated that the old support would deflect approximately 0.010" under a fifteen pound load, compared to .0224" as indicated by the FEA model. The new support design deflected approximately 0.045" when loaded with fifteen pounds, while its FEA model deflected .0505".

These test indicate that the FEA models are too "soft". But, the physical samples were not supported at their extreme ends. Also, their materials were not exactly 0.050" thick. The old supports were 0.054" thick, and the new supports were 0.047" thick. The FEA model's thicknesses were changed and resolved, resulting in the old support

TABLE 4.

VERIFICATION TEST RESULTS

LOAD (LBS)	NEW SUPPORT		OLD SUPPORT	
	SAMPLE #1	SAMPLE #2	SAMPLE #1	SAMPLE #2
5	0.013	0.012	0.002	0.002
10	0.030	0.028	0.007	0.007
15	0.044	0.046	0.009	0.010
20	YIELD	YIELD	0.013	0.012
25	YIELD	YIELD	0.016	0.015

model's deflection being reduced to 0.0178", and the new support model's deflection was increased to 0.0596". The new support's samples were made including strengthening ribs at the "Y", and the production supports were made with production stamping dies. These details would also contribute to the physical parts being stiffer than the FEA models.

CONCLUSIONS

In addition to the differences between physical parts and the FEA models, the FEA results were to be used to compare the relative strength of the new support design to the old design, and not strictly for their raw numerical results. The stresses predicted by FEA models that deflect slightly more than the actual supports would be higher than reality, and thus would yield more conservative conclusions on the new support's strength in comparison with to the old support design. Therefore, the FEA models were judged to be acceptable for use in performing the load test simulation discussed in the next chapter.

CHAPTER XV

FEA ANALYSIS OF PLATE LOADING

INTRODUCTION

As was introduced in Chapters I and X, the primary finite element analyses that were performed for this thesis were designed to simulate a standard "surface unit load test" that GEA has traditionally used to test different support and heating coil designs for structural integrity. In Chapter X the construction of the FEA models of the new and old surface unit supports was discussed. Also, the creation of flat "load plates" above these supports was discussed. The plates were specified as 0.100" thick, and constructed of aluminum alloy 3003. The final steps needed in the preprocessing of the two plate loaded support models was the connection of the load plates to the supports with gap elements, and the pressure loading of the plates. These steps are discussed in this chapter. Once these steps were completed, the remainder of Chapters X through XI were followed in applying I-DEAS Engineering Analysis and ANSYS for preprocessing, ANSYS for model solution, and then ANSYS and I-DEAS for post processing of the results.

SELECTION OF GAP ELEMENT TYPE

The load plates were connected to the surface unit support models using "GAP" finite elements. The reason for using gap elements was so that when the pressure applied to the plates caused them to deflect downward against the supports, the supports could move down away from the plates, if the loads transferred to the outer contact points caused the supports to be deflected a greater distance at their center nodes the plates center nodes. If the plates had been coupled to the supports, or connected to then with rigid spar type elements, rather than using gap elements, the plates might have constrained the supports from moving away from them. That would be not accurately model the contact between actual cooking utensils and surface units' heating coils.

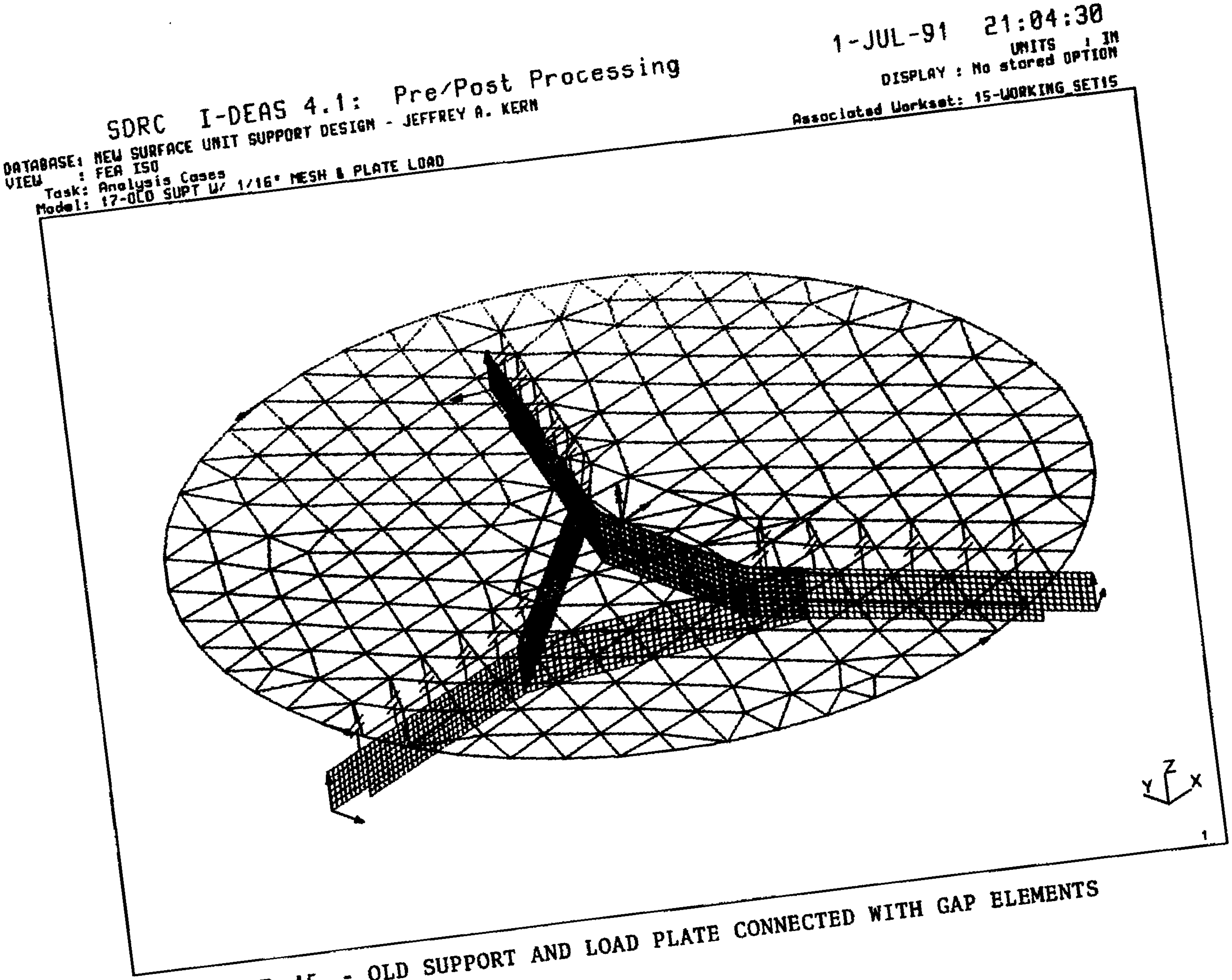
ANSYS provides several gap type elements. I-DEAS' gap element type is referred to as a STIFF52 element in ANSYS, which ANSYS calls a "three-dimensional interface" element. These could have been used here, by they are more complex than was desired. These elements do represent two surfaces which may maintain or break contact during a model's solution, which was desired. But, they are also capable of modeling two surfaces which may slide relative to each other during the solution of the models, taking into account Coulomb friction between them. Although in reality, there is friction between cooking utensils and heating units' coils, the effect that this would have on the supports' deflections and stresses is assumed to be negligible for these analyses. This seems to be a fair assumption given that their overall deflections are relatively small, and the range of actual friction coefficients large, the smallest being quit small. Because of this, a less complex gap element type was desired. All that was required was an element type that would transfer no force to the supports when the gap between them opened up, and that would transfer force to the supports when the plate and the gap elements were in contact.

A search of ANSYS element library indicated that the desired element type was STIFF10, called "compression-only spar" elements. These elements can actually be used for two purposes depending on the setting of its keyopt(3). If this is set to 1, as was done for this thesis, the elements are compression-only spar (gap) elements. But, if keyopt(3) = 0 (its default), they act as tension-only spars, or as "cables", "ropes" or "chains". In cases where this is applied, these elements will only transmit uniaxial forces when they are placed in tension. But, for this thesis (with keyopt(3) = 1), these elements could only transfer forces to the supports when they were placed in compression.

CONNECTION OF THE LOAD PLATES TO THE SUPPORTS

In I-DEAS gap elements are created in the MODEL_PREPARATION Task by setting the DEFault ELEMENT TYPE_OF_ELEMENT to GAP. I-DEAS prompts the user to specify whether the gap elements are to be NODE_TO_NODE or NODE_TO_GROUND. For this thesis NODE_TO_NODE gap elements were applied. After this default type was setup, the ELEMENTS CREATE command was used to pick the nodes that were desired at the ends of each gap. Note that the nodes had to be displayed in order to pick them as VISIBLE on the graphics screen.

The gap elements were placed at the nodes at the points the surface units' heater coils cross the supports. Identification of these nodes was assisted by making the working set containing the arcs representing the heater coils visible prior to executing the ELEMENT CREATE command. Figures 45 and 46 display the support and load plate models connected with the gap elements (which are are symbolized like capacitors).



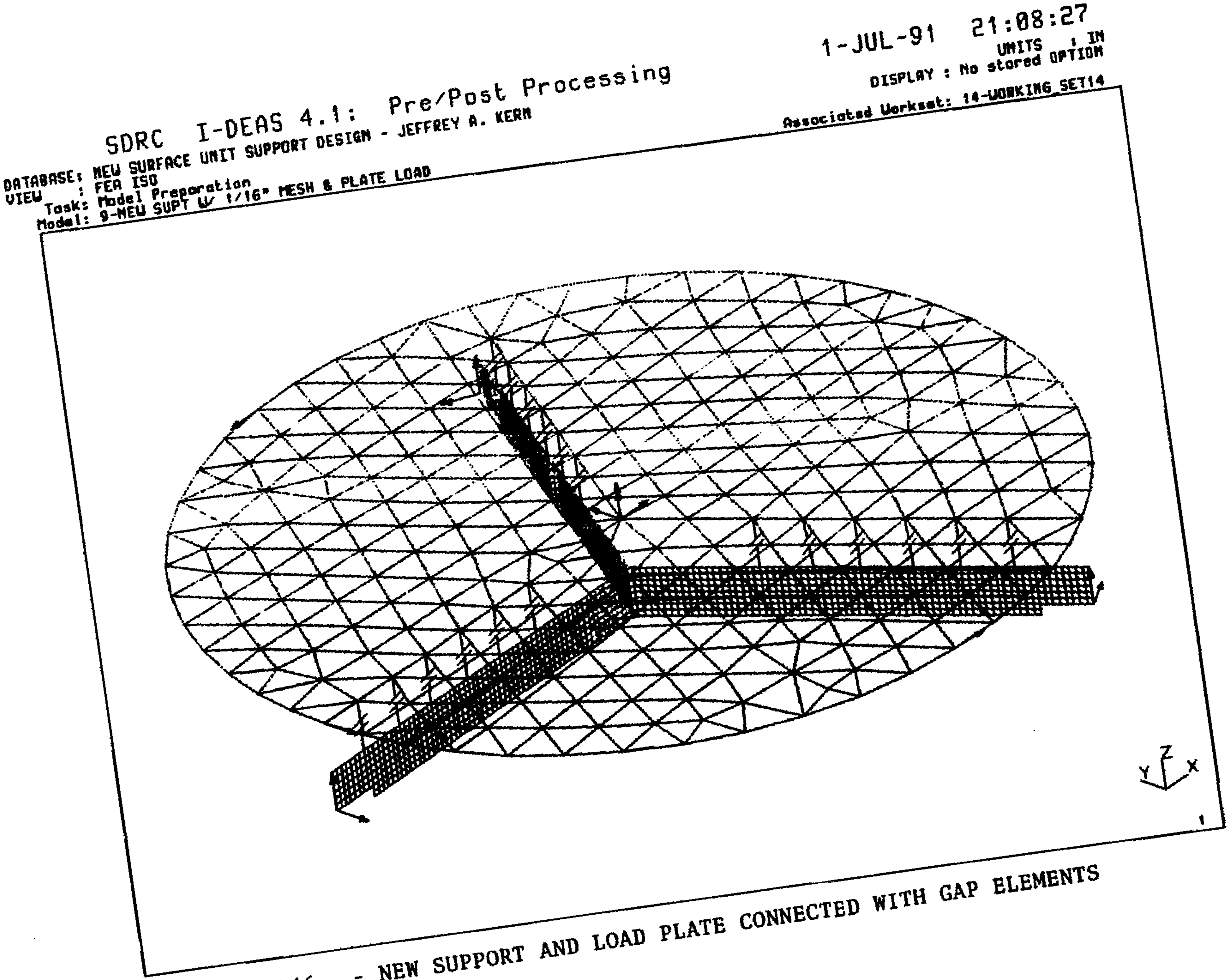
SDRC I-DEAS 4.1: Pre/Post Processing

1-JUL-91 21:04:30

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW : FEA ISO
Task: Analysis Cases
Model: 17-OLD SUPT W/ 1/16" MESH & PLATE LOAD

UNITS : IN
DISPLAY : No stored OPTION
Associated Workset: 15-WORKING SET15

FIGURE 45. - OLD SUPPORT AND LOAD PLATE CONNECTED WITH GAP ELEMENTS



SDRC I-DEAS 4.1: Pre/Post Processing

1-JUL-91 21:08:27

UNITS : IN

DISPLAY : No stored OPTION

Associated Workset: 14-WORKING_SET14

```
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW    : FEA ISO
   Task: Model Preparation
  Model: 9-NEW SUPT W/ 1/16" MESH & PLATE LOAD
```

FIGURE 46 . - NEW SUPPORT AND LOAD PLATE CONNECTED WITH GAP ELEMENTS

CORRECTION OF GAP ELEMENT SPECIFICATION IN ANSYS

Because I-DEAS created STIFF52 gap elements during the preprocessing of the models, before they were transferred into ANSYS, the transfer files were edited to change these elements from type 52 to type 10.

Once the models had been written out from I-DEAS, edited, and read onto ANSYS for final preprocessing, the ANSYS on-line help was used to review the documentation for these elements. This was accessed by typing:

```
STIFF, 10 <RETURN>.
```

This indicated that these elements have two applicable real properties, area, and gap per unit length; and one applicable material property, EX. For this thesis, the gap elements were intended to be very stiff, and small. Therefore, the real constants declaration line for the gap elements was changed to read:

```
R, 3, .049, 0.0.
```

This specifies gap elements with the cross sectional area of a 1/4" diameter cylinder, and initially closed (i.e., no initial gaps). The material properties of the gap elements were changed to read:

```
EX, 3, 0.28E+08.
```

This made the gap elements represent stainless steel cylinders standing on top of the supports.

APPLICATION OF PRESSURE TO THE LOAD PLATES

These analyses required the application of a uniform pressure to the top faces of the load plates' elements, with a pressure value (in PSI) that would yield a vertical force pushing down on the plates with a total of twenty-five pounds of force. In order to determine the appropriate pressure value for each supports' load plates (which have slightly different diameters), the working set's curves were made visible. Once the curves were visible, the radii of the two load plates were measure with the GEOMETRY_DEFINITION Tasks LIST ENTITY command. This allowed the author to determine that the old support's load plate has a 4.176299" radius, and the new support's load plate has a 4.210161" radius.

Once the load plates' radii were known, the face pressure for each plate was calculated as follows:

for the old support: $FP = (25 \text{ lbs}) / (\text{Pi} \times (4.176299 \text{ in})^2)$

$= 0.456254297$ PSI;

and,

for the new support: $FP = (25 \text{ lbs}) / (\text{Pi} \times (4.210161 \text{ in})^2)$.

$= 0.448944575$ PSI.

The commands needed to apply these pressures to the load plates are located in I-DEAS Preprocessing Module's ANALYSIS_CASES Task. The STRUCTURAL_LOADS menu was used to create a load set named "25# PLATE LOAD (.456254297 PSI)" for the old support's model, and a load set named "25# PLATE LOAD (.448944575 PSI)" for the new support's model. Then the FACE_PRESSURE command was used to access the commands needed to apply the desired pressures. Its DEFaults sub-menu was used to specify the PRESSURE for each load plate, and a color for the display of the pressures.

After the default pressure and color values had been setup, the FACE_PRESSURE CREATE command was used to apply the constant pressure values to the upper face of each element of the load plates. Figures 47 and 48 show the pressures as they appear in the I-DEAS display.

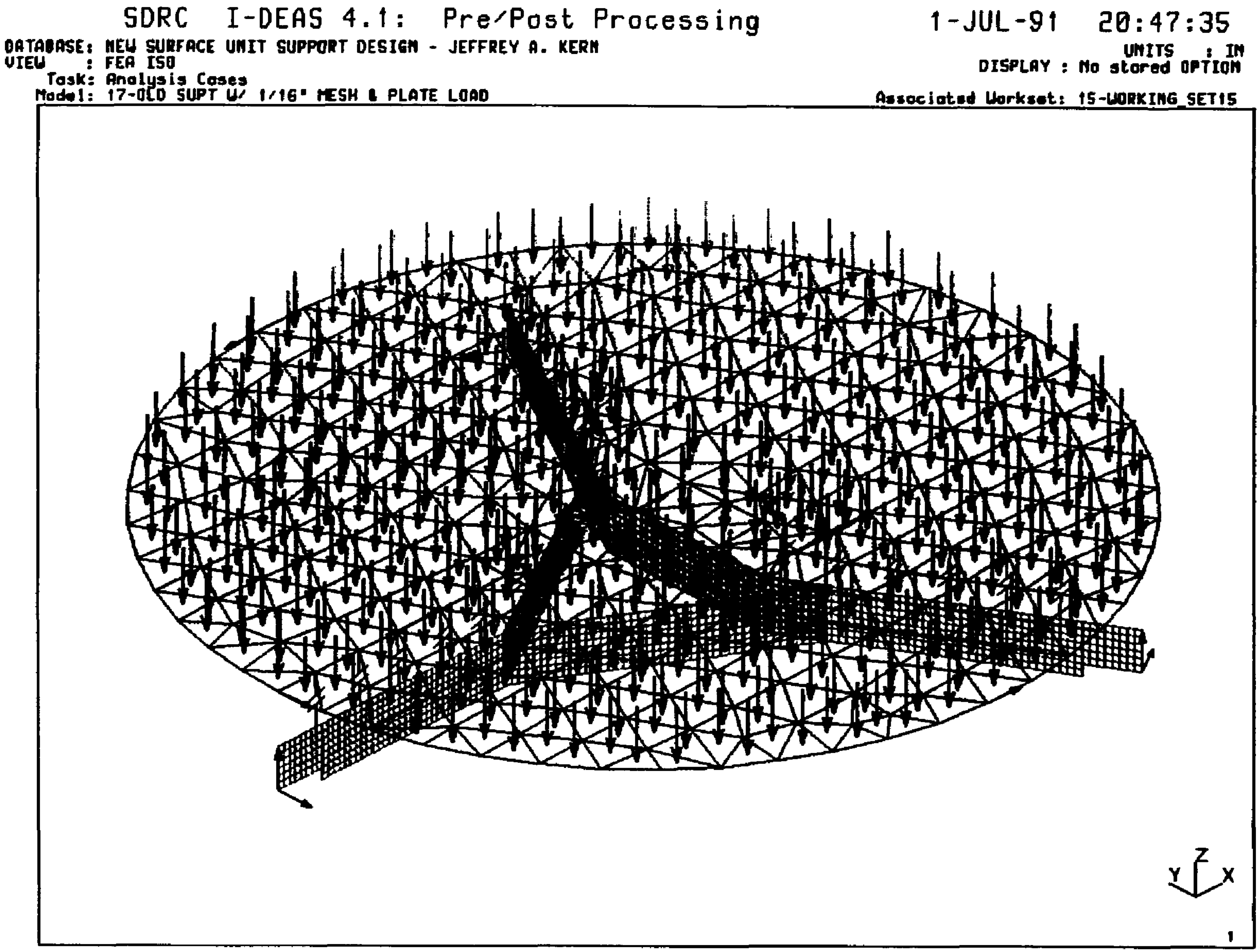
TRANSFER OF THE MODELS FROM I-DEAS TO ANSYS

Once the load plate's had been connected to the supports, and pressure loaded, the procedures discussed in Chapters X and XI were followed in transferring the models to ANSYS. Once in ANSYS, the final preprocessing was completed.

PROCEDURE FOR ITERATIVE SOLUTIONS[16]

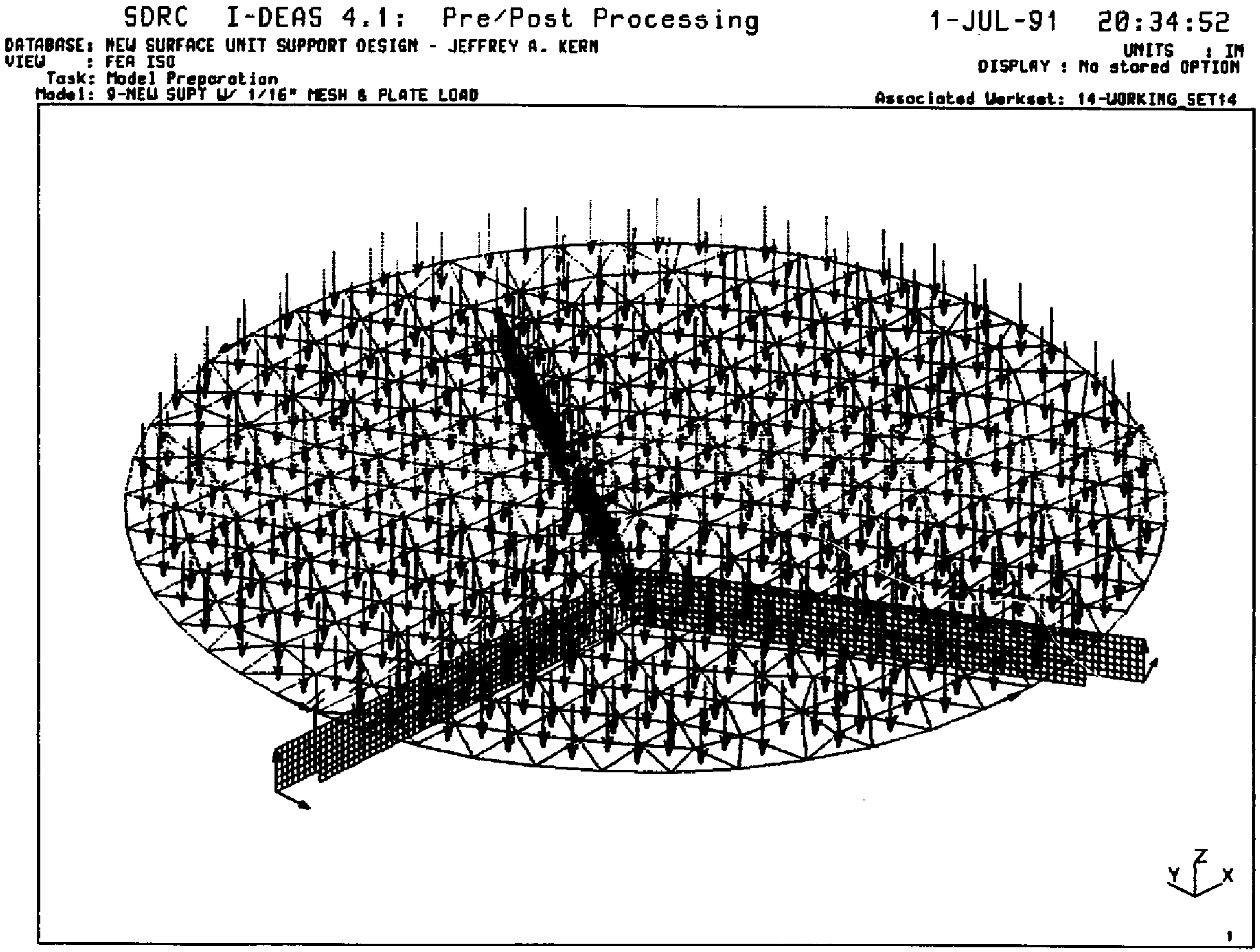
Because gap elements were used in these analyses, the solution of the plate loaded FEA models required iterative solutions. The gap elements are "bilinear elements". That is, they have two different stiffness values, one being zero, depending upon the relative displacement of the support and plates' elements. These elements have two different status conditions, open or closed. This requires the program to start with the gap elements in the closed position (as specified), and then solve the model for the first iteration. If at the end of that iteration the status of any of the gaps have changed, a convergent solution has not been obtained. In this case, the model must be solved for a second iteration, starting with the status of the gaps from the end of the first iteration. This iterative procedure must be repeated until the status of all of the gap elements is the same at the end of an iteration as they were at the beginning of that iteration.

[16] Gabriel J. DeSalvo, Ph.D., Robert W. Gorman, ANSYS Engineering Analysis System User's Manual, Swanson Analysis Systems, Inc., Houston, Pennsylvania, 1987, 2.30.1 - 2.30.2.



```
        SDRC  I-DEAS 4.1:  Pre/Post Processing              1-JUL-91  20:47:35
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                UNITS   : IN
VIEW    : FEA ISO                                               DISPLAY : No stored OPTION
   Task: Analysis Cases
  Model: 17-OLD SUPT W/ 1/16" MESH & PLATE LOAD          Associated Workset: 15-WORKING_SET15
```

FIGURE 47. - OLD SUPPORT'S LOAD PLATE WITH PRESSURES APPLIED



```
SDRC  I-DEAS 4.1:  Pre/Post Processing            1-JUL-91  20:34:52
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : FEA ISO                                     DISPLAY : No stored OPTION
    Task: Model Preparation
   Model: 9-NEW SUPT W/ 1/16" MESH & PLATE LOAD     Associated Workset: 14-WORKING SET14
```

FIGURE 48 . - NEW SUPPORT'S LOAD PLATE WITH PRESSURE APPLIED

In order to have ANSYS perform an iterative solution for a model, the ITER command must be executed before writing out an analysis file with the AFWRITE command. The format for this command is:

ITER, NITTER, NPRINT, NPOST.

NITTER is the number of iterations that are to be performed per load step during model solution. NPRINT defines the print frequency, which determines which iterations' are printed. If this is set to 1, every iteration is printed, if set to 2, every other, etc. NPOST, like NPRINT, determines which iterations are written onto solution FILE12.

If the user knows that a model's solution should converge within a specified number of iterations, he can enter a positive number for NITTER. For example, if the user knows (possibly from previous solutions of the same model under other load conditions) that if a convergent solution is not reach within 10 iterations, there must be a problem with the model. Then, by using 10 for NITTER, he can limit the amount of computation time that might be wasted in having ANSYS attempt to solve a non-convergent model. The model will be iterated ten times, and the user will have to interrogate the printed or post processing data to determine if convergence had been reached before the tenth iteration.

Note that if convergence has not been reached in the specified number of iterations, the iterative solution can be restarted at the point it was left off.[17] But, only if FILE3 or MODEL_NAME.F03 (e.g., N116.F03) if the batch command file was used to access ANSYS for solving the FEA models, remains in the users disk directory. Therefore, if an iterative solution is being run, it may be in the users best interest to

[17] Ibid., 2.23.

specify that all files except the model and solution files should not be deleted at the end of the batch solution run, so that FILE3 will be available for a restart. The ANSYS Engineering Analysis System User's Manual, page 2.23 should be consulted if a restart is desired.

ANSYS provides a special feature to assist the user in specifying the number of iterations, which it calls the nonlinear convergence criteria. The user may enter a negative value for NITTER. Then, when the iterative solution is run, a maximum of |NITTER| iterations will be performed for each load step. But, if the solution converges before this maximum number of iterations, the iteration process will be stopped. At that point, ANSYS will perform a partial iteration, identified as iteration |NITTER|, to output the results. When this option is used, all post data controls default to |NITTER|.

ANSYS also makes it easy for the user to specify the converged iteration when retrieving the results into POST1 for post processing. The SET command discussed in Chapter XII, which is used to extract the requested results from the solution file and store them in memory for post processing, can be used with 0 entered for ITER. This cause ANSYS to automatically locate the last iteration in the results file for the load step specified and store that iteration's in memory for post processing. For example, if NITTER is entered as -10, and the convergent solution was reached at iteration 6, by entering:

```
SET, 1, 0
```

in POST1, the results from the sixth iteration will be retrieved.

PLATE LOAD ANALYSIS RESULTS

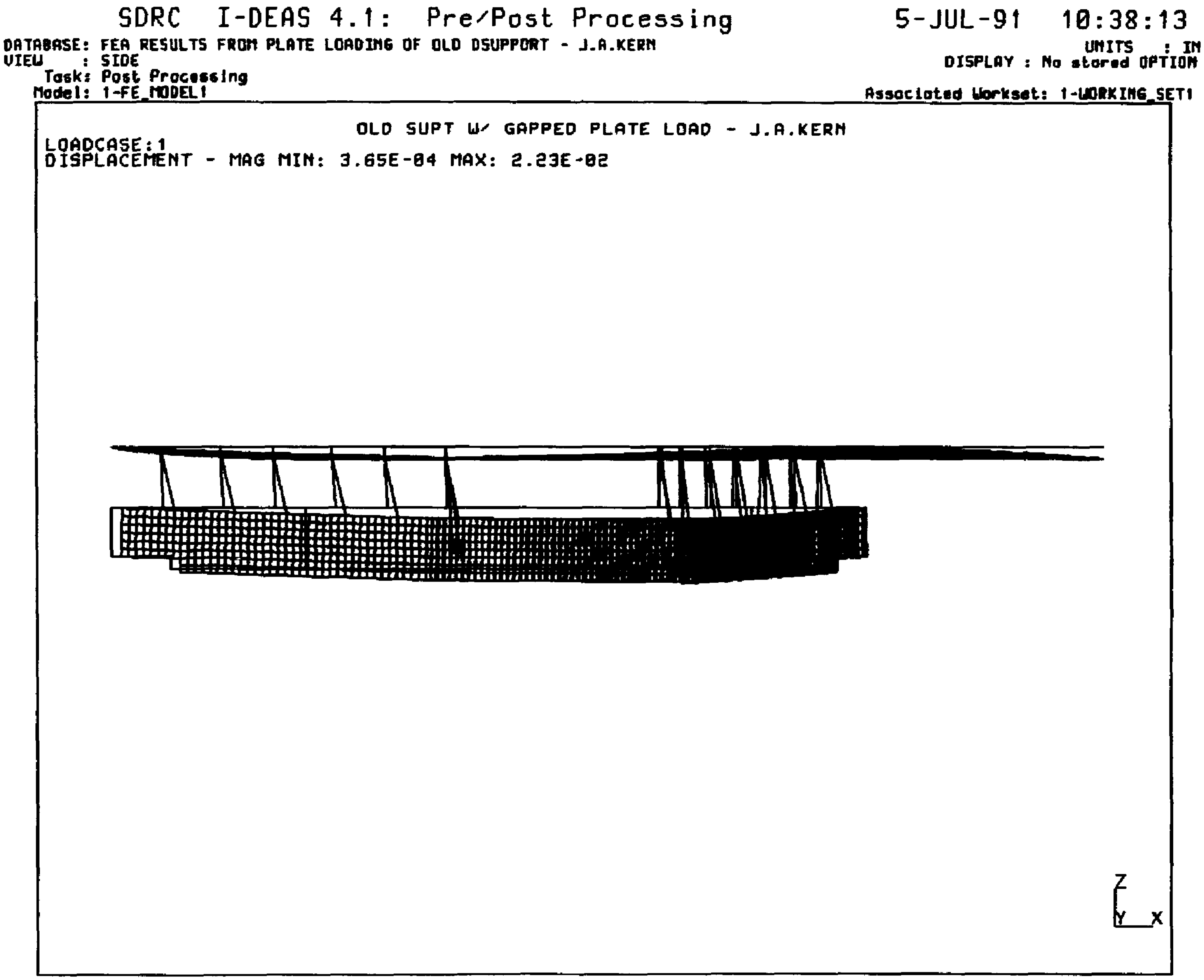
The iterative solutions to the plate loaded models with gap elements connecting the load plates to the supports both converged in less than ten iterations. The old support required six iterations before the gaps' status (i.e., staying closed or open) remained unchanged from the start to the finish of the iteration. The results of these analyses definitely show that gap elements, rather than rigid links, were needed between the plates and the supports. The final status of the gap elements was checked in ANSYS post processing by entering:

```
STRESS,GAPS,10,6 <RETURN>,
SET <RETURN>,
ESEL,TYPE,2 <RETURN>,
PRSTR,GAPS <RETURN>.
```

The status of the STIFF10 gap elements is 1 when they are in compression, and 2 when they are open. The old support had only 7 of its 21 gap elements closed and the new support had only 6 of its 18 gap elements closed when these models were solved.

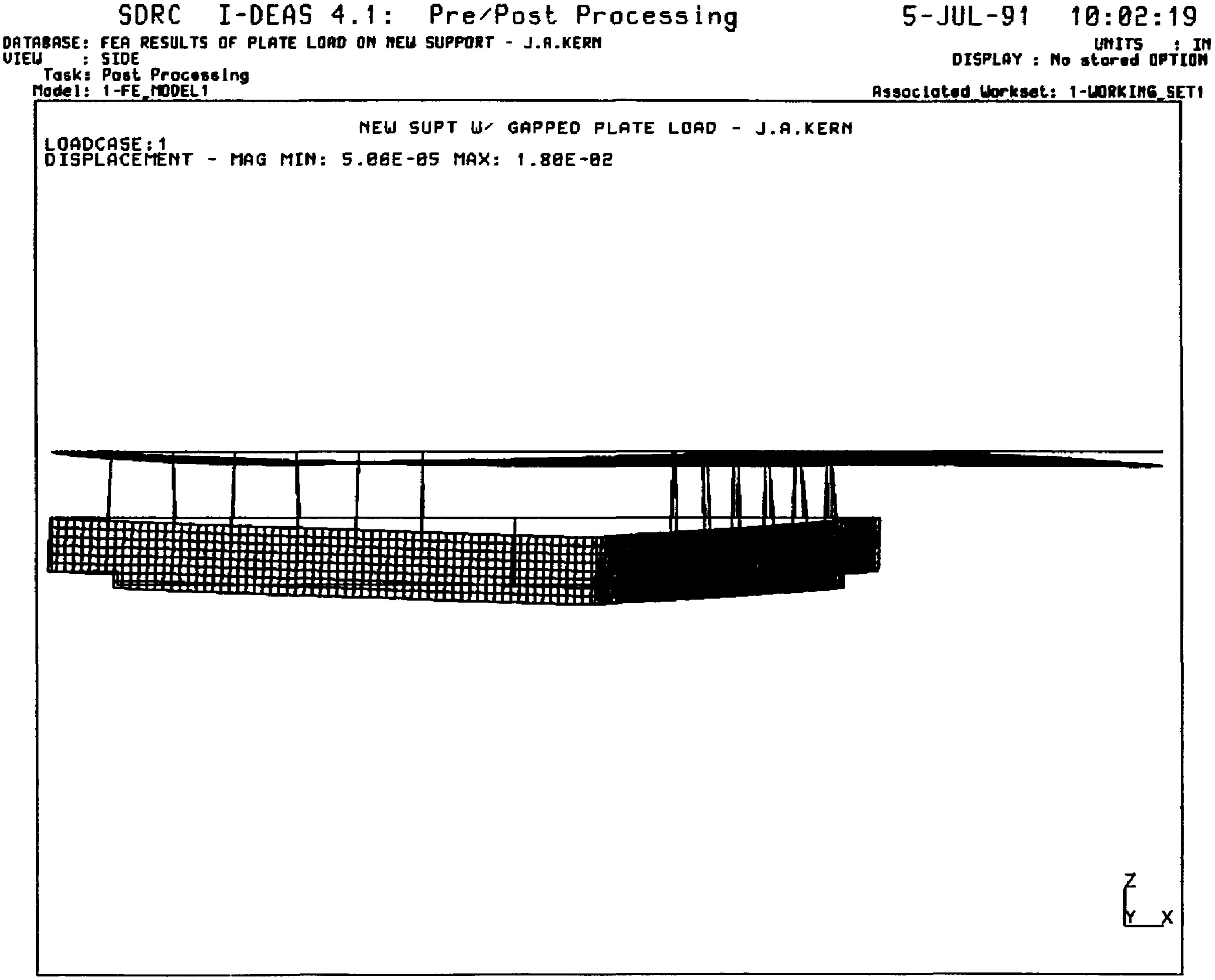
The maximum vertical deflection calculated for the supports were 0.012" for the old support and 0.015" for the new support. Figures 49 and 50 show the displacement plots of the supports under these loading conditions. The SCALE_OF_DISTORTION factor in these plots is 10:1. Therefore, the displacements shown are ten times the actual displacements, with the support's (1/2" wide) to scale.

For the old support under these loading conditions, the FEA model's solution yielded: a maximum Maximum Principal stress (SIG1) of 3,360 psi; a minimum Minimum Principal stress (SIG3) of - 7,049 psi; and a maximum Von Mises equivalent stress of 8,076 psi.



```
SDRC  I-DEAS 4.1:  Pre/Post Processing            5-JUL-91  10:38:13
DATABASE: FEA RESULTS FROM PLATE LOADING OF OLD DSUPPORT - J.A.KERN       UNITS   : IN
VIEW    : SIDE                                                 DISPLAY : No stored OPTION
   Task: Post Processing
  Model: 1-FE_MODEL1                                  Associated Workset: 1-WORKING_SET1
                    OLD SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
DISPLACEMENT - MAG MIN: 3.65E-04 MAX: 2.23E-02
```

FIGURE 49 . - OLD SUPPORT'S DISPLACEMENTS UNDER 25 LBS PLATE LOAD



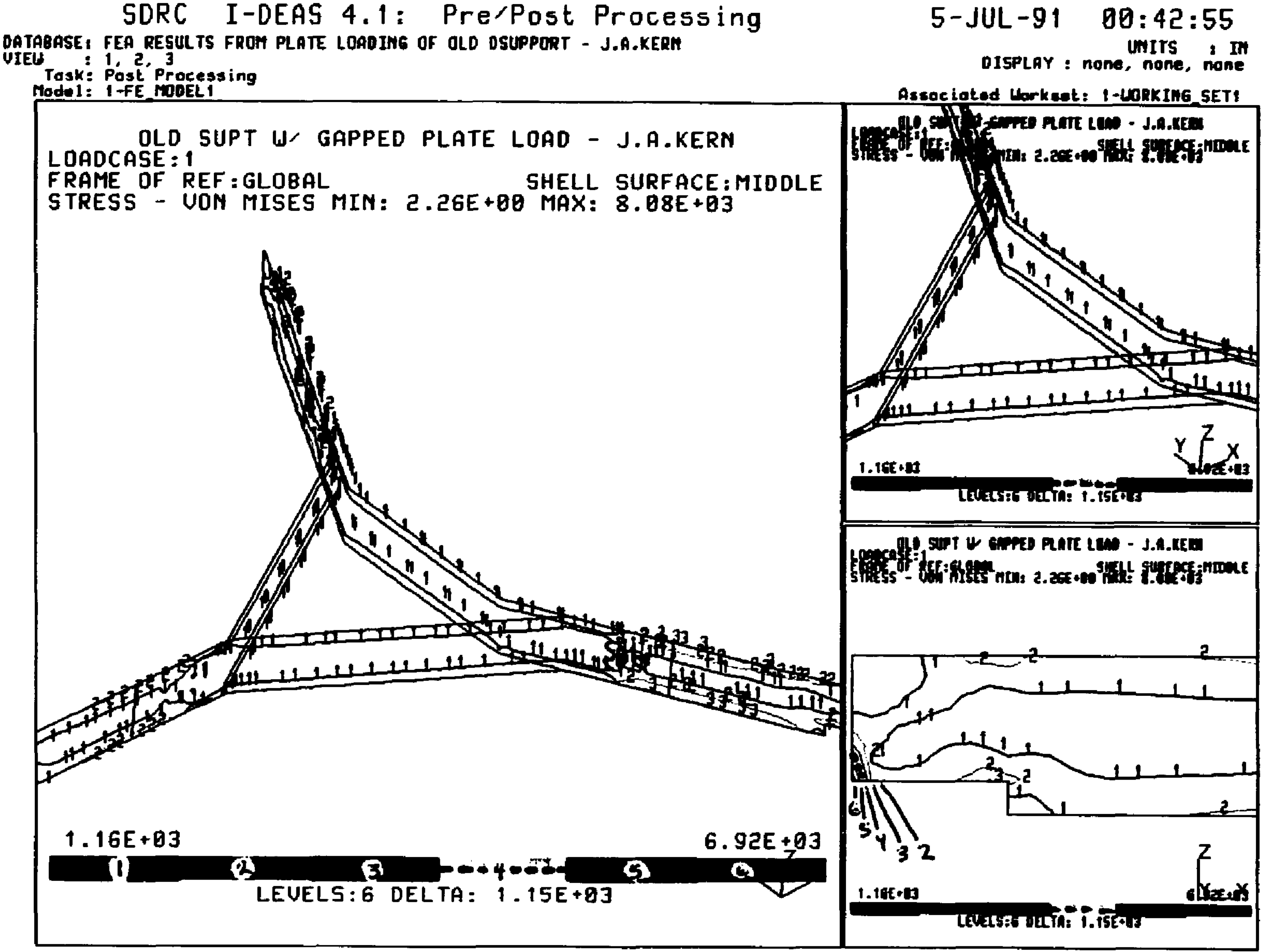
```
SDRC  I-DEAS 4.1:  Pre/Post Processing                5-JUL-91  10:02:19
DATABASE: FEA RESULTS OF PLATE LOAD ON NEW SUPPORT - J.A.KERN          UNITS   : IN
VIEW    : SIDE                                          DISPLAY : No stored OPTION
    Task: Post Processing
   Model: 1-FE_MODEL1                               Associated Workset: 1-WORKING_SET1
```

```
                    NEW SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
DISPLACEMENT - MAG MIN: 5.06E-05 MAX: 1.80E-02
```

FIGURE 50. - NEW SUPPORT'S DISPLACEMENTS UNDER 25 LBS PLATE LOAD

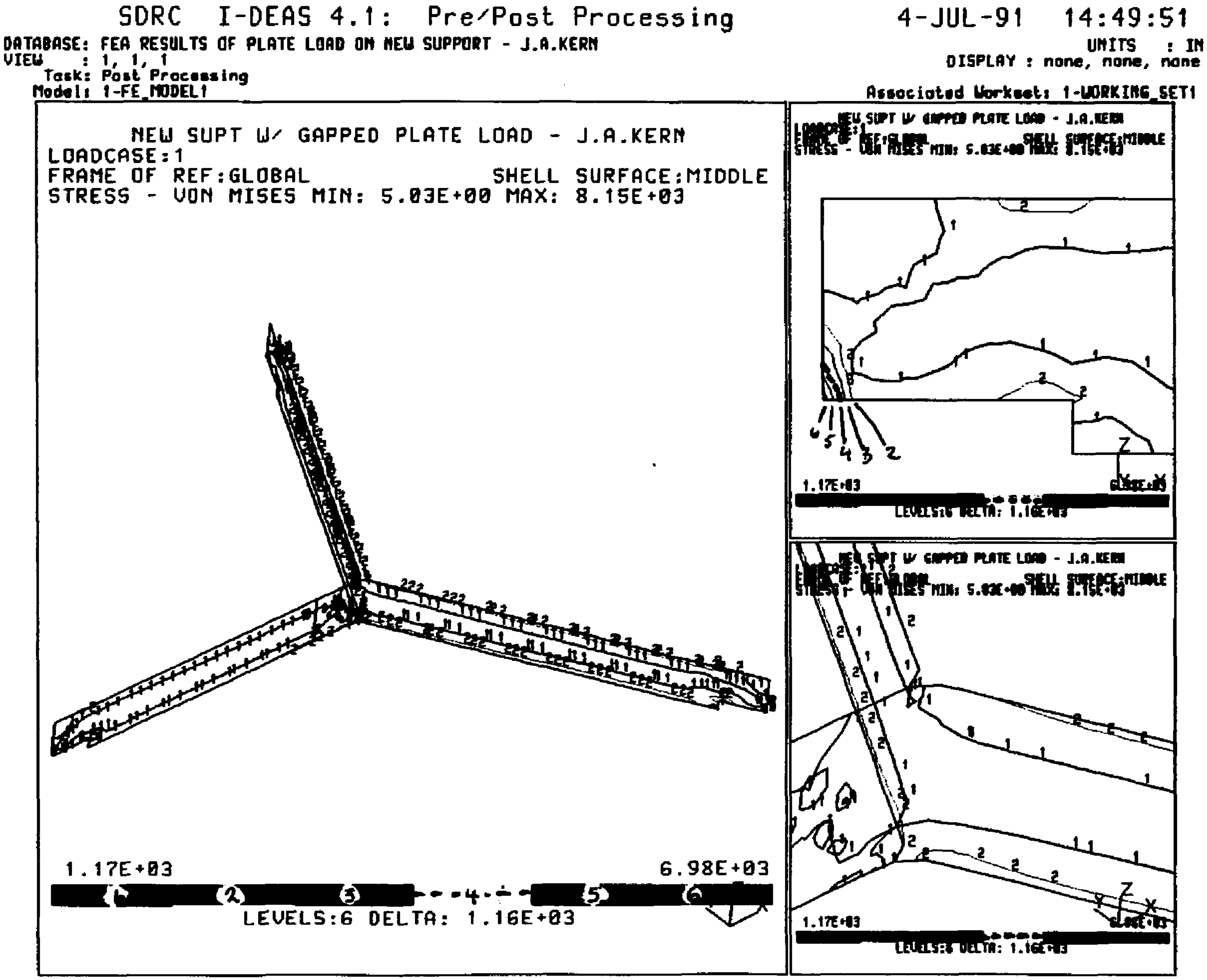
For the new support under these loading conditions, the FEA model's solution yielded: a maximum Maximum Principal stress (SIG1) of 3,976 psi; a minimum Minimum Principal stress (SIG3) of - 7,110 psi; and a maximum Von Mises equivalent stress of 8,147 psi.

Principal stress reports from I-DEAS post processing sessions are listed in Appendix VII. Figures 51 and 52 show stress contour plots of the Von Mises stress values calculated at the shell element's middle surface for each of the two support designs.



```
SDRC  I-DEAS 4.1:  Pre/Post Processing                5-JUL-91  00:42:55
DATABASE: FEA RESULTS FROM PLATE LOADING OF OLD DSUPPORT - J.A.KERN      UNITS   : IN
VIEW    : 1, 2, 3                                      DISPLAY : none, none, none
   Task: Post Processing
  Model: 1-FE_MODEL1                            Associated Workset: 1-WORKING_SET1

        OLD SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
FRAME OF REF:GLOBAL                       SHELL SURFACE:MIDDLE
STRESS - VON MISES MIN: 2.26E+00 MAX: 8.08E+03

1.16E+03                                              6.92E+03
        LEVELS:6 DELTA: 1.15E+03

        OLD SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
FRAME OF REF:GLOBAL                       SHELL SURFACE:MIDDLE
STRESS - VON MISES MIN: 2.26E+00 MAX: 8.08E+03
1.16E+03                                              6.92E+03
        LEVELS:6 DELTA: 1.15E+03

        OLD SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
FRAME OF REF:GLOBAL                       SHELL SURFACE:MIDDLE
STRESS - VON MISES MIN: 2.26E+00 MAX: 8.08E+03
1.16E+03                                              6.92E+03
        LEVELS:6 DELTA: 1.15E+03
```

FIGURE 51. - OLD SUPPORT'S VON MISES STRESS CONTOURS



```
SDRC  I-DEAS 4.1:  Pre/Post Processing                4-JUL-91  14:49:51
DATABASE: FEA RESULTS OF PLATE LOAD ON NEW SUPPORT - J.A.KERN          UNITS   : IN
VIEW    : 1, 1, 1                                          DISPLAY : none, none, none
  Task: Post Processing
 Model: 1-FE_MODEL1                                  Associated Workset: 1-WORKING_SET1
```

```
     NEW SUPT W/ GAPPED PLATE LOAD - J.A.KERN
LOADCASE:1
FRAME OF REF:GLOBAL                    SHELL SURFACE:MIDDLE
STRESS - VON MISES MIN: 5.03E+00 MAX: 8.15E+03
```

FIGURE 52. - NEW SUPPORT'S VON MISES STRESS CONTOURS

CHAPTER XVI

SUMMARY AND CONCLUSIONS

Based on the results of the plate loaded FEA model's solutions, it appears that the new "Y" shaped support design should function acceptably, from a strength standpoint alone. The new support's maximum Maximum Principal stress is approximately 18.3% higher than the old support's. This is well below the yield stress for the surface unit support's material, which is 55 ksi. Additionally, the maximum stresses calculated by these models are concentrated at the steps in the ends of the supports and at the point contacts between the load plates and the supports. Therefore, if the steps had been rounded, and if the contact areas between the plates and supports were increased, the maximum stresses would be even lower.

Also, the maximum vertical displacement of the new support was calculated to be 0.015", and the samples tested (see Chapter XIV) did not yield until they had been displaced at least 0.045". Therefore, the new support's should not have a strength problem.

There are other things that GEA should consider in deciding if the new design should be used. For one thing, simple beam equations were used to calculate the deflections of the new support design, using standard mechanics of materials theories. This yielded maximum deflection of 0.009" when subjected to a fifteen pound central load. But, under the same load, the shell FEA model deflected 0.050" (see Chapter XIII); and the sample supports deflected 0.045" (see Chapter XIV).

In order to investigate these differences, the nodes in the FEA model along the line between the top and bottom tangent points of the support pieces' center bends (where the overlapping materials first make contact) were merged to simulate more complete welding of the two pieces of the new support. This variation of the new support design's shell FEA model was then solved under the same central fifteen pound vertical load. The resulting maximum deflection was 0.0095". This indicates that the quality of the welds is very important. If the pieces of the new surface unit support could be welded together completely in the overlapping area, the support's maximum deflection could be reduced by a factor of five.

Another thing that GEA should consider is that the new design will have a different dynamic response than the old design. This may effect the sounds produced when surface units are excited by vibrations or impact. Therefore, due to the effects that "noisy" or "cheep sounding" surface units have on customers, GEA might want to extend this work to include modal analyses.

APPENDIX I

I-DEAS QUICK REFERENCES

KEY	COMMAND	MNEMONIC	COMMAND'S RESULTS
<F1>	AUTOSCALE	AU	SCALES ENTITIES TO FILL VIEWPORT
<F2>	DRAW	DR	DISPLAYS DRAWING W/ CURRENT VIEW
<F3>	GLOBAL_DISPLAY	GD	DISPLAYS DRAWING W/ GLOBAL DISPLAY
<F4>	ZOOM	ZM	USER PICKS DIAGONALS FOR ZOOM IN
<F5>	MAGNIFY 141.4%	MG 1.414	SCALES VIEW UP BY 41.4%
<F6>	MAGNIFY 70.7%	MG 0.707	SCALES VIEW DOWN BY 29.3%
<F7>	ROTATE_ABOUT_SCREEN	RS	ROTATE ENTITIES ABOUT SCREENS AXES
<F8>	REDISPLAY	REDI	ERASES AND REDRAWS W/ CURRENT VIEW
^C	<CONTROL>	N/A	DEPRESS ^C TOGETHER WITH "KEY"
^C C	STOP DISPLAY	N/A	STOPS TRANSMISSION OF GRAPHICS
^C O	OUTPUT ON/OFF	N/A	TOGGLES OUTPUT ON & OFF
^C Q	RESTART DIALOG	N/A	RESTARTS STOPPED TEXT TRANSMISSION
^C S	STOP DIALOG	N/A	STOPS (HOLDS) TEXT TRANSMISSION
^C T	SYSTEM STATUS	N/A	DISPLAYS COMPUTER SYSTEM'S STATUS
^C X	CANCEL INPUT	N/A	CANCELS INPUTS NOT RECEIVED BY CPU
^C Y	BREAK OUT	N/A	LAST RESORT BREAK OUT OF I-DEAS

APPENDIX II

ANSYS QUICK REFERENCES

COMMAND	ROUTINE	COMMAND'S RESULTS
AFWRITE	PREP7	ANALYSIS FILE WRITE & DATA CHECK
/B	ALL	DISPLAY GRAPHICS & TEXT TOGETHER
DOCU	ALL	ON-LINE DOCUMENTATION ON MODULE
DOCU,ALL	ALL	ON-LINE DOCU' ON MODULE W/ FORMATS
DOCU,/CSMD	ALL	MODULE'S "/" COMMANDS ON-LINE DOCU
DOCU,NAME	ALL	ON-LINE DOCU' ON COMMAND NAMED
DOCU,GRPH	PREP7	LIST OF GRAPHICS CONTROL COMMANDS
/EOF	ALL	EXIT ROUTINE WITHOUT SAVING WORK
FINI	ALL	FINISH ROUTINE AND SAVE WORK
/G	ALL	GRAPHICS ONLY (DIALOG OFF)
/INPUT,FILE,EXT	ALL	READ NAMED INPUT FILE (FILE.EXT)
/INT	BEGIN	SETS ANSYS FOR INTERACTIVE USE
MODU	ALL	LISTING OF MODULES IN ROUTINE
/OUTPUT,FILE,EXT	ALL	WRITE OUTPUT TO FILE (FILE.EXT)
/POST1	BEGIN	PLACES USER IN POST1 MODULE
/PREP7	BEGIN	PLACES USER IN PREP7 MODULE
PROC	PREP7	PREPROCESSING, STEP-BY-STEP
PROC,#	PREP7	INFO' ON STEP # IN PROCEDURE

COMMAND	ROUTINE	COMMAND'S RESULTS
SAVE	ALL	SAVE CURRENT WORK FOR RESUME
STAT	ALL	DISPLAYS STATUS OF CURRENT MODULE
/T	ALL	DISPLAY FULL SCREEN TEXT (DIALOG)

ANSYS CONTROL COMMANDS

KEY	COMMAND	ROUTINE	COMMAND'S RESULTS
<F1>	RECALL LAST COMMAND	ALL	RECALLS LAST COMMAND FOR REUSE
^C	<CONTROL>	ALL	DEPRESS ^C TOGETHER WITH "KEY"
^C O	OUTPUT ON/OFF	ALL	TOGGLES OUTPUT ON & OFF
^C P	STOP DISPLAY	ALL	STOPS TRANSMISSION OF GRAPHICS
^C Q	RESTART DIALOG	ALL	RESTARTS STOPPED TEXT TRANSMISSION
^C S	STOP DIALOG	ALL	STOPS (HOLDS) TEXT TRANSMISSION
^C T	SYSTEM STATUS	ALL	DISPLAYS COMPUTER SYSTEM'S STATUS
^C X	CANCEL INPUT	ALL	CANCELS INPUTS NOT RECEIVED BY CPU
^C Y	BREAK OUT	ALL	LAST RESORT BREAK OUT OF I-DEAS

APPENDIX III

UNIVERSAL FILES

In addition to the providing use of IGES formatted files for the transfer of data into and out of I-DEAS, I-DEAS also provides its own transfer file format, call Universal Files. These files are used for the transfer of data from one I-DEAS family to another. For example, from Solid Modeling to Drafting, or from Solid Modeling to Engineering Analysis. Universal files can also be used to: transfer entities from one I-DEAS model file to another; archive complete model file data bases; or to store individual solid models for use in other model files. Universal files are upwardly compatible with higher revisions in I-DEAS software than just the revision used to created the entities (e.g., I-DEAS 4.1 Universal files can be read into I-DEAS V). This make Universal files ideal for creating permanent files for archiving model files, or for creating parts libraries of solid models, etc.

The dimensional data stored in I-DEAS model files is always stored in SI units. If the user specifies another unit system (e.g., IN), the entities' units are converted to the specified units system for display, calculations, etc. In order to maintain the highest accuracy of the data involved, before universal files are written out, the active unit system should be set to SI. This will prevent the data from being converted from SI to another system for storage in the Universal file.

Universal files are more "compact" in size than model files. One Universal file can contain all of the original (i.e., non-standard) entities from one Family of data within a model file (i.e., MODEL.FILE.-.MF1 and MODEL-FILE.MF2 for Solid Modeling or Engineering Analysis), and its size will be less than the total of the model files. This is partially due to the fact that the Universal files written out of a model file will not contained any of the entities that have been saved in I-DEAS "safety nets". That is, "OLD#" entities that have been saved in the model files, due to their possible destruction or loss while that entity type's OLD_STACK_LENGTH in set greater than zero, will not be written into a Universal file. As introduced in Chapter II, it is this "compact" quality of the Universal files which the author has used "to 'deflate' the size of the [model] files, to the minimum size needed".

When the size of the data base for this thesis got too large, it caused problems, including lack of free disk space and long time periods required to retrieve and store the model files. Thus, by: 1) deleting all unneeded entities from the existing model files; 2) writing out a Universal file containing all of the entities in Solid Modeling; 3) reading the Universal file into a new (empty) model file data base; and 4) deleting the original model files, the model files could be compressed to the minimum size possible.

Universal files can be used to read and write entities into and out of a model file with use of the MANAGE, READ/WRITE, UNIVERSAL commands located in any of I-DEAS Solid Modeling's Tasks.

The WRITE UNIVERSAL file command will prompt the user for a Universal File name, and then for confirmation that a new file is to be written, or that an existing file is to be over written. The program will then ask the user to select which entities are to be written, from those used in the active program Family. In Solid Modeling these choices include: ALL, ALL_OBJECT_MODELING, ALL_SYSTEM_ASSEMBLY, VIEWS_3D, DISPLAY_OPTIONS, LAYOUTS, LIGHT_SOURCES, COLORS, WORKING_SETS, PROFILES, SKIN_GROUPS, OBJECTS, COMPONENTS, and SYSTEMS. If the user selects any other than "ALL", the user will be further prompted for selection of the specific entities to be written.

The author wrote out Universal files containing ALL entities from Solid modeling, and then read them into new model files, several times during the completion of this thesis. This allowed the data base's size to be compressed. But, once Engineering Analysis was used, the ability to compress the model files in this way became less successful. When the data from two different I-DEAS Families (e.g., Solid Modeling and Engineering Analysis) were written out to separate Universal files, and then read into a common data base, some entities were duplicated (e.g., WORKING_SETs). This did not always yield a net reduction in disk space required to store the model files.

APPENDIX IV

ANSYS INTERACTIVE COMMAND FILE

The following is a listing of the command file used to call up ANSYS for interactive use for this project (see Chapter XI):

```
$!********************************************************************
$!*** ANSYS_INTER.COM                                              ***
$!*** DESCRIPTION.... ANSYS INTERACTIVE COMMAND FILE               ***
$!*** PURPOSE........ DEFINES LOGICAL NAMES FOR ANSYS LEVEL 4.3A,  ***
$!***                 ACCESSES THE PROGRAM FOR INTERACTIVE USE     ***
$!***                 AND DELETES UNNEEDED FILES                   ***
$!*** CREATED BY..... DR. MIKE DAY, UNIVERSITY OF LOUISVILLE, 1988 ***
$!********************************************************************
$!*** DESCRIPTION OF PROMPTS ***
$! 1. ENTER MODEL_NAME:
$!    Files will be specified as MODEL_NAME.Fnn, where nn is the file
$!    number (from 02 to 40).
$! 2. ENTER /INT TO BEGIN RUNNING ANSYS
$!    The ANSYS program needs to know that it is to be ran
$!    interactively.  Either  /INT or /INT,NO (the NO stands
$!    for no beginning claimer) is appropriate.
$! 3. DELETE ALL FILES EXCEPT 12-POST & 16-PREP: Y/N/P? (Y)
$!    Only FILE12 (Results for Post-Processing) and FILE16 (Model Data
$!    for Pre-Processing) normally need to be saved.  Entering Y will
$!    delete all files MODEL_NAME.F*, excluding 12 & 16; entering N
$!    will result in all files being saved.  Entering P will prompt
$!    for files to delete.
$!********************************************************************
$!*** INITIALIZE ***
$     SET NOON
$     ON CONTROL_Y THEN GOTO FINISH
$     ESC[0,7]  = %D27
$     PAGE[0,7] = %D12
$     WRITE SYS$OUTPUT ESC+"%!0"+ESC+"LZ"   ! CLEARS SCREEN
$!********************************************************************
$!*** ASSIGN LOGICAL NAMES TO FILES FROM ANSYS DISK ***
$     ASSIGN/NOLOG ANSYS$DISK:[ANSYS4.3A]ANS_IDENT.DAT ANS_IDENT
$     ASSIGN/NOLOG ANSYS$DISK:[ANSYS4.3A]DTDOCU.43A ANS_DTDOCU43A
$!********************************************************************
$!*** ASSIGN MODEL NAME TO FILES ***
$     WRITE SYS$OUTPUT "************************************
$     WRITE SYS$OUTPUT "*** ANSYS 4.3A: INTERACTIVE MODE ***
$     WRITE SYS$OUTPUT "************************************
$     WRITE SYS$OUTPUT ""
$ MODEL:
$     INQUIRE NAME "--->" ENTER MODEL_NAME"
$     IF NAME .EQS. ""  THEN GOTO MODEL
$     I = 2
$ ASSIGN_LOOP:
$     IF I .LT. 10  THEN ASSIGN/USER_MODE/NOLOG 'NAME'.F0'I' FILE0'I'
$     IF I .GE. 10  THEN ASSIGN/USER_MODE/NOLOG 'NAME'.F'I'  FILE'I'
$ SKIP_ASSIGN:
$     I = I + 1
$     IF I .GT. 40  THEN GOTO RUN_ANSYS
$     GOTO ASSIGN_LOOP
$!***********************************************************************
$!*** RUN ANSYS ***
$ RUN_ANSYS:
$     WRITE SYS$OUTPUT ""
$     WRITE SYS$OUTPUT "---> ENTER /INT TO BEGIN RUNNING ANSYS"
$     DEFINE/USER_MODE SYS$INPUT SYS$COMMAND
$     RUN ANSYS$DISK:[ANSYS4.3A]ANSYS43A.EXE
$!***********************************************************************
$!*** DELETE SELECTED FILES ***
$     DIRECTORY/SIZE 'NAME'.F*
$     WRITE SYS$OUTPUT ""
$  INQUIRE OK "---> DELETE ALL FILES EXCEPT 12-POST & 16-PREP: Y/N/P? (Y)"
$     IF OK .EQS. "" THEN OK = "Y"
$     IF OK THEN GOTO DELETE_FILES
$     IF OK .EQS. "P" THEN GOTO DELETE_PROMPT
$     GOTO FINISH
$  DELETE_FILES:
$     DELETE 'NAME'.F*;*/EXCLUDE=(*.F12,*.F16)
$     GOTO FINISH
$  DELETE_PROMPT:
$     DELETE/CONFIRM 'NAME'.F*;*
$!********************************************************************
$!*** EXIT ***
$ FINISH:
$     WRITE SYS$OUTPUT ESC+"LZ"      ! CLEAR SCREEN
$     WRITE SYS$OUTPUT ESC+PAGE      ! CLEAR GRAPHICS
$     WRITE SYS$OUTPUT ""
$     WRITE SYS$OUTPUT "*******************************************"
$     WRITE SYS$OUTPUT "*** END OF ANSYS 4.3A: INTERACTIVE MODE ***"
$     WRITE SYS$OUTPUT "*******************************************"
$     WRITE SYS$OUTPUT ESC+"%!1"     ! CODE ANSI
$     EXIT
```

APPENDIX V

ANSYS BATCH COMMAND FILE

The following is a listing of the command file used to call up ANSYS for batch use for this project (see Chapter XI):

```
$!********************************************************************
$!*** ANSYS_BATCH.COM                                              ***
$!*** DESCRIPTION.... ANSYS BATCH COMMAND FILE                     ***
$!*** PURPOSE........ DEFINES LOGICAL NAMES FOR ANSYS LEVEL 4.3A,  ***
$!***                 CREATES BATCH COMMAND FILE, AND SUBMITS      ***
$!***                 JOB TO BATCH QUEUE                           ***
$!*** CREATED BY..... DR. MIKE DAY, UNIVERSITY OF LOUISVILLE, 1988 ***
$!********************************************************************
$!*** DESCRIPTION OF PROMPTS ***
$! 1. ENTER MODEL_NAME:
$!    Files will be specified as MODEL_NAME.Fnn, where nn is the file
$!    number (from 02 to 40).  The file MODEL_NAME.DAT will be used as
$!    input for the ANSYS run.  If MODEL_NAME.DAT does not exist, it
$!    is assumed that MODEL_NAME.F27 has already been created, and a
$!    MODEL_NAME.DAT file is written (/INPUT,27  FINISH  /EOF).
$! 2. SELECT NODE: 2/3/4/5/6 (3)
$!     Use Node 3 if it is not too crowded. It is a VAX 8700 and is
$!     about 2x faster than nodes 5 and 6 (VAX 3100's).
$! 3.  SYS<5MIN  SHORT<15MIN  MEDIUM<1HR  LONG>1HR
$!     SELECT BATCH QUEUE: SYS/SHORT/MEDIUM/LONG (SYS)
$!     These time limits are for Node 3; the limits for nodes 5 and 6
$!     are (15 min, 1 hr, 4 hr, and no limit).
$! 4.  ENTER TIME TO RUN JOB: NOW/TONIGHT (NOW)
$!     Normally the job is ran NOW.  If the job is large and is not
$!     to be checked on until tomorrow, run it TONIGHT (it will start
$!     automatically at midnight) when nobody is using the computers.
$! 5.  DELETE ALL FILES EXCEPT 12-POST & 16-PREP: Y/N? (Y)
$!     Only FILE12 (Results for Post-Processing) and FILE16 (Model Data
$!     for Pre-Processing) normally need to be saved.  Entering Y will
$!     delete all files MODEL_NAME.F*, excluding 12 16 & 19; entering N
$!     will result in all files being saved.
$! *** NOTE ***
$!     The symbols P1 thru P5 (corresponding to the 5 prompts) can be
$!     defined when accessing this command process.  As an example:
$!     @ANSYS_BATCH MODEL 3 SYS NOW Y
$!***********************************************************************
$!*** INITIALIZE ***
$      SET NOON
$      ON CONTROL_Y THEN GOTO FINISH
$      ESC[0,7]  = %D27
$      WRITE SYS$OUTPUT ESC+"%!0"+ESC+"LZ"+ESC+"%!1"     ! CLEARS SCREEN
$!***********************************************************************
```

```
$!*** ASSIGN MODEL NAME FOR INPUT FILES ***
$       WRITE SYS$OUTPUT "******************************"
$       WRITE SYS$OUTPUT "*** ANSYS 4.3A: BATCH MODE ***"
$       WRITE SYS$OUTPUT "******************************"
$       WRITE SYS$OUTPUT ""
$ MODEL:
$       NAME = P1
$       IF NAME .EQS. "" THEN INQUIRE NAME "---> ENTER MODEL_NAME"
$       IF NAME .EQS. "" THEN GOTO MODEL
$       WRITE SYS$OUTPUT "     *** ''NAME'.DAT WILL BE USED FOR INPUT"
$       IF F$SEARCH(NAME+".DAT") .NES. "" THEN GOTO GET_NODE
$       WRITE SYS$OUTPUT "     *** ''NAME'.DAT WAS CREATED
$       OPEN/WRITE MODELDAT 'NAME'.DAT
$       WRITE MODELDAT "C*** STANDARD INPUT FILE FOR BATCH ***"
$       WRITE MODELDAT "/INPUT,27"
$       WRITE MODELDAT "FINISH"
$       WRITE MODELDAT "/EOF"
$       CLOSE MODELDAT
$!********************************************************************
$!*** SELECT NODE TO RUN BATCH JOB ***
$ GET_NODE:
$       NODE = P2
$       IF NODE .NES. "" THEN GOTO SET_NODE
$       WRITE SYS$OUTPUT ""
$       SHOW QUEUE/ALL AP35V3*BATCH
$       WRITE SYS$OUTPUT ""
$       INQUIRE NODE "---> SELECT NODE: 1/2/3/4/5 (3)"
$       IF NODE .EQS. "" THEN NODE = "3"
$ SET_NODE:
$       QUE = "QUEUE=AP35V"+NODE
$!**********************************************************************
$!*** SELECT QUEUE TO RUN BATCH JOB ***
$ GET_QUE:
$       Q = P3
$       IF Q .NES. "" THEN GOTO SET_QUE
$       WRITE SYS$OUTPUT ""
$       WRITE SYS$OUTPUT "SYS<5MIN  SHORT<15MIN  MEDIUM<1HR  LONG>1HR"
$       INQUIRE Q "---> SELECT BATCH QUEUE: SYS/SHORT/MEDIUM/LONG (SYS)"
$       IF Q .EQS. "" THEN Q = "SYS"
$ SET_QUE:
$       QUE = QUE+"_"+Q+"$BATCH"
$!**********************************************************************
$!*** SELECT TIME TO RUN BATCH JOB ***
$ GET_TIME:
$       WRITE SYS$OUTPUT ""
$       WHEN = P4
$       IF WHEN .NES. "" THEN GOTO SET_TIME
$       INQUIRE WHEN "---> ENTER TIME TO RUN JOB: NOW/TONIGHT (NOW)"
$ SET_TIME:
$       IF        WHEN THEN TIME = "AFTER=TOMORROW"
$       IF .NOT. WHEN THEN TIME = "AFTER=TODAY"
$!**********************************************************************
$!*** CHECK FOR DELETION OF FILES ***
$       WRITE SYS$OUTPUT ""
$       DEL = P5
$       IF DEL .NES. "" THEN GOTO CREATE_COM
$       INQUIRE DEL "DELETE ALL FILES EXCEPT 12-POST, 16-PREP &
                    19-ERRORS:(Y)/N?"
$       IF DEL .EQS. "" THEN DEL = "Y"
$!**********************************************************************
$!*** CREATE BATCH COMMAND FILE ***
$ CREATE_COM:
$       DEFDIR = F$ENVIRONMENT("DEFAULT")
$       OPEN/WRITE COM 'NAME'.COM
$       WRITE COM "$ SET NOVERIFY"
$       WRITE COM "$ SHOW TIME"
$       WRITE COM "$ SET NOON"
$       WRITE COM "$ SET MESSAGE/NOFACILITY/NOIDENTIFICATION/NOSEVERITY"
$       WRITE COM "$ ASSIGN/NOLOG ANSYS$DISK:[ANSYS4.3A]ANS_IDENT.DAT
                 ANS_IDENT"
$       WRITE COM "$ ASSIGN/NOLOG ANSYS$DISK:[ANSYS4.3A]DTDOCU.43A
                 ANS_DTDOCU43A"
$       WRITE COM "$ SET DEFAULT ''DEFDIR'"
$       WRITE COM "$ ASSIGN/USER_MODE ''NAME'.DAT FOR005"
$       WRITE COM "$ ASSIGN/USER_MODE ''NAME'.LIS FOR006"
$       I = 2
$ ASSIGN_LOOP:
$       IF I.LT.10 THEN WRITE COM "$ ASSIGN/NOLOG ''NAME'.F0''I'
                                  FILE0''I'"
$       IF I.GE.10 THEN WRITE COM "$ ASSIGN/NOLOG ''NAME'.F''I'  FILE''I'"
$       I = I + 1
$       IF I.LE.40  THEN GOTO ASSIGN_LOOP
$ ASSIGN_SCRATCH:
$       IF .NOT. DEL THEN GOTO RUN_ANSYS
$       WRITE COM "$ DEASSIGN FILE02"
$       WRITE COM "$ DEASSIGN FILE03"
$       WRITE COM "$ DEASSIGN FILE11"
$       WRITE COM "$ DEASSIGN FILE30"
$       WRITE COM "$ ASSIGN SYS$SCRATCH:FILE02''PID'''USER'.DAT FILE02"
$       WRITE COM "$ ASSIGN SYS$SCRATCH:FILE03''PID'''USER'.DAT FILE03"
$       WRITE COM "$ ASSIGN SYS$SCRATCH:FILE11''PID'''USER'.DAT FILE11"
$       WRITE COM "$ ASSIGN SYS$SCRATCH:FILE30''PID'''USER'.DAT FILE30"
$ RUN_ANSYS:
$       WRITE COM "$ RUN ANSYS$DISK:[ANSYS4.3A]ANSYS43A.EXE"
$       WRITE COM "$ TYPE FILE19"
$!*******************************************************************
$!*** DELETE SELECTED FILES ***
$       IF .NOT. DEL THEN GOTO SUBMIT_JOB
$       WRITE COM "$ DELETE/LOG SYS$SCRATCH:FILE%%''PID'''USER'.DAT;*"
$       WRITE COM "$ DELETE/LOG ''NAME'.F*;*/EXCLUDE=(*.F12,*.F16,*.F19)"
$!*******************************************************************
$!*** SUBMIT ANSYS JOB TO BATCH ***
$ SUBMIT_JOB:
$       CLOSE COM
$       LOGFILE = DEFDIR+NAME+".LOG"
$       LOG = "NOPRINTER/LOG_FILE=''LOGFILE'"
$       SUBMIT/NOTIFY/'TIME'/'LOG'/'QUE' 'NAME'.COM
$!*******************************************************************
```

```
$!*** EXIT ***
$ FINISH:
$     WRITE SYS$OUTPUT ""
$     WRITE SYS$OUTPUT "**************************************"
$     WRITE SYS$OUTPUT "*** END OF ANSYS 4.3A: BATCH MODE ***"
$     WRITE SYS$OUTPUT "**************************************"
$     IF F$LOGICAL("MODELDAT").NES."" THEN CLOSE MODELDAT
$     IF F$LOGICAL("COM")     .NES."" THEN CLOSE COM
$     EXIT
```

APPENDIX VI

PLATE LOAD STRESS RESULTS

```
SDRC  I-DEAS 4.1:  Pre/Post Processing                4-JUL-91  02:32:56
OLD SUPT W/ GAPPED PLATE LOAD - J.A.KERN

Group ID           : SUPPORT'S NODES & ELEMENTS
Analysis Dataset   : 7 - LOAD 1, STRESS7
Report Type        : Contour                  Units          : IN
Dataset Type       : Stress                   Load Set       : 1
Frame of Reference: Global                    Data Component: Max Prin
Surface Type       : Middle

           Max Prin   Mid Prin   Min Prin   Max Shear  Von Mises
                730        253         48         50         50
Maximum   3.848E+03  2.353E+01  1.042E+03  4.406E+03  8.076E+03
                713        210         50          1          1
Minimum  -2.596E+03 -2.368E+01 -7.049E+03  0.000E+00  0.000E+00

Average   1.641E+02  1.818E-03 -1.700E+02  1.670E+02  3.315E+02
```

```
SDRC  I-DEAS 4.1:  Pre/Post Processing                 4-JUL-91  01:22:10
NEW SUPT W/ GAPPED PLATE LOAD - J.A.KERN

Group ID           : SUPPORT'S ELEMENTS
Analysis Dataset   : 7 - LOAD 1, STRESS7
Report Type        : Contour                      Units          : IN
Dataset Type       : Stress                       Load Set       : 1
Frame of Reference: Global                        Data Component: Mid Prin
Surface Type       : Middle

          Max Prin   Mid Prin   Min Prin   Max Shear  Von Mises
               411        230          6          23         23
Maximum   3.976E+03  1.298E+02  1.149E+03  4.445E+03  8.147E+03
                35        347         23          15         15
Minimum  -3.022E+03 -3.446E+02 -7.110E+03  9.754E-01  5.028E+00

Average   6.663E+02 -1.044E+00 -6.927E+02  6.795E+02  1.357E+03
```

APPENDIX VII

GENERAL ACCESS TO AND USE OF I-DEAS

ENTERING I-DEAS

At GEA, I-DEAS is available on various VAXs and workstations in the VAX Cluster. To access I-DEAS, the user must first log onto an appropriate VAX. I-DEAS may not be installed on all VAXs, and some installations of the software may not contain all of the I-DEAS Families (e.g., DRAFTING). Entering:

WHEREIS IDEAS <RETURN>,

at the VAX system prompt will display a listing of the VAXs on which I-DEAS is available. This will not show which I-DEAS Families are present. That must be learned through use, or by consulting with the VAX systems' I-DEAS software support personnel.

Once logged onto the appropriate VAX entering:

IDEAS <RETURN>,

at the VAX system prompt will call up the program.

I-DEAS VERSION

When multiple versions of the I-DEAS program are available on the GEA VAX computer network, the first prompt the user must respond to is to specif y which version of the software is desired. This is usually requested with a prompting question that list the versions that are available (e.g., 4.1, 5.0, etc.). One of the versions is usually a default for this question, and can be selected by entering a <Return>.

TERMINAL TYPE

The user must specify the terminal type that is to be used. It is assumed that anyone using this text as a handbook has access to an appropriate graphics terminal (at GEA, either a Tektronix, Calma, Lexidata, or DEC terminal or workstation, or a personal computer with EMU-TEK) for use with I-DEAS and ANSYS, and is familiar with its setup and use. Discussion of this type of information is outside the scope of this work. Some comments appear in this text on the use of various types of Tektronix terminals, because of that hardware's available to GEA's engineers and the author's familiarity with it. Other equipment may work equally well.

The terminal type entered is a code that usually represents the physical terminal being used. If a <RETURN> is entered without a code, a listing of available codes will be displayed. Or, if a partial code is entered, a listing of codes for that family of terminals will result. This allows the user to select the appropriate code for the terminal and mouse or tablet being used.

At GEA, the user will likely be using a Tektronix, Calma, or Lexidata terminal with one of the following codes: T4105, T4107, T4205, T4207, T4129, or one of these will an A or B suffix for a tablet or mouse; CDS80X; or LSVDMA, respectively. When a Tektronix terminal is used, several lower level models' codes may be used in place of a higher level unit's terminals code. For example, the codes T4105 or T4107 can be used on a Tektronix 4207 or 4129 terminals, which have a tablet or mouse attached. This effects the type of menus and mouse or tablet pickable codes that will appear on the graphics display.

DRAFTING is the only I-DEAS Family which requires the use of a mouse or tablet (even though the Tektronix terminal may have a joy pad, that works in all other I-DEAS Families). But, if other I-DEAS program Families will be used before entering DRAFTING, a non-mouse code can be used, and the user will be prompted to change the code when entering DRAFTING. Therefore, if Solid Modeling is to be use first, T4107 might be used, and then switched to T4207A when entering DRAFTING. Experience has shown that using the codes which allow mouse or tablet "picking" of menu entries from the screen can make the system response time slower in some installation (e.g., at low baud rates, or when using emulators) than if I-DEAS is only looking for keyboard entries. Also, screen picking of menu selection is more tedious than keyboard entries of the menus' mnemonic codes, and is seldom used. The picking of graphic items will still function as required, and can be done with the joy pad, cursor keys, or mouse depending upon the device or emulator being used.

If a personal computer (PC) is being used with EMU-TEK Tektronix terminal emulation, this switching of codes may greatly effect the response time. Tektronix terminals operate with two different processors at the same time. One controls the scrolling "dialog" (text), and the other controls the graphics display. EMU-TEK works similarly, using the PC's main processor and it graphics controller. The user can switch the text on alone (<Alt>L), and the PC's (higher speed) processor will control the display speed. But, when the graphics are displayed (<Alt>F), the text and graphics are controlled by the graphics card, which generally has a slower display speed. Thus, the user can speed up the response time by toggling between these "Limited" and "Fine" display modes. But, if the T4207A code is used, and I-DEAS is constantly looking for screen picks from a mouse, only the (slower) Fine display mode can be used. Whereas, unless DRAFTING is being used, the T4107 code will allow the user to toggle between Fine and Limited display modes. In any case, when EMU-TEK is used, toggling between these two display modes, and turning the dialog on and off (<Alt>D) will effect the system's speed and its display colors (again due to the common use of the graphics controller). Experimentation with these options will improve user's productive use of this emulator.

MODEL FILE

The next I-DEAS input required after the terminal type is the model file name. I-DEAS stores all of the model information in model files. These files are named NAME.MF1, NAME.MF2, etc., the number of files created depends on the version of I-DEAS being used. When prompted by I-DEAS, the user types in the single word name for the data base files that are to be used. The file name can either be that of an existing set of files or a new name for which new files will be created. If the name entered is not found in the VAX subdirectory through which the user entered I-DEAS, I-DEAS will ask if it is OK to create a new file. This gives the user a chance to say no, and retype the desired name, or to leave I-DEAS and move to the subdirectory containing the model files desired.

During an I-DEAS session all changes will be made in "scratch" files which I-DEAS creates when the system is entered. Initially, these are copies of the existing files, if an existing database is being used. These scratch files will have names like IDEASTXD1.MF1. The user should issue a SAVE command frequently during an I-DEAS session, to save any work that has been done. This copies the scratch file over the selected model files (or a new model file name if desired). By saving frequently, the user will prevent the accidental loss of any desired information and work done. This will also allow the user to experiment with commands which may result in undesired changes to the model and solids. If an undesired result occurs, and if the original entity does not remain in the stored information, the scratch file can be abandoned, and the permanent files can be reloaded with the NEW_FILE command.

Within I-DEAS, STORE commands are used to store entities in the scratch model files. The stored items are not actually saved in the permanent model files until the SAVE command is used. It is very important for the user to use the SAVE command frequently when working in I-DEAS. This will prevent the loss of time and effort in the event that the I-DEAS session is cut short by user or by system errors.

I-DEAS can store entities, that have not been stored with a STOre command, as OLD1, OLD2, etc., before they are destroyed by the execution of an I-DEAS command. This allows the user to recover these "old" entities, even though they were not stored by the user. The number of old entities stored is determined by the "OLD_STACK_HEIGHT" specified by the user in the MODEL_FILE menu. The default height is one. Because of the storage of "old" entities and other I-DEAS information in the scratch model files, some undesired space is added to the permanent files when a SAVE is issued. Also, when items are intentionally deleted from the stored directories within I-DEAS, the size of the scratch (and in turn permanent) files will not be reduced. The result is that the permanent files will be larger than is actually necessary, or useful. This is only a problem if the model files become very large, which will easily occur when a large data base is created, and when disk space becomes limited. It is a little like working inside of a balloon which stretches with each move. The only way that the author has found to "deflate" the size of the files, to the minimum size needed, involves the operations documented in Appendix III.

If a new model file is being created, I-DEAS will prompt the user for a description of the model file. This will appear at the top of the I-DEAS monitor screen, and on any prints made of the screen. This description can be changed from within I-DEAS by selecting the MODEL_-FILE menu and then the DESCRIPTION command. The user will also be prompted to select the desired system of units to be used in the model file. This determines the units of length, force, area, temperature, etc. For this thesis, the BRITISH_GRAV_(MOD) (mnemonic, IN) units system was used. This can be change from within I-DEAS' MODEL_FILE menu.

OPERATION MODE

Following the selection or creation of a model file, I-DEAS prompts the user to select the operation mode. This is either NORMAL_OPERATION or PROGRAM_FILE_OPERATION mode. The PROGRAM_FILE mode will create a program file containing all of the commands the user enters during the I-DEAS session. The NORMAL mode will execute all entered commands without creating a file. The program file is in ASCII format which can be edited. Each line of the file contains a string of I-DEAS command mnemonics that were entered during the session. This file can be used to document the work done in I-DEAS. It can also be called up and run in a subsequent I-DEAS session by executing: MODEL_FILE, PROGRAM_FILE, RUN. If this is done, the previous session will be rerun on the graphic screen. This can be useful for presenting demonstrations of the I-DEAS system, or to present the model which has been created. It can also be used to reconstruct a session's work which may have been lost due to an incorrect exit from I-DEAS, such as: exiting without saving the results; a system "crash"; or being timed out due to inactivity. When this is done, the end of the file should be edited to remove any errors that may have caused the crash or exit, and once the file is run, the SAVE command should be used before continuing.

I-DEAS MENU STRUCTURE

I-DEAS is divided into logical sections. Each of these sections are interconnected through their menus, which are built in a root like fashion. After the mode of operation is selected, the user enters the menu system at the top of this root. Then a "Family" of the system is selected, either Solid Modeling, Engineering Analysis, System Dynamics, Drafting, Numerical Control, or Test Data Analysis. The next level down the root system contains the Family's "Modules". The next level in the root system contains "Tasks". Each Task contains groups of related commands and options menus. Finally, each sub-menu in a task contains commands, which in turn contain further sub-menus that are presented during the execution of the commands.

Unlike the selection of Families and Modules, a default Task is entered following the selection of a Module. If a single Module is specified at the Module selection level, its default Task will be entered. The user would need to input:

```
/TA <RETURN>,
```

if another Task is desired. This will yield a menu from which the desired TASK can be selected. If, the mnemonic is known for the desired Task, that TASK can be reached by typing:

```
/TA MNEMONIC <RETURN>,
```

or back at the Module selection,

```
MODULE-MNEMONIC TASK-MNEMONIC <RETURN>,
```

can be entered to override the MODULE'S default task.

In I-DEAS, a "/" moves the user up to the top level in the active task's branch of the I-DEAS menus' root system. Entering a "!" moves the user up one level in the root system, or back one prompt when executing an I-DEAS command. These can also be used in a command string, like "/TA W", to start the desired command string at the top level of the root system. Entering a "$" in response to any prompt when executing an I-DEAS command will cancel that command.

GLOBAL_MENU AND VIEW MENU OPTIONS

I-DEAS assigns default values for each display command parameter. But, to obtain optimum views of any entity in I-DEAS (e.g., solids, profiles, working geometry, etc.), the commands in the VIEW menu and the view commands in the GLOBAL_MENU (GM <RETURN>) should be understood.

The GLOBAL_MENU contains a list of commands that are available at any position within the I-DEAS menu structure. The commands can be keyed in directly, or

GM <RETURN>

can be entered to display the global menu. On some terminals, the mnemonics for the global commands are also displayed along the right side of the display, and can be picked with the graphics cursor.

Several of the global display commands are programmed into the terminal function keys by I-DEAS upon entering I-DEAS. These commands are shown in TABLE 5.

TABLE 5.

FUNCTION KEY VIEWING COMMANDS

KEY	MNEMONIC	COMMAND
F1	AU	Autoscale
F2	DR	Draw
F3	GD	Global_Display
F4	ZM	Zoom
F5	MG 1.414	Magnify 141.4%
F6	MG 0.707	Magnify 70.7%
F7	RS	Rotate_About_Screen
F8	REDI	Redisplay

The AUTOSCALE commands causes the display to be erased, and the active entities to be redrawn using a magnification that will allow all of them to be displayed. This is most often used to back out of a zoomed in position, allowing the user to see the entire display.

The transmission of the graphical information to the screen can be stopped before it is completed by depressing

<CONTROL> C

simultaneously. This can save a great deal of time spent waiting for the complete display of an undesired view (which is often obvious long before transmission is complete), especially if a low baud rate communications line is being used. All subsequent view commands (e.g., zoom) will work exactly as if the display had not been interrupted.

The DRAW command causes the graphics to be erased, and the display to be drawn using the active view parameters (e.g., magnification, rotations, eye location, etc.). If no changes have been made to the view since the last redraw, only the banner information will be updated. If changes have to been made to the entities, but the view has not been changed, this command will work like a "refresh" operation. But, if the changes are not recognized by the terminal being used, and a refresh is desired (i.e., no view change is desired, but the screen needs to be updated), the use of the Redisplay command (key F8), or

MG 1 <RETURN>

will yield the desired results.

The GLOBAL_DISPLAY command can be used anywhere in the I-DEAS menu structure to execute any type of DISPLAY_OPTION (e.g., HIDDEN_LINE, SHADED_IMAGE, etc.) the user has SET_AS_GLOBAL in the DISPLAY_OPTIONS menu. This allows the more complex DISPLAY_OPTIONS to be setup and then executed at any time, as easily as the simple DRAW command. The use of these more complex DISPLAY_OPTIONS are discussed in the chapter entitled SOLID MODEL DISPLAY.

The ZOOM command allows a selected area of the display to be enlarged for enhanced viewing, or to obtain clear shots at objects while executing "picking" commands. Once the ZOOM command has been issued, the user is prompted to pick the two diagonal corners (i.e., upper-left and lower-right, or lower-left and upper-right) of the rectangular display area that is to fill the graphics screen.

The MAGNIFY command is used to enlarge or reduce the size of the view displayed around the center of the screen. The amount of enlargement or reduction is entered as a fraction of the existing screen size. For example, MG 0.75 will reduce the displayed view by 75%, resulting in a display of a larger amount of the total active entities. Or, MG 1.50 will enlarge the displayed view by 50%, resulting in a display of a smaller amount of the total active entities. I-DEAS programs two preset magnifications into function keys F5 and F6, but the user can manually type in any magnification desired with the MAGNIFY command.

The ROTATE_ABOUT_SCREEN command is used to rotate the view port about the graphics screen's horizontal and vertical axes. This command is similar to the ROTATE_ABOUT_MODEL command in the VIEW menu, except it uses the graphics screen's axes rather than the global 3D axes for the rotation of the view point.

REDISPLAY "refreshes" the graphics display. Thus, the view is not changed. But, the screen is erased and then redrawn.

Each task in I-DEAS' Solid Modeling and Engineering Analysis contain a VIEW menu which can be obtained by entering

/V <RETURN>.

The VIEW menu contains the commands listed in TABLE 6.

TABLE 6.

I-DEAS' VIEW COMMANDS

MNEMONIC	COMMAND
CE	CENTER
E	EYE_DIRECTION
P	PERSPECTIVE
RM	ROTATE_ABOUT_MODEL
SP	SPECIAL_OPERATIONS
V	VIEW_DIRECTION
LO	LOCATE_TARGET_&_EYE
RE	RESET
LA	LAYOUT
AF	AFFECTED_VIEWPORTS
CU	CURRENT_VIEW
STO	STORE
MA	MANAGE_VIEWS
STA	STATUS

The CENTER command allows the user to select a point in the current view to be translated to the center of the display screen. The new center point can either be picked from the screen with the cursor; or it can be keyed in as a fraction of the screen, for which the lower left corner of the screen is 0,0 and the upper right is 1,1.

By specifying an EYE_DIRECTION, the user can key in a location (in the model's coordinates) from which to view the geometry when looking toward the center of the screen.

I-DEAS' can present 3D objects in such a way that they appear to have depth. The PERSPECTIVE command is used to turn this feature on and off. It is also used to customize the appearance of the perspective view by setting the viewing DISTANCE away from the object in the active unit system, or the FIELD_OF_VIEW in degrees.

The ROTATE_ABOUT_MODEL command is used to rotate the user's view point about the model axes.

SPECIAL_OPERATIONS presents a sub-menu of view commands that are used less often. These allow the user to specify a view Angle, or a VIEW_DIAMETER; to place a 3D point from the model space at the center of the screen; to Translate the geometry along the screen axes; and to apply THICKNESS and VOLUME_CLIPPING.

Each of these commands are described in on-screen HELP, as are all the I-DEAS commands. HELP can be get gotten by entering:

/HE <RETURN>,

and then selecting the kind of help that is desired, which includes descriptions of all commands in the CURRENT_MENU and in the GLOBAL_MENU.

The VIEW_DIRECTION command can be used to define vectors for the line of site to the geometry, and the direction of vertical.

LOCATE_TARGET_&_EYE is useful for selecting a point from which to view the geometry and a target point to look at. This is done interactively in 3D model space.

RESET will replace any view parameters the user has modified with their original default values.

The LAYOUT command is used to control the screen format, which can contain multiple viewports. I-DEAS presents a menu of preset layouts containing from one to four viewports that can be selected, or the user can create custom layouts.

If multiple viewports are displayed, any subsequent command which effects the view may prompt the user to specify which viewports are to be effected. But, if the user desires all subsequent commands to effect selected viewports only, the AFFECTED_VIEWPORTS command can be used.

Once a view is created and modified to the user's satisfaction, the STORE command can be used to store the view for later use with any active geometry. This future use of the stored view(s) is done with the CURRENT_VIEW command. The stored views and their names are managed with the commands in the menu call up with the MANAGE_VIEWS command; and STATUS generates a list of the current values of the VIEW options.

MANAGE

Nearly all tasks within the I-DEAS Solid Modeling program's modules contain a MANAGE menu, which can be reached by entering:

/MA <RETURN>.

These menus contain commands used to manage the entities created and manipulated in the individual tasks. They allow the user to DELETE, RENAME, COPY, LIST, specify the OLD_STACK_LENGTH, GET, STORE, and list DIRECTORIES of the entities created within each task. In addition, some tasks' MANAGE menus also provide means for CLEANing, and APPENDing the data stored in the sets created. Others provide for the READing and WRITing of entities into and out of I-DEAS for use with external programs (e.g., ANSYS and Calma).

Solid models, and system assemblies can be stored in user specified "bins". This allows the user to separate the entities created into logical groups to aid in the management of their storage. The MANAGE menus in the appropriate tasks within these modules provide BIN commands used to DECLARE, DELETE, and RENAME bins, and to identify a CURRENT (active) bin. Only those entities contained in the "current" bin can be retrieved. But, it is possible to make a bin current, get an object, and then make another bin current. This will allow the object to be stored in another bin, or to work with one object, along with objects from another bin.

DISPLAY_OPTIONS AND DISPLAY_ATTRIBUTES

In addition to the commands presented in the VIEW menus presented in each of the I-DEAS tasks, which effect the ways in which the various entities can be viewed, there are many different options available for the type of displays which can be viewed. These display options are obtained with the commands presented in the DISPLAY_OPTIONS and DISPLAY_ATTRIBUTES menus . These display options are discussed in detail in Appendix X.

APPENDIX VIII

I-DEAS SOLID MODEL CREATION OPERATIONS

CREATE MENU

The CREATE menu contains the commands need to create both primitives and user defined solids. The primitives (blocks, cones, cylinders, spheres, and tubes) each have a command listed in this menu. These basic solids are not only useful as pieces with which to build larger solid models. They are also very useful as "tools" for cutting other solids. They can be used like drill bits, files, and dies, etc.

CREATION OF PRIMITIVES

The BLOCK command prompts the user for the block's X, Y, and Z dimensions, and its color. The resulting block will be centered at the origin of the global 3D space.

The CONE command asks for the cone's bottom and top radii; its height and the number of facets used for its sides, whether it is OK to cap it with triangles, and for its color. The cone's axis will be along the global Y-axis, with its bottom centered at the origin. If the top or bottom are capped with triangles, a center point will results, which may be useful in orienting or shaping the cone.

The CYLINDER command prompts the user for the cylinder's radius, height, and number of facets, if it should be capped with triangles, and for its color. The cylinder's axis will be the global Y-axis, and its center will be place at the global origin.

The SPHERE command asks for the sphere's radius, the number of facets about the Z-axis, the number of facets about the Y-axis, and for its color. The center of the sphere will be place at the global origin.

The TUBE command will prompt the user for the tube's inner and outer radii, its height and number of facets, and its color. Its axis will be the global Y-axis, and it will be centered about the origin.

In each of these cases, the number of facets specified will effect the way the object displayed only. Each object will also have a "precise definition" which will consist of planar and rational B-spline surfaces.

CREATION OF USER DEFINED SOLIDS

In addition to the commands used to create primitives, Object Modeling provides EXTRUDE, REVOLVE, SKIN, FILLET_TOOL_@ commands.

EXTRUDE

The EXTRUDE command works like a cookie cutter. It uses any profile that has been previously created and draws it through space a specified distance perpendicular to the profile's plane, twisting as it goes if desired, to cutout an object.

The EXTRUDE command will first prompt the user for the profile to be used. Next it will ask for the extrusion distance. Then it will ask for the total twist angle, with a default of zero for a straight part.

If a non-zero twist angle is specified, it will ask for the point about which to pivot the profile. The default is the profile's centroid.

EXTRUDE will then ask for the edge generation option, which is either ABSOLUTE_DEVIATION, or PERCENT_OF_CHORD_LENGTH. These options control how coarse or fine I-DEAS will generate the approximate topology model which is displayed on the graphics screen using faceted planar surfaces. This does not effect the precise geometry which is created with rational B-spline surfaces. The number of facets are generated based on the degree of deviation between the true curves and the facets' linear edges or chords. The default method used for faceting the model is the PERCENT_OF_CHORD_LENGTH option.

The PERCENT_OF_CHORD_LENGTH method is independent of the object's scale. It uses a ratio between the maximum distance from a facet's edge to the true curve and the length of the facet's edge. This causes a larger number of facets to be created in areas of great detail, and less facets are created in areas of low detail. If this option is used to control the faceting, a maximum percentage deviation must be entered. The default value is eleven percent (11%). This percentage will yield sixteen (16) facets around a cylinder, which is also the default for a cylinder created as a primitive.

The ABSOLUTE_DEVIATION option uses a fixed maximum distance (in the current units) from the faceted edges and the true curves. This method is recommend when creating solids of long slender objects using the skinning technique.

In either case (percentage or absolute) higher maximum deviations yield fewer facets, and lower deviations yield more facets.

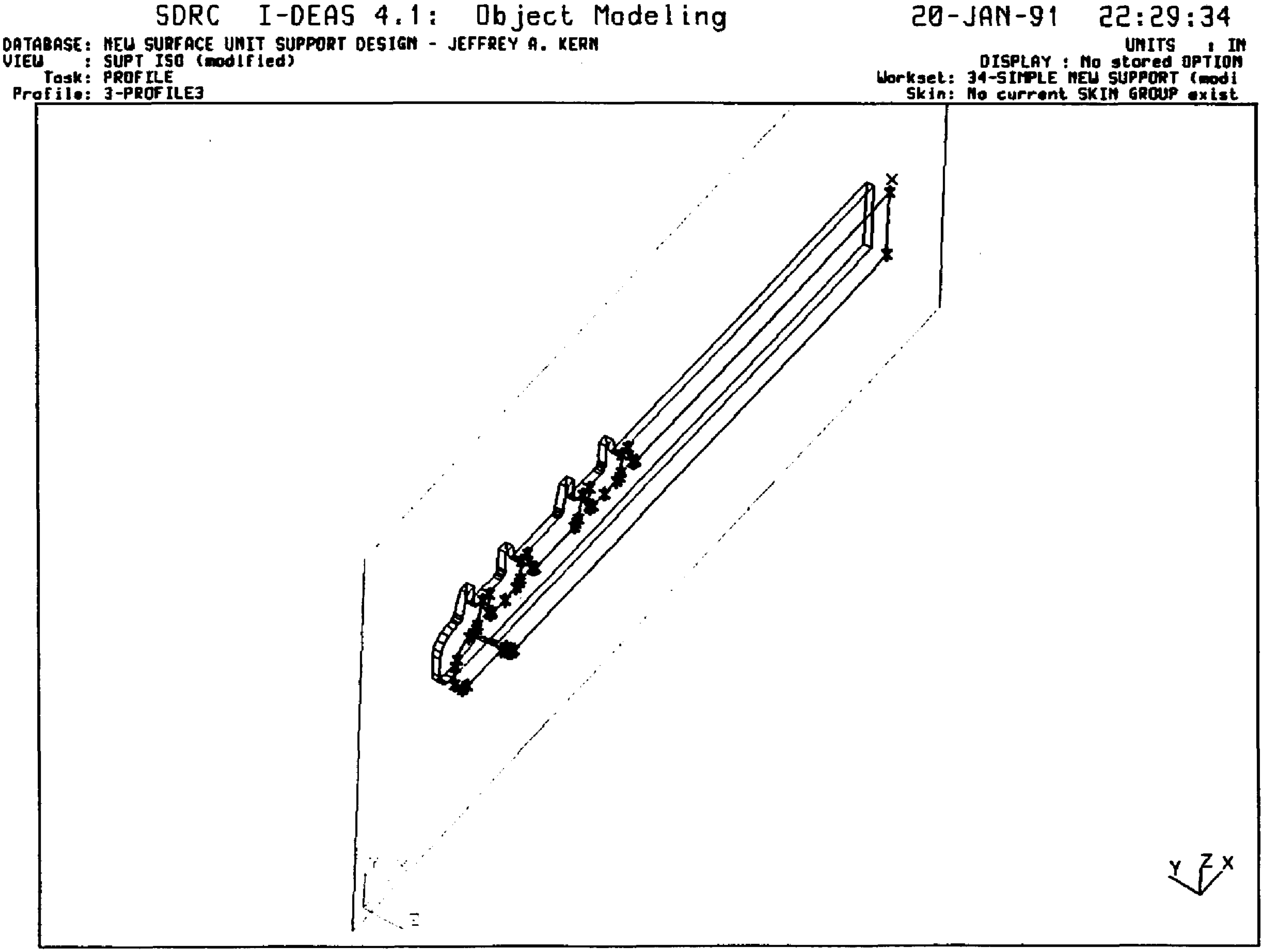
The last two inputs required for the EXTRUDE command are the number of sections and the objects color. The number of sections selected will create duplicates of the profile. If the object is a straight solid, one section is all that is needed. If the profile is being rotated during the extrusion, I-DEAS will select a reasonable number of duplicates for the default value.

Figure 53. displays an example of a 2D profile (shown with the "*" points) which was extruded to create a straight solid. The profile was extruded 0.05", the thickness of the supports, without twisting. The solid has been translated away from the profile for better illustration.

REVOLVE

The REVOLVE command is used to create solids by revolving a 2D profile about the global Y-axis through a specified number of degrees of rotation. Because this operation works around the Y-axis, the profile may require orientation (in the PROFILE_2D task) to place its axis of rotation on the Y-axis. The profile can not cross the Y-axis. But, it can be located away from the Y-axis, as was done to create each section of the heating units' heater sheath, as will be shown later in this chapter. A solid of revolution can be created by revolving a profile of half of the object about its axis of symmetry, lying on the Y-axis. For example, cylinders and cones could be created by rotating rectangles and triangles about the Y-axis.

The REVOLVE command will first prompt the user for the profile to be used, the default is the currently active profile. If the profile crosses the Y-axis, the operation will be terminated with an error message stating that this is the case. Next REVOLVE will ask for the angle (in degrees) through which to rotate the profile about the Y-axis.



SDRC I-DEAS 4.1: Object Modeling 20-JAN-91 22:29:34

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN UNITS : IN

VIEW : SUPT ISO (modified) DISPLAY : No stored OPTION

Task: PROFILE Workset: 34-SIMPLE NEW SUPPORT (modi

Profile: 3-PROFILE3 Skin: No current SKIN GROUP exist

FIGURE 53. - EXAMPLE OF A SOLID CREATED BY EXTRUSION

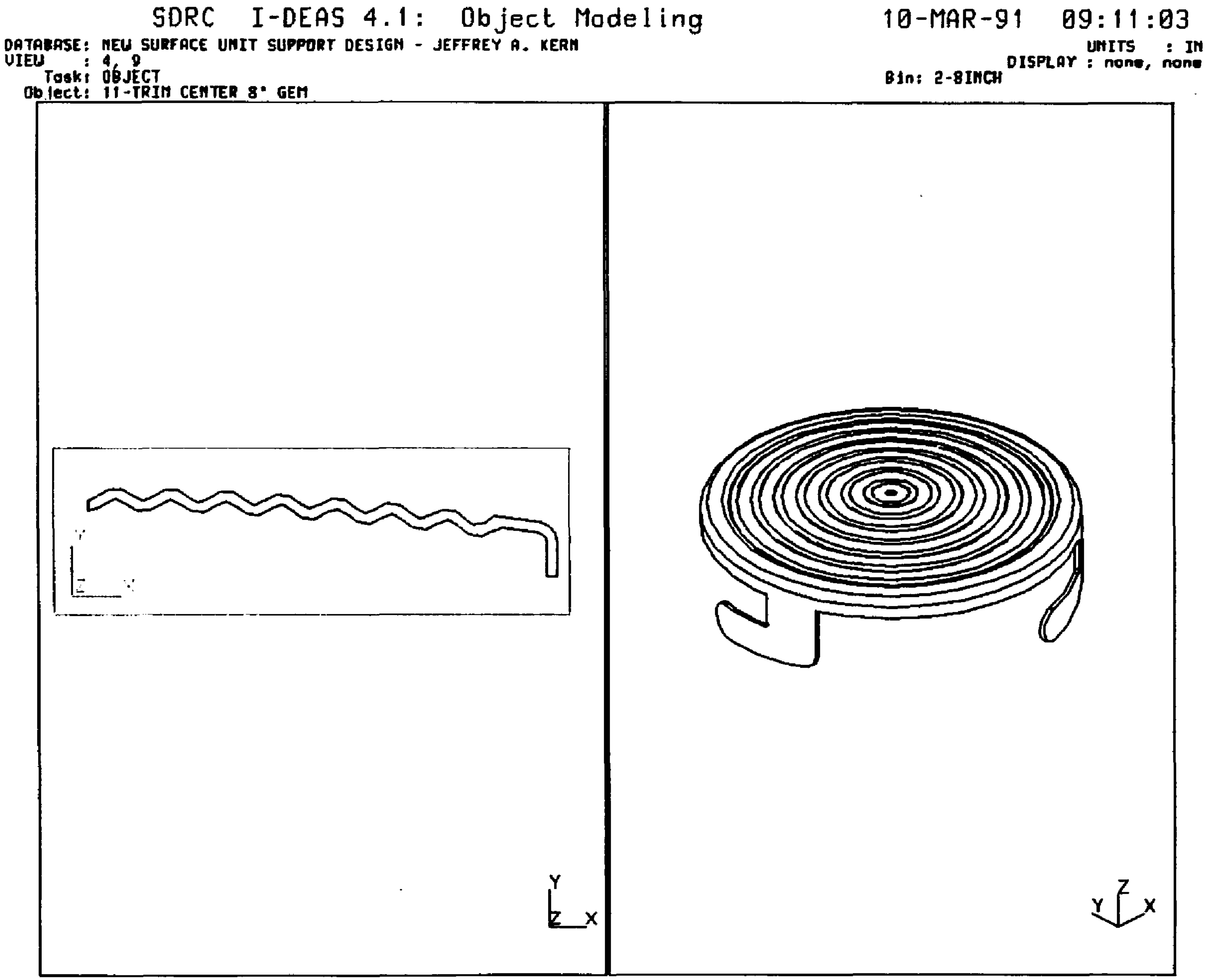
If an angle less that 360° is used, the program will cap the ends of the solids with planes bounded by duplicates of the 2D profile.

The next options that the REVOLVE command presents are: translation along the Y-axis; and change in radius. These can be used to create solids of such things as springs and machine threads. If either of these options is used, the changes are distributed linearly over the angle of revolution. If the profile is located away from the Y-axis, it can spiral inward, toward the axis, with the change in radius option. But, the profile can still not cross the Y-axis. Thus, before the prompt for these options is presented, I-DEAS will display the "Minimum radius change". This value is usually a negative number, and represents the largest negative change in radius that can be used without crossing the Y-axis.

REVOLVE will then ask for the edge generation option (ABSOLUTE_-DEVIATION, or PERCENT_OF_CHORD_LENGTH), which were discussed previously along with the EXTRUDE command.

The last two inputs required for the REVOLVE command are the number of sections to be created, and the object's color. The number selected will create duplicates of the profile at equal angular spacing through the revolution. The solid's faceted representation will be completed by connecting each point around the duplicate profiles to those of the next with chords. The default is sixteen (16) sections per 360° of revolution, like a primitive cylinder.

Several solids were created for this thesis using the REVOLVE command, in combination with the EXTRUDE and JOIN commands. Figure 54. shows one example of a part the was created using the REVOLVE command. It is a Center Trim that is designed to be mounted onto the center of



SDRC I-DEAS 4.1: Object Modeling 10-MAR-91 09:11:03

DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN
VIEW : 4, 9
Task: OBJECT
Object: 11-TRIM CENTER 8" GEM

UNITS : IN
DISPLAY : none, none
Bin: 2-8INCH

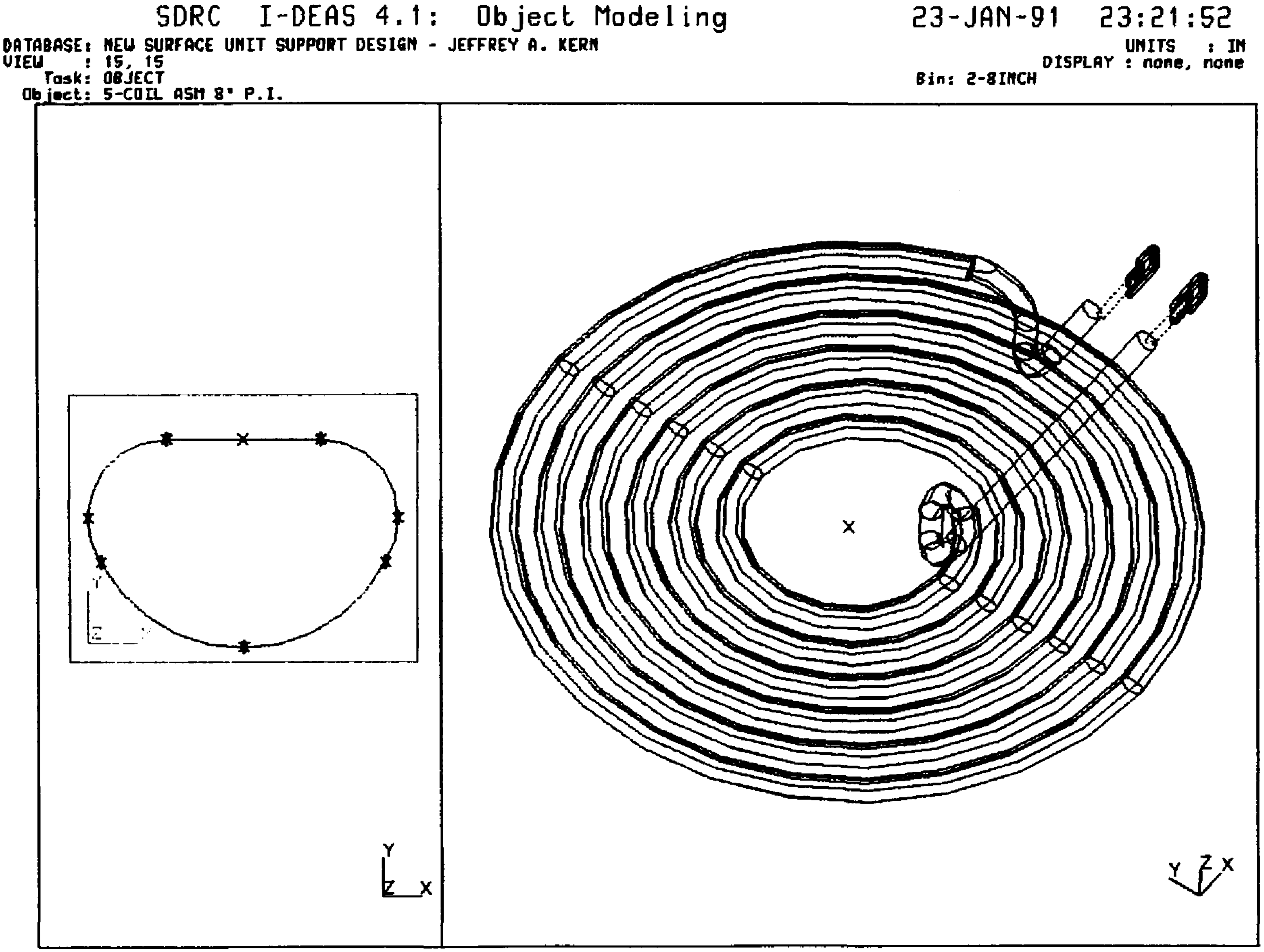
FIGURE 54. - EXAMPLE OF A SOLID CREATED BY REVOLUTION

the new support by twisting the three tabs into holes in the legs of the support, and then bending them over. A bull's eye, or drop-in-a-bucket pattern is stamped into these parts for a replacement parts company supplied by GE. Although it was not analyzed physically or structurally, it did need to be designed, and served as good exercise in the use of a working set, profile, and solid revolution for the creation of a unique part.

Figure 55. displays the complete model that was created for the sheathed heating unit to which the old and new surface unit supports are to be attached. The unit's round sheath is formed into a six turn scroll shape, and then flattened into a flat top D shape. The shape is not actually a mathematical scroll, and is could not be created using the change in radius option within the REVOLVE command. Each of the main semicircular arks and the two end arks of the unit have their own constant radii. Therefore, each of these segments were created with the command, and then oriented and JOINed to construct the complete unit. The joints of each segment can be seen were their faceted surfaces meet, primarily along the global YZ plane.

SKIN

The SKIN command was not used in this thesis. But, it provides a very unique method for creating solids, and its use should be explored for any project requiring the creation of non-geometric solids. Basically, the skinning method of object creation involves creating multiple 2D profiles that are sections of the object taken from known distances along the object's main axis, and then using the I-DEAS program to create faceted surfaces over these sections to create the



```
        SDRC  I-DEAS 4.1:  Object Modeling              23-JAN-91  23:21:52
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN             UNITS   : IN
VIEW    : 15, 15                                                 DISPLAY : none, none
    Task: OBJECT                                           Bin: 2-8INCH
  Object: 5-COIL ASM 8" P.I.
```

FIGURE 55. - HEATING UNIT'S SOLID MODEL (REVOLUTIONS)

body. This works much like creating a model airplane with wooden ribs and tissue paper stretched over the frame. This technique has been used by others in the creation of solid models of bones that have been sectioned at measured distances and digitized to create the profiles, and of body organs that have been sectioned with CAT scan and MRI equipment.

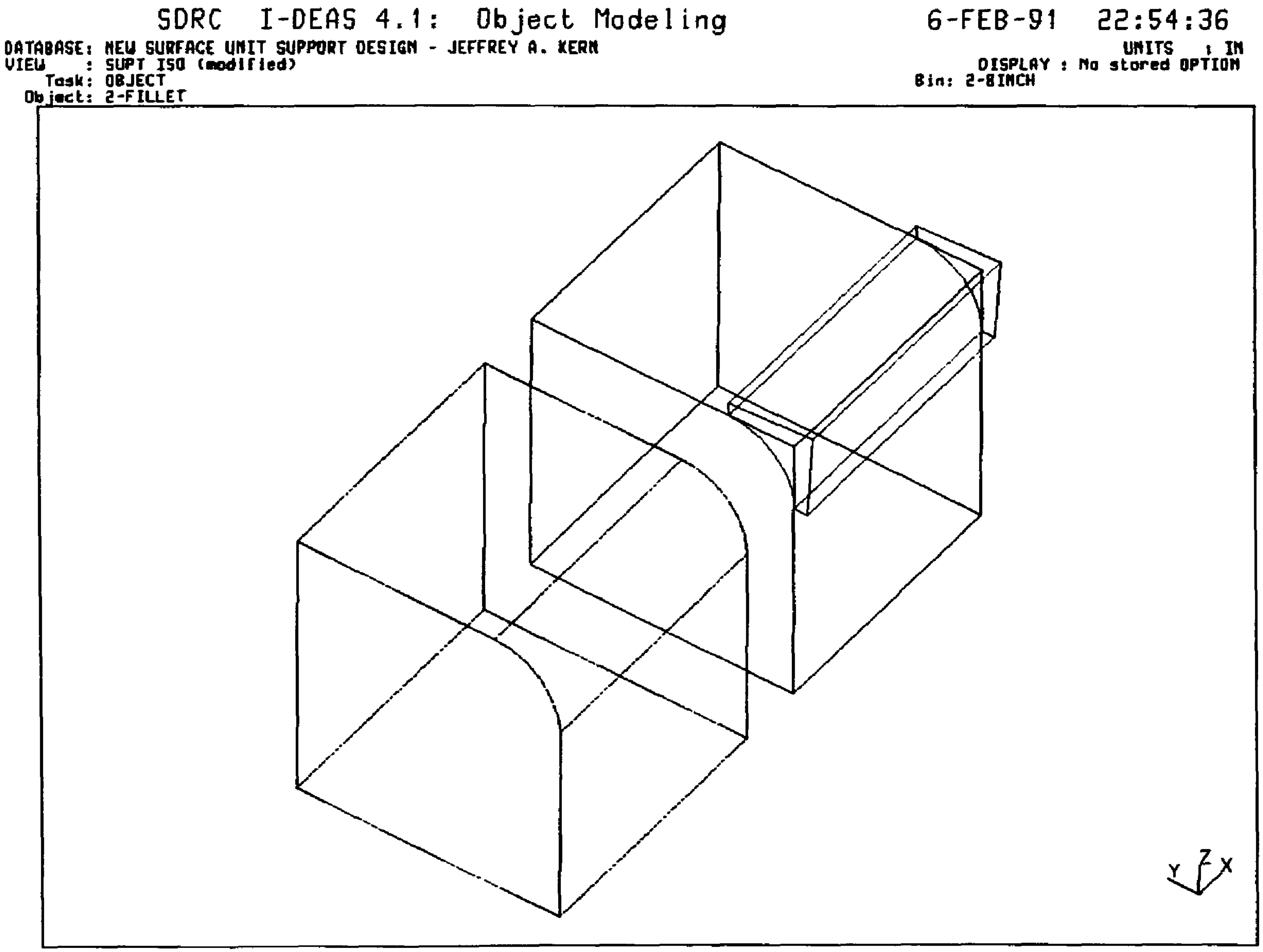
FILLET TOOLS

Object Modeling provides two commands for the creation of fillets: EDGE_FILLET_TOOL_@, and CORNER_FILLET_TOOL_@. These commands, and all other I-DEAS commands that end with an "@", are not tracked in the objects' history file. As their names imply, the commands do not actually change the object for which the fillets are being created, but rather, they create "tools". These tools are separate solids, stored in the active object bin, which can then be cut from or added to other solids. When possible, it is more efficient to fillet objects when creating the objects' working sets of 2D profiles, especially in the case of rounded corners for straight sided parts' edges.

The EDGE_FILLET_TOOL_@ command is used to create tools for use in rounding off the outside edges, or for jointing the inside edges of two adjoining surfaces with a radius. The first items required by the EDGE_FILLET_TOOL_@ command are the selection of the start and end points of the boundary between the two surfaces to be rounded off. Then the user is prompted to input the radius of the fillet at the starting point, followed by the radius at the ending point. If desired, the radii at the two ends can be different, in which case I-DEAS will vary the fillet tool's radius linearly from the start to the end. The default radius for the end point is the radius input for the start point. I-DEAS next asks for the length of the side of the fillet tool which will extend perpendicular to the fillet surface at the tangent line between the fillet and the object's surface being filleted. Next the user is asked for the extension length for the first end of the fillet, followed by the extension of the second end. The extension of the ends of the fillet tool may be required to cut the full length of a solid which has material to be removed with the tool that lies beyond the ends of the fillet edge shared by the two surfaces to be filleted. The edge generation option, and its maximum deviation must be input, as were discussed along with the EXTRUDE command. The final input required for this command is the name desired for storing this solid (fillet tool).

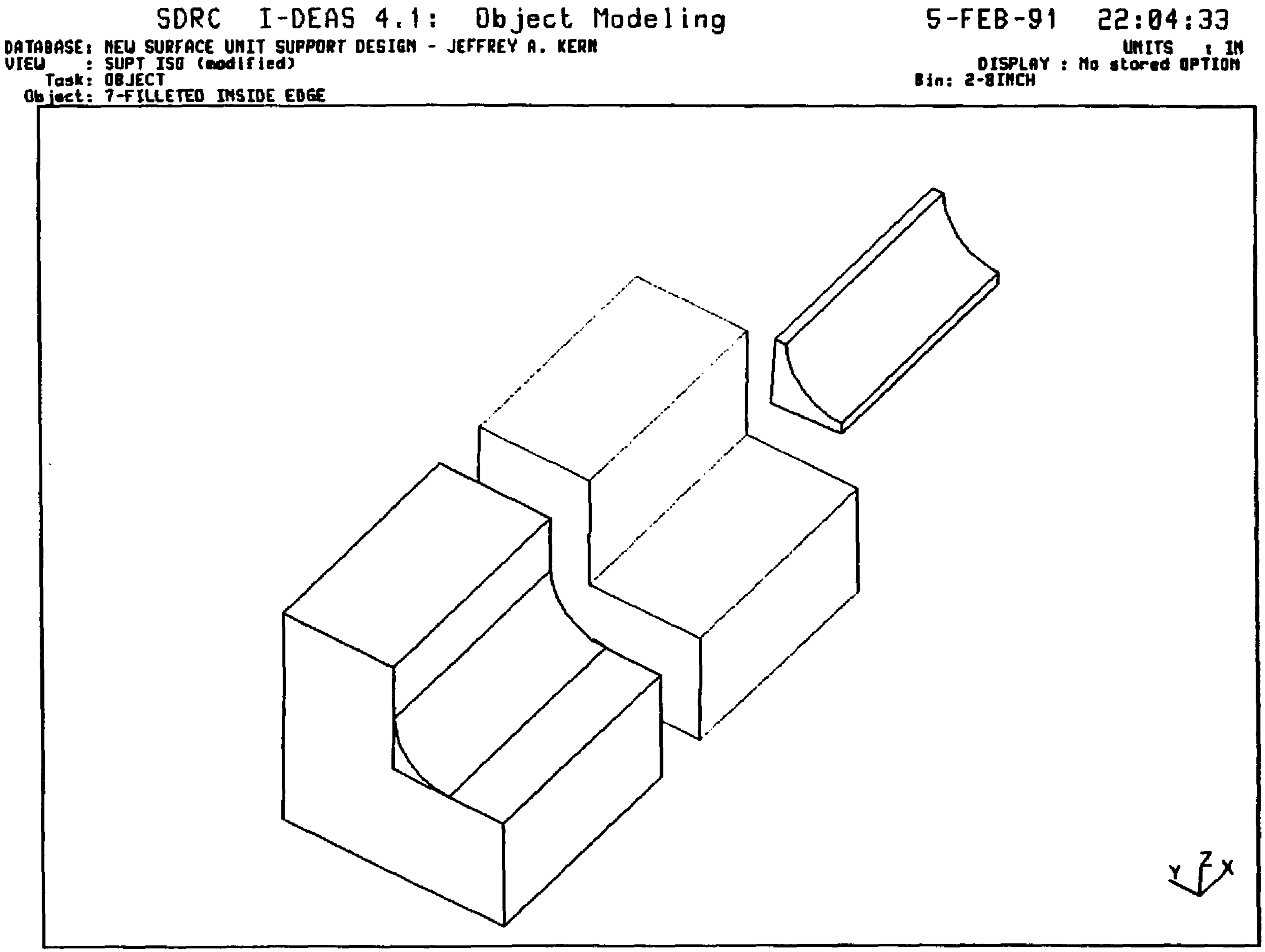
Two examples where created to display the use of the EDGE_FILLET_-TOOL_@ command. Figure 56. shows a one inch square block which has had one outside edge filleted. This figure displays the original block with the fillet tool in position on the block, and the resulting filleted object created by cutting the block with the fillet tool. In this example the fillet radius is constant (0.25") along the entire fillet; the fillet extension is 0.050"; and the end extensions are zero. Figure 57. shows a straight sided object which has an inside edge that was filleted. As in the outside fillet example, the fillet radius is constant and no end extension were required to cut the straight sided object.

The CORNER_FILLET_TOOL_@ command is used to create tools for use in rounding off the outside corner, or for joining together the inside corner of three adjoining equal sized curved faceted surfaces with a



```
        SDRC  I-DEAS 4.1:  Object Modeling                    6-FEB-91  22:54:36
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN              UNITS   : IN
VIEW    : SUPT ISO (modified)                               DISPLAY : No stored OPTION
   Task: OBJECT                                          Bin: 2-8INCH
 Object: 2-FILLET
```

FIGURE 56 - FILLETED OUTSIDE EDGE EXAMPLE



```
          SDRC  I-DEAS 4.1:  Object Modeling                    5-FEB-91  22:04:33
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN                 UNITS   : IN
VIEW    : SUPT ISO (modified)                                   DISPLAY : No stored OPTION
    Task: OBJECT                                              Bin: 2-8INCH
  Object: 7-FILLETED INSIDE EDGE
```

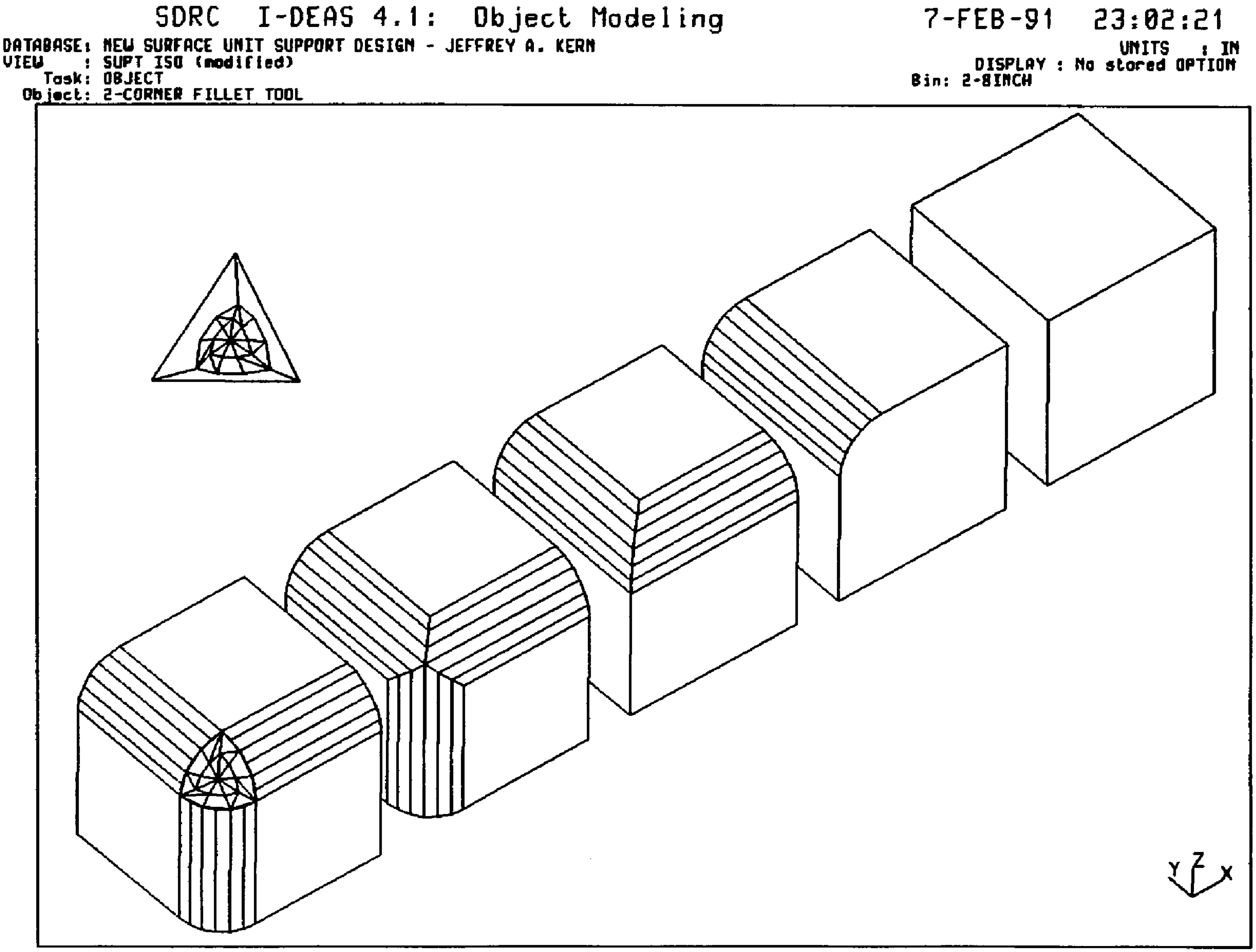
FIGURE 5.7.. - FILLETED INSIDE EDGE EXAMPLE

spherical surface. This command is used to round a corner formed on the outside or inside of three adjoining surfaces which have previously been filleted with the FILLET_TOOL_@ command. When the three pairs of adjoining sides are filleted successively with identical radii, which requires fillet extensions on the corner end of the second and third fillet tools equal in length to the fillet radius, a point will result. It is this point that is rounded with the CORNER_FILLET command. The first input required for this command is the selection of this point. Then the length of the filler tool's side is requested, followed by the edge generation option, and its maximum deviation, the tool's color, and name for storage.

Figure 58. shows an example of how the CORNER_FILLET_TOOL_@ command is used. The five block displayed show the steps required to create the fillet tools and corner fillet tool (rotated and move to the upper left for display), used to cut the block in order to round the one corner.

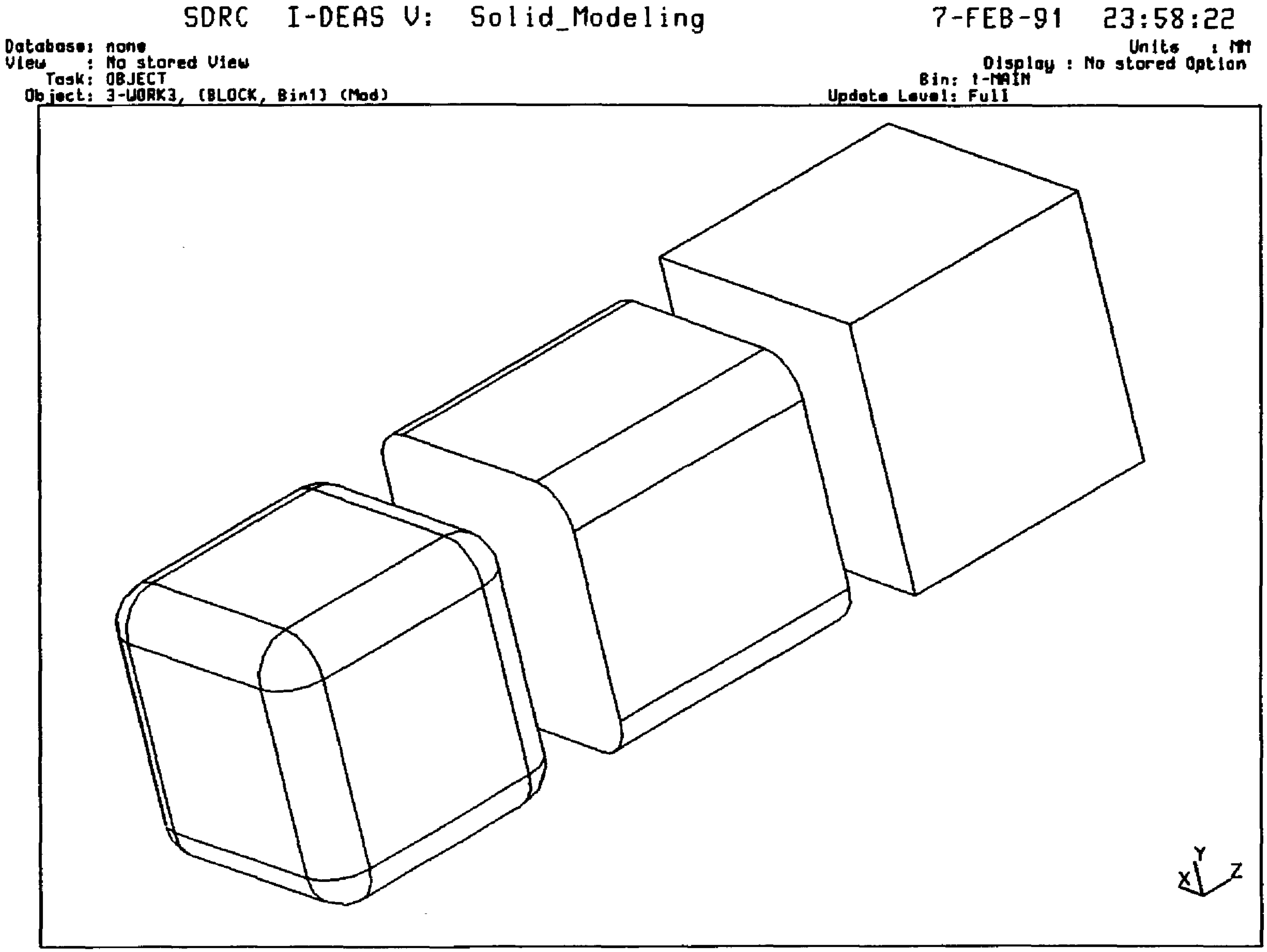
This illustration was made with the "FEATURE LINES" turned on to show the edges of the facets created, and in hidden line format.

This example clearly illustrates how tedious and time consuming filleting (rounding) the edges of a solid model with the commands provided in I-DEAS 4.1 can be. But, in I-DEAS V, a new CONSTRUCT command "ROUND" is provided, which can be much easier to use when rounding the edges of solids with a constant radius. No tools are created, and no cutting or joining operations are required. The user simply selects the edges that are to be rounded, and the radius to be used, and I-DEAS V does the rest. Figure 59. illustrates this.



```
SDRC  I-DEAS 4.1:  Object Modeling                    7-FEB-91  23:02:21
DATABASE: NEW SURFACE UNIT SUPPORT DESIGN - JEFFREY A. KERN          UNITS   : IN
VIEW    : SUPT ISO (modified)                          DISPLAY : No stored OPTION
    Task: OBJECT                                       Bin: 2-8INCH
  Object: 2-CORNER FILLET TOOL
```

FIGURE 58. - FILLETED CORNER EXAMPLE



```
               SDRC  I-DEAS V:  Solid_Modeling                    7-FEB-91  23:58:22
Database: none                                                               Units   : MM
View    : No stored View                                          Display : No stored Option
    Task: OBJECT                                                 Bin: 1-MAIN
  Object: 3-WORK3, (BLOCK, Bin1) (Mod)                        Update Level: Full
```

FIGURE 59. - I-DEAS V ROUNDING EXAMPLE

OBJECT BLANKING

An important feature of the Object Modeling should be discussed is object "BLANKING". Whether an object is made current by creating it, or by retrieving it from a storage bin, it will be visible (as long as it is within the viewport and the display of objects is toggled ON). But, it is possible to also have non-active objects visible along with the currently active object. In I-DEAS terminology, stored objects that are visible are "unblanked".

This can be very useful. Because, the user may be able to more clearly see the effects of changes that have been made to an object, if various versions of the object have been stored. It can also allow the user to see the spacial relationships between objects when orienting the current object. Also, the user can pick the end points on any facet of an unblanked object's facets when measuring distances, or when orienting the current object.

When I-DEAS is initially entered, the defaults are set such that the current object is left unblanked when it is replaced with another object, either by the creation of a new one, or by retrieving a stored one. If the user desires to make additional objects (that are stored in the current storage bin) visible, this can be done with the UNBLANK_-STORED_OBJECT command. If the user wishes to make any "unblanked" objects invisible, this can be accomplished with the BLANK_STORED_OBJECT command. This command will allow the user to pick the visible objects to be blanked with the screen cursor, or by entering their LABELs; or all unblanked objects can be blanked by entering a "*". Objects can be blanked and unblanked by specifying their Label names of numbers. If their labels are unknown, the user can request a Directory of them.

CONSTRUCTION COMMANDS

Object Modeling provides commands which can be used to cut, join, add, or intersect one object with another, to check one object for interference with another, and to cut one object with a plane. The operations all function by checking each facet of the solids for intersection with the other facets. The commands used for modifying objects, or creating new objects create new facets incorporating the lines of intersection found.

In some cases facets from the original objects are combined with new facets, and new objects are created. No new curves are calculated. Rather, the "precise" rational B-spline surfaces from the original object(s) are associated with the new object(s) to form their new "precise" geometries. This can result in facet edges that do not lie exactly along the intersection of the rational B-spline surfaces. But, it is only necessary to correct these errors in cases were the faceted representation doesn't approximate the precise object closely enough, such as when two objects of vastly different size, or with little interference have been operated on with these boolean commands. In these cases it may be necessary to "re-syncronize" the precise and faceted representations of the object. This can be done with the FACET_REFINE and MOVE_TO_PRECISE commands. Some commands, such as SURFACE_BOUNDARY_DISPLAY and WRITE, IGES preform re-syncronization themselves.

If two object's precise models overlap, but their facets do not interfere, the boolean operations may not find any intersections. If this occurs, the models can be refined to a higher level of faceting with the FACET_REFINE command.

CUT COMMAND

The CUT command is used to cut one object with another. At least one on the objects must the stored. The other can be the active object, which may not have been stored. Any volume of the objecting doing the cutting which intersects the object being cut will be removed from the cut object. This can result in breaking the cut object into several bodies. If this occurs, any undesired bodies can be eliminated with the BODY DELETE command.

The first input required for this operation is the selection of the first body (it can cut or do the cutting). The default object is the currently active object. This makes it possible to perform successive boolean operations on objects without storing the resultants from each step. The next input needed is the selection of the second object. The default for this is a name of number of a stored object. For both the first and second objects, a ? can be entered to display the directory of the active object bin, from which the selections can be made. Another option is to pick any unblanked object which appears on the display screen. To use this option, a V is entered to select a Visible object, and then the screen cursor is used to pick a point, face, surface or body of the object desired. After the first and second objects are selected, I-DEAS asks which object is to act as the "cutter", by selecting FIRST_CUT_SECOND, or SECOND_CUT_FIRST. If the user is using a back up of unstored objects (i.e., MANAGE, OLD_STACK_LENGTH greater than zero) and the current object has not been stored, I-DEAS will store it as OLD1 before proceeding, and then complete the operation.

If the two object's colors were not the same, some of the facets of the new object will be of each of these colors. To change them all to one color, or to change the objects color entirely, the COLOR_OF_OBJECT command can be used.

If the resulting object is what the user desired, it should be STOred to prevent its loss, before any further action is taken. In fact, if the operation was very complex and required long computing time, it is a good idea to SAVE the data base after the object is stored. This will prevent its loss in the event the scratch data base is lost, or if the system is exited without having executed a save.

JOIN COMMAND

The JOIN command is used to combine two objects. Any overlapping bodies between the two objects will be combined into a single volume. If there are any separate, non-overlapping bodies involved, they will remain as separate bodies related to the new object created.

The first two inputs required for this operation are the selection of the two objects to be joined. The defaults and options provided for these selections are the same as were described for the CUT_OBJECT command. Then, as was the case with the CUT_OBJECT command, if the current object has not been stored, and the OLD_STACK_LENGTH is greater than zero, I-DEAS will store it as OLD1, and then complete the joining operation. The screen will then be cleared, and the resulting object will be displayed. If desired, the object's color and orientation can be changed, any undesired bodies can be deleted, and then if satisfactory, the object should be stored.

If two objects are to be joined by "welding" them at a planar boundary, they should be oriented to touch exactly, or to overlay by more than the POINT_COINCIDENCE tolerance (set in the MODEL_FILE, TOLERANCE menu) to avoid problems which can arise due to computational inaccuracy. This can easily be accomplished by orienting one object as desired and storing it. Then, the second object can be made gotten and the first object UNBLANKED (or left on the workbench in Rev. V). Next, the ORIENT, MOVE_TO command can be used to pick a point on the second object's surface which is to be joined to the first, and a point on the first object's joining surface. This will translate the second object into exact contact with the first. Then, if necessary, the second object can be rotated about this point to align the two planes to the joined, using the ORIENT, ROTATE command. Finally, if the second object is not intended to be joined at this point, if convenient, the second object can be translated in global, X, Y, Z, coordinates along the jointing plane, with the ORIENT, TRANSLATE command. Or, it can be translated along a user defined vector, a specified distance through the use of the ORIENT, MOVE_ALONG command. This command gives several methods for defining the vector, including keying in DIRECTION_VALUES, and of picking POINT_TO_POINT start and end points for the vector. These orientation techniques are also useful when orienting one object which is to be used to cut another.

INTERSECT COMMAND

The INTERSECT command is used to create a new object which consists of the intersection of two objects, which means that it will be made up of any volume of material that is common to the two objects being intersected. This operation, like the CUT and JOIN commands, is computed based on the interference between the two objects' facets, and not on the intersection of their precise (rational B-spline) models. Thus, it may be necessary to increase the objects' faceted model resolutions with the FACET_REFINE operations, in order to yield the desired results. The inputs and post result operations described for the CUT and JOIN commands apply to this command.

PLANE_CUT COMMAND

The PLANE_CUT command is used to cut an object with a specified 3D plane. The choices for output are: the volume of the object on the POSITIVE_SIDE of the cutting plane; the volume of the object on NEGATIVE_SIDE of the cutting plane; or the object's CROSS_SECTION. The two "sides" of the cutting plane are determined by the direction of the arrows which are displayed on the positive side of the cutting plane.

The first input required is the specification of the cutting plane which is to be used. Several options are presented for this selection, which are: COEFFICIENTS; THREE_POINT; POINT_NORMAL_VECTOR; DISTANCE_-NORMAL_VECTOR; XY_PLANE; YZ_PLANE; XZ_PLANE; SURFACE_PLANE_(EXISTING); OFFSET_SURFACE_(EXISTING); and USE_VIEW_PLANE. If a plane has been defined for a prior PLANE_CUT, its COEFFICIENTS will be the default values presented if the COEFFICIENT option is selected. If one of the global planes (e.g., XY_pane) is selected, its position along the normal axis (e.g., the Z-axis) will be requested. I-DEAS 4.1 requires that this be input numerically, but gives the option to MEASURE the location. This allows the user to use the MEASURE options (including the COORDI-NATES_OF_POINT) to determine the value to be used, and then to key it in. I-DEAS V give an option to pick a point directly.

Once the plane is defined, the user is prompted for the output desired. If either side is selected, the program will cut the object and create a new current object consisting of the solid volume(s) on that side of the plane. If CROSS_SECTION is selected, the user will be prompted to input a name or number for the cross-section to be stored under as a profile. The profile can then be used as any other profile which has been created in the PROFILE task.

The curves in the cross-sections' profiles will be rational B-splines, and not faceted curves. The exactness of the curves yielded are determined by the tolerance which is preset with the MODEL_FILE, TOLERANCE, CROSS_SECTION command. Because of the higher degree of precision, compared to the faceted models, they can be used to measure clearances more accurately than the interferences check with the INTERFERENCE command.

ORIENT MENU

Object Modeling contains a menu devoted to commands which can be used to manipulate the current object in global 3D space.

TRANSLATE is used to move the object in 3D space by keying in X, Y, and Z values in the active unit system. If the desired values are not known, a MEASURE option is presented within the TRANSLATE command. This allows the user to measure the distance between selected entities, and to call up the coordinates of selected points. Once the values are known the user can key them in and the solid will be move to the new location.

MOVE_TO is used to pick a point on the object to be moved, and a point in space to which the first point will be moved to. Once these points have been picked, the solid will be translated in space such that these point will coincide. Thus, it will "slide" to the new location, without rotation.

The MOVE_ALONG command performs a translation of the object like the MOVE_TO command, except the translation is based on a user defined vector in 3D space. This command presents several options for defining this vector.

The ROTATE command is used to pivot the current object about the global X, Y, and Z axes, pivoting about a selected pivot point. The default pivot point is the origin. Thus, the active object can be pivoted about the origin by typing:

/OR R <RETURN> <RETURN>,

followed by the desired rotations separated with commas, as:

X-ROTATION, Y-ROTATION, Z-ROTATION <RETURN>,

in degrees (e.g., 10,0,30). If another pivot point is desired, rather than entering a return when prompted for the selection of the pivot point, the user can use the activated screen cursor to pick a visible point. Or, a selection can be made from the menu presented for other options, including: selection of a point's LABEL; keying in a point location (KEY_IN_XYZ); selecting a TRANSLATED point; an option to INTERPOLATE between two points; selecting the object's CENTER_OF_GRAVITY; or picking a working set or profile point (POINT_FROM_2D).

The POINT_PIVOT command is used to rotate an object about a selected pivot point through the alignment of two implied lines. The command is based on the selection or specification of three points. The options presented for defining these points are the same as were discussed above for the ROTATION command. The first point that must be defined is the pivot point. Next the "point to rotate from" is required. These first and second points define an implied line which defines the object's current orientation in space. Finally a "point to rotate to" is required. This point and the pivot point define and implied line which defines the desired orientation for the object. Then the program rotates the object about the pivot point such that the second implied line aligns with the first implied line.

The VECTOR_PIVOT will rotate the current object about a selected pivot point such that two user defined vectors are parallel. The user is prompted to pick the pivot point. Then the program requests the definition of the first vector. Several selection methods are presented, with POINT_TO_POINT as the default. If it is used, the user picks the start point for the vector and then picks its end point. Final the user is asked for the definition of the second vector. Again the user can use the default POINT_TO_POINT vector definition method to pick the start and end point for the second vector. Once the inputs are complete, the program will rotate the object about the pivot point such that the first vector is parallel with the second vector and pointed in the same direction.

The ORIENT command ABOUT_VECTOR is useful for rotating the active solid about an axis other than those parallel to the global axes. The command leads the user through the definition of the vector used for the axis of rotation. First the user is prompted to select a vector definition method, and to define the vector. The vector need not be positioned on the axis of rotation. It only serves to define its orientation. After it is defined, the program will prompt the user to define a point on the axis of rotation which will serve to position the vector. Then once the vector's orientation and position are defined, the program asks the user to select between the ROTATION_ANGLE and POINT_TO_POINT methods for defining the amount of rotation desired. If the ROTATION_-ANGLE method is chosen the user will be asked for the angle in degrees. If the POINT_TO_POINT method is used, the program will ask for two points. Then it will define an imaginary line from the points perpendicular to the axis of rotation, and then it rotates the first line and the object until it is parallel to the second line.

The THREE_POINT_METHOD redefines the global axis system about the active object, to in turn redefine the orientation of the object with respect to the axes. The first point required is the new origin. The second point is any point which is to lie on the X-axis. And, the third point is any point which is to lie on the new XZ plane. Once these points are defined, the axes will be redrawn in the new orientation.

Each of the eight ORIENT commands will result in the object automatically being redisplayed in it new orientation, unless the user has chosen to turn this feature off. If that is desired, the user can input MODEL_FILE, USER_INTERFACE_SWITCHES, AUTO_DRAW, OFF.

INTERPOLATION

Many I-DEAS operations involving the specification of points in space allow the user the option to INTERPOLATE. This can be used quite creatively, and thus deserves some discussion here. When INTERPOLATE is selected as an option from within an I-DEAS command, the user is prompted to pick two points in space, and then to key in the percentage for interpolation between them. The percentage value is input as a whole number (i.e., 10.5 for 10.5%, and not 0.105). The default value is always 50.0 (i.e., 50%). Once the percentage is keyed in, the program calculates a point in 3D space which is located that percentage of the distance between the two end points, away from the first point selected. It then completes the command's operation in which INTERPOLATE was selected (e.g., creating a point at that location, or using that point as an end point for a vector being defined, etc.).

MEASUREMENT COMMANDS

Object Modeling provides the following measurement commands: POINT_TO_POINT, POINT_TO_LINE, POINT_TO_PLANE, THREE_POINT_ARC, ANGLE, and COORDINATES_OF_POINT. The POINT_TO commands are used to pick points and lines, or planes between which the program will calculate the distance. The THREE_POINT_ARC commands allows the user to measure the radii of arcs. The ANGLE_MEASURE command allows the user to measure the angle between two lines. It requests the selection of a point on the first line, a point at the vertex of the angle to be measures, and a third point that lies on the second line. The program will then calculate the angle between the two lines, in the plane of the three points selected. Finally, the COORDINATE_OF_POINT command does what its name implies. It lists the coordinates of an point specified.

MODIFYING OBJECTS

Object Modeling contains commands which can be used to alter solids in order to produce new objects or to refine objects through the movement of their facets, and by reflection of the existing facets. Some of these commands perform operations which can not be tracked in their objects histories. These end with an "@". Other commands in this menu, those ending with an "*", remove some of the objects' precise surfaces, and result in the deletion of their object histories. The remaining commands changes are recorded in the objects' histories.

Because these operations can drastically alter the objects, before they are executed the program will determine if the existing object has been stored. If it has not, and if the OLD_STACK_LENGTH has been set to a value greater than zero, the program will store the object as OLD1 before altering it.

Some caution must be taken when using these commands. Because these commands increase the number of facets contained in the objects model, some at a rate of two raised to the iteration selected plus one power (i.e., 2^(i+1)), the results can easily get unwieldy. This can even result in numbers exceeding the program's allowable limits. More facets require more: time to draw the fully faceted model on the display screen; disk space to store; and computation for further operations, etc. Thus, the lowest number of usable facets for the desired results (e.g., for understandable display or for further object creation, etc.) is desired.

The FACET_REFINE_@ command, which was mentioned along with the BOOLEAN commands, is used to increase the number of facets in the model by decreasing the maximum deviation between the facets' edges and the object's precise (rational B-spline) surfaces. This does not effect the precise model. But, it can be helpful, or even necessary, for the proper execution of some other commands which rely on the faceted models, such as the CUT and JOIN commands. This command first asks which of the current object's surfaces are to be refined. The surfaces can be selected by cursor picking of VISIBLE surfaces; by keying in their LABELS; or they can all be refined, by entering an "*". Once the surface(s) have been identified, the user is prompted for the method of subdivision that is desired; either ABSOLUTE_DEVIATION or PERCENT_OF_-EDGE_LENGTH. These are the same methods presented for the creation of objects, described earlier in this chapter along with the EXTRUDE command. If a smaller percentage or absolute deviation value is applied than was used to create the object, a finer faceted model will result. If the same deviation is applied, only areas which were created at intersections with CUT and JOIN operations will be refined. In either case, the faceted model yielded will more closely approximate the precise model. Once the deviation method and the maximum value have been input, the program will ask for maximum number of subdivision iterations that are to be performed on each facet associated with the surfaces being operated on. The default is 2, and the maximum is 5. This can be used to prevent long processing times. Finally, a point convergence tolerance is requested. This effects the distances between the new facets' corner points, and the model's precise surfaces, with smaller values resulting in higher precision.

The Surface Split command is used to subdivide the facets associated to selected surfaces on the object, yielding a finer faceted model. It also places all of the new facets' corner points within a specified distance from the objects precise surfaces. This can be useful for increasing the number of points for use with some of the other SHAPE operations, and for the TWEAK operations. The user is prompted to select the surfaces to be modified, with the same options for surface selection presented in the FACET_REFINE_@ command. Next the user must input the maximum number of subdivision iterations desired. This can be from 1 to 5, which will resulting each edge of the selected facets being subdivided into 2 raised to the iterations power, new edges. The final input required is the point convergence tolerance. This will determine the allowable distance between the new corner points and the object's precise surfaces.

The MOVE_TO_PRECISE command can be used to clean up objects that have been created through the use of the CUT and JOIN operations. These objects may have points that define the intersections that were used in these operations which do not lie on the object's precise surfaces; unlike objects developed with the CREATE commands, whose points always lie on the rational B-spline surfaces which define the solid's precise surfaces. The user is ask to select the points that are to be modified. Two options are provided for this selection: IMPRECISE_POINTS, and ALL_POINTS. I-DEAS carries internal markers on each point telling the program whether they lie within a known distance to the precise surfaces. Thus, it is possible for the program to modify only those points which are not within the convergence tolerance value, or all points can be reevaluated. Once this selection is made, the user must input the convergence tolerance that is to be applied to the points being moved.

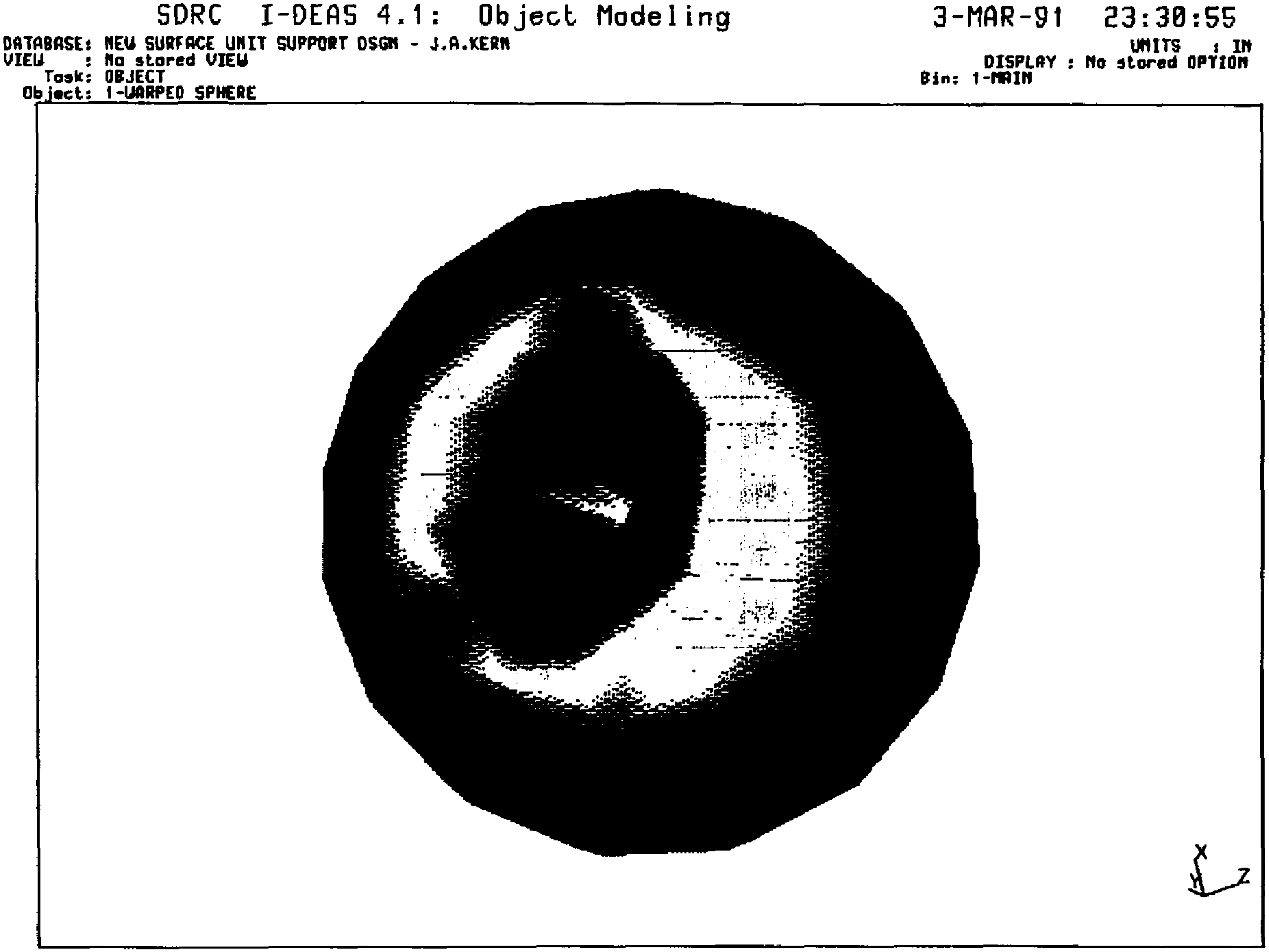
The REFLECT command is used to create new solids, or to complete the creation of whole solids by reflecting the existing object across a plane of symmetry. This allows the user to create symmetrical solids by creating half of the object, and then reflecting that half over the plane of symmetry to complete the part. This command can also be used to create a new object which is a mirror image of the existing object. Which allows the user to design a "left hand part" and then reflect it to create a matching "right hand part", or visa-versa. To provide these options, the commands asks for the desired output, which can be either BOTH_SIDES, or the REFLECTION_ONLY. It then asks for the plane of symmetry to be used. The options for plane specification methods are the same as were described for the CUT command.

The SCALE command can be used to scale the object up or down in each of the three global Cartesian directions. This allows the user to increase of decrease the object size. One example of the use for this is: if an object was created with the active unit system set to millimeters, when the user was entering lengths in inches, he could change the active units to inches, and then use the scale command to increase the objects size back up to the appropriate values. Because this command allow the user to input separate value for the X,Y, and Z scaling factors, it can be used to distort the object as well as simply changing its size. The first input require by the SCALE command is the specification of a point about which the object is to be scaled. The selection of this point can effect the resulting object's location is space. This is because the object's point locations will be multiplied by the scale factors entered, with respect to the point specified. But, this only means that the resulting object may be translated in space, which will require some use of the ORIENT, TRANSLATE command to reposition it as desired. The final inputs required by this command are the X,Y, and Z scaling factors desired.

The Surface WARP_@ command can be used to alter the contour of an objects' precise surfaces elastically. It allows the user to specify a point on the interior of one of the object's surfaces (i.e., it can not be a point on the edge of a surface) to move in space. Once a point is specified, the user is asked for the X, Y, and Z displacement desired. This is followed by a prompt for the radius of influence within which the modifications are to be made. This is the radius of an imaginary sphere centered at the point specified. Any of the points on the surface containing the point that was specified, which lie inside this sphere can be displaced by this operation. The default radius is the maximum value which will not include any of this surface's boundary points. The last options presented by this command are for the type of shape the modified surface is to have. These are either LINEAR, CONCAVE, or CONVEX. The program will displace each of the object's points that are contained in the imaginary sphere of influence by fractions of the displacement input for the specified point, with the fractions ranging from 1 to 0 depending on the points' locations relative to the point specified. Thus, if a LINEAR shape is chosen, the resulting surface will be conical. If CONVEX, or CONCAVE are selected, the user will be asked for a weighting factor, which will be used to control the transition of the fractions applied for each point's displacement.

The operation does not add any new points, or facets to the model. It can only displace those which already exist. Thus the Surface SPLIT or FACET_REFINE commands may need to be used to generate a finer faceted model for warping.

The author has not used this operation for this thesis, or any other projects. But, it was experimented with for this writing, and it appears to be very powerful for use in creating complex shapes and fine details. As an example, Figure 60. shows a sphere which was modified with this command to impose a concave "dent" in its surface.



```
            SDRC  I-DEAS 4.1:  Object Modeling               3-MAR-91  23:30:55
DATABASE: NEW SURFACE UNIT SUPPORT DSGN - J.A.KERN                   UNITS   : IN
VIEW    : No stored VIEW                                 DISPLAY : No stored OPTION
   Task: OBJECT                                      Bin: 1-MAIN
 Object: 1-WARPED SPHERE
```

FIGURE 60'. - WARP COMMAND EXPERIMENT

Object Modeling also provides several special object modification command which perform similar operations to those discussed above. But, each of these commands may result in a loss of association between the faceted models and the precise rational B-spline models surfaces. These special Surface and Facet modification commands are: BEND_*; BLEND_*; STRETCH_*; WARP_FACETS_*; THICKNESS_*; AND SURFACE_SCALE_*. The author has never found a need, or reason to want to modify an object in any way that would destroy its precise surface definition. Thus, without any experience in using these commands, their operations will not be discussed here.

One special modification command that was used for this thesis is the BODY DELETE command. It was used to delete unwanted bodies from objects which included several bodies that were created when objects were divided with the CUT command.

APPENDIX IX

I-DEAS ASSEMBLY MODELING OPERATIONS

INTRODUCTION

The Assembly Modeling provides commands for use in the construction assemblies of solid models, orienting the entities used in assemblies, adding auxiliary geometry and notes to the assemblies, and to display and interrogate system assemblies. An assemblies created in I-DEAS is constructed as a "HIERARCHY" of "INSTANCES" of COMPONENT, and sub-SYSTEMS.

Assembly Modeling is like Object Modeling in the way entities are treated as "active"; inactive, but visible (i.e., "unblanked"); stored; and saved. Also, like in Object Modeling, assembly entities are only held in the program's working memory, unless they are stored, with STORE commands, into the scratch data base files. They are not saved into the permanent data base files, unless a SAVE command is executed.

ASSEMBLY MODELS' STRUCTURE

The organizational structure of I-DEAS Assembly Models is based on a tiered hierarchy, built is a pyramid fashion. I-DEAS uses the term "system" for the combination of objects into groups that are associated and located with respect to each other. At GEA these same concepts are usually termed "subassemblies" and "assemblies". But, the concepts are the same. The user wants to combine sets of objects as they will be assembled in the actual mechanical system's construction or manufacture. These sets of objects must be located in space such that they will fit together as they are designed to be assembled. Also, it is desirable to be able to construct pieces of the overall system in subsystems or subassemblies one time; and then to be able to use these subsystems several time throughout the complete system assembly requiring only the positioning or orientation of these subsystems with respect to the complete system. The Assembly Modeling commands provide these capabilities.

Subassemblies are constructed as "systems", and then they are stored, and larger "systems" are constructed using stored "systems" as subassemblies. Thus, I-DEAS always refers to "systems", even though they may be used as sub-systems in other systems. Solid models that are used in the Assembly Modeling are call "components". When components or sub-systems are used in other systems, they are call "instances" of the stored component or subsystem.

CURRENT SYSTEM AND STORAGE

Assembly Modeling works like the other I-DEAS Modules, all of the work is done on a temporary "current" system assembly. Any changes made to the current system do not effect any of the entities that have been stored in the working model files or those that have been saved in the permanent model files. Once a system is assembled to the user's satisfaction, it should be STOred. This will store it in a system assembly storage bin of the user's choosing, within the working or "scratch" model files. Then when the user uses the SAVE command, the changes made in the scratch files will be saved by writing the scratch files over the permanent model files.

Similarly, Assembly Modeling also provides a MANAGE menu which have the same type of commands as Object Modeling's MANAGE menus. With these commands the user can manage the storage bins, components, and systems. They also provide commands used for interfacing with external programs, Object Modeling, and several of the other I-DEAS Families of programs.

CREATING COMPONENTS AND INSTANCES

When a solid model is to be used in system assembly, a "component" must be created, and stored in the current System Assembly system bin with the CREATE command. Each component acts as a locator in the Assembly Modeling for the geometry that is stored as an OBJECT in an Object Modeling object bin. Once a component is created, I-DEAS will not allow the user to delete the solid model stored as an OBJECT in Object Modeling. This prevents the user from inadvertently destroying a system assembly.

In addition to storing components, the CREATE commands also adds an "instance" of that component to the current system. I-DEAS refers to an instance as a "place holder in a system". It tells the system assembly which component or subsystem the instance represents. An instance also contains the component or subsystem's orientation in a system. This allows a several systems to contain many "instances" of a single "component". But the component only needs to be stored once in the system storage bin. With this method of system assembly, changes to the orientation of an entity in a system assembly, only change an instance's orientation information. The orientation of the component remains constant.

The CREATE commands allow the user to specify which Object Modeling object bin contains the object(s) desired, and then to select the specific object(s). At the object bin and object selection prompts, a "?" can be entered to view a directory of the available entities.

After the bin and an object have been selected, the user is asked to name the new component, or to enter its number for storage in the system bin. The defaults are the solid model's object name, and the next available number.

After the component is identified, the program asks for a name for the instance that is being added to the current system. The default instance name is "IN1". If this component is "instanced" a second time in the same system, the instance's default name will be "IN2", and a third will be "IN3", etc. But, the user can choose to name the instances in a more meaningful way. For example, if an electric range cooktop system assembly was being constructed, the user might choose to name the four surface unit assemblies: LF SU; LR SU; RF SU; and RR SU, to indicate which instance is the left front, the left rear, the right front, and the right rear surface unit.

In addition to naming each instance, the program automatically assigns a sequential "hierarchy number". The system is implicitly given the hierarchy number 1. If a system is instanced into another system as a subsystem, that instance will be assigned the next sequential hierarchy number, and the other instances in the subsystem will assigned sequential numbers following the subsystem's number.

The ALL_OBJECTS_FROM_BIN command will bring all of the solid models stored in one Object Modeling object bin into the Assembly Modeling. Each object will be stored as a component in the current system bin within Assembly Modeling. Their component names will be made the same as their object names. If an object name already exists in the system bin, the program will prompt the user for a new name or number to use in creating the component. If this occurs, the default new number will be the next available instance number, and the default name will be COMPONENT#, where # will be the default number (i.e., COMPONENT15). Each new component will also be instanced once in the current system. The instance names will all be made IN1. Once the components and instances have been created, the user can change their names through the use of the MANAGE, RENAME and INSTANCE, RENAME_INSTANCE commands respectively.

INSTANCES IN THE CURRENT ASSEMBLY MODEL

Once components have been created and stored in the current system bin, they can be added to the current system assembly, without the need to create new components. The instances in the current system can also be duplicated, replaced, deleted, and renamed. These operations are performed with the commands presented in the INSTANCE menu, which are logically named: ADD_FROM_DIRECTORY, DUPLICATE_INSTANCE, REPLACE_-INSTANCE, DELETE_INSTANCE, and RENAME_INSTANCE respectively.

ORIENTATION OF INSTANCES

The instances of components and subsystems in the current system can be oriented with the commands in the ORIENT menu. These commands function much like those discussed in Chapter V, for the orientation of solid models, and therefore do not need to be discussed here. But, there are differences which do warrant some discussion.

The graphical representation of the effects of each of the ORIENT commands will be displayed on the terminals screen in one of two ways. The choice between the two is made by toggling the MODEL_FILE, USER_INTERFACE_SWS, AUTO_DRAW_SWitch ON and OFf. If the AUTO_DRAW_SW is ON, when an orient command is executed, the effected instance(s) will be erased, and then redrawn in the new orientation. If it is OFF, they will not be redisplayed. An arrow will be drawn on the screen which points from the instance's original position to their new position. The next DRaw or REDIsplay command that issued will cause the instance(s) to be displayed in their new orientation(s).

When instances' orientations are modified with these commands, they can be moved relative to one of several coordinate systems. Either the current system's, the entity's, or another entity's. The user is prompted for the selection of the desired coordinate system during the execution of the ORIENT commands.

The user can store multiple configurations of the orientation of the current system's instances with the Hierarchy, ORIENT, STORE command. These configurations also store the blanking data for the system's configuration that is stored. This feature allows the user to store several configurations of a system, without the need to store them as separate systems. Once a configuration is saved, it can be reestablished with the Hierarchy, ORIENT, GET_CONFIGURATION command. This operation will allow the user to store the configuration of the entire current assembly model, or of a selected subsystem. If a subsystem's configuration is store, it is associated with that subsystem. If that subsystem is made the current system, the configurations stored will be available. The current system's configuration is automatically stored as 1-AUTOMATIC_STORE/GET whenever the system assembly is stored. This STORE command is like all other I-DEAS store commands, it only stores the configurations in the scratch model files, and they will only saved in the permanent model files when the user issues a SAVE.

Each of the ORIENT commands require the user to pick the instance that is to be re-oriented. Both component and subsystem instances can be oriented with these commands. Each instance in the current system is identified by a component or system name and number, system bin number, hierarchy number, and instance name. Because of this, several methods are presented for the selection of the instance.

VISIBLE_INSTANCE is used to picked with the entity with the screen cursor. Instances can be picked at the corner of a facet or the end of a line. If the user has difficulty getting a clear shot at picking an entity, the ORIENT commands provide DRAW and CHANGE_VIEW options to improve the visibility of the entities desired. Once an entity is picked, the program will present a list of its instances in the current system. Thus, the component instance, and any subsystem(s) in which it is instanced will be listed. The listing will present the full description of each (i.e., their component or system name and number, system bin number, hierarchy number, and instance name). This allows the user to choice whether the single component instance's or a complete subsystem's orientation will be changed.

In addition to the ability to pick an entity for orientation from the display screen, the ORIENT commands also allow the user to type in specifications for the instance desired. The LABEL_FROM_CURRENT option allows the user to input an entity's name or number. The HIERARCHY_NUMBER option allows the entity to be specified by its hierarchy number. If either of these options is used, and the user doesn't know the name or number of the desire entity, DIRECTORY options are available within these commands. By typing a "?", a listing of the entities or the hierarchy of the current system will be displayed.

The ORIENT commands provides the capability to move more than one instance at once, and they do not have to be in the same subsystem. When the user first enters an orient command, before an instance is selected, the user can select INCLUDED_INSTANCES from the menu presented. This will cause the program to continually prompt the user to select entities, until the user issues a quit with the default <RETURN>.

The Hierarchy Task's ORIENT commands also allow the user to make additional changes to an instance's orientation, without the need to re-select the entity. When the program prompts the user for the selection of the instance to be oriented, if user selects PREVIOUS_INSTANCE from the command's menu, the last instance(s) will be selected.

When the user first enters an orient command, before an instance is selected, the user can select REPEAT. This allows the user to move entities in steps, and to store each step as a system configuration if desired. This could be used to create animation sequences, or to provide several steps in a mechanism's motion for investigation (e.g., interference checks). REPEAT will cause the program to ask the user for the number of times to repeat, and if the configurations should be stored. If the user answers that he does want to store the configurations, the program will prompt the user for a configuration name or number for the storage of each step. If the user would like to use the default names for all of the step, he can enter a single ":". The total motion of the instance being oriented will be divided by the number of steps specified into equal sized steps.

The user can choose to create duplicates of an instanced component or subsystem as it is oriented, rather than moving the present instance. This is done by selecting DUPLICATE_ORIGINAL before selecting an instance in one of the ORIENT operations.

The INCLUDED_INSTANCES, REPEAT, and DUPLICATE_ORIGINAL options can be used in combination during a single orient operation. To help the user keep track of which options have been selected, the Hierarchy's ORIENT commands contain a STATUS option, which will display a listing of these options' status.

As an alternate to using the TRANSLATE and/or ROTATE commands, along with INCLUDE_INSTANCES, and REPEAT options, the Hierarchy Task's ORIENT, MULTI_INSTANCE_ORIENT command (/OR MIO <RETURN>) can be utilized. This commands prompts the user for the translation and/or rotation of multiple instances. It allows the change in orientation to be different for each instance selected. It also prompts the user for the number of times the instances are to be REPEATed through the orientation, and it asks if the configurations are to be stored.

The ORIENT menu provides an ALIGN command. It can be used to set the translation and/or rotation values of one instance equal to those of another instance's. If the translation values are matched, the moved instance's origin will coincide with that of the other instance selected. If the rotational values are matched alone, the instances' coordinate systems will be parallel to each other.

The most complex method presented for the orientation of an instance is the MATRIX_METHOD. It allows the user to orient an instance with respect to either the current system or to the instances parent subsystem. This is accomplished by entering the twelve value for the program's internal transformation matrix, whose mathematical format is:

$$[\ X,\ Y,\ Z,\ 1\ ]_{\substack{\text{LOCAL}\\ \text{INSTANCE}\\ \text{SPACE}}}\ \text{X}\ \begin{vmatrix} A & B & C & 0\\ D & E & F & 0\\ G & H & I & 0\\ T_X & T_Y & T_Z & 1 \end{vmatrix} = [\ X,\ Y,\ Z,\ 1\ ]_{\substack{\text{CURRENT}\\ \text{SYSTEM}\\ \text{SPACE}}\ \text{or}\ \substack{\text{IMMEDIATE}\\ \text{PARENT}\\ \text{SPACE}}}$$

The user is prompted to enter the nine values in sets of three, separated with comas, by row (i.e., row 1 = A , B, C, etc.) in the matrix.

Assembly Modeling provides two ORIENT commands which can be used to return the systems orientation to a previous or known state. This can be very useful if the results of an executed orientation operation yield undesirable results. The UNDO_LAST_ORIENT command does what its name implies. It return the system's orientation to the same state as it was when it was last stored, or gotten from storage. If the user STOres the system, the previous orientation is overwritten, and the UNDO command is reset to the new configuration. The RESET command returns all instances to their initial positions, as they were brought into the system. This is done by deleting any orientation values from their instance definitions. RESET presents a menu from which the user can choose to reset either the translation or the rotation only, or both.

Assembly Modeling's ORIENT and LIST menus contain the same MEASURE capabilities found in the Object Modeling. This allows the user to measure the distances between points (i.e., ends of line and facet corners) and points, lines, and planes; to measure arcs; and to measure angles. It also allows the user to list the coordinates of points. This can be helpful when determining how far an instance needs to be translated, or to investigate the systems layout. The TRANSLATE command also offers MEASURE as an option within the execution of its operation.

Assembly Modeling's ORIENT menu offers a sub-menu which is not available in other I-DEAS modules. Its commands are used to view system assemblies with their instances spread apart. This can be useful for getting clear shots at items when making screen cursor picks. It can also be used in the creation of exploded view drawings. Two commands are presented when the EXPLODE sub-menu is requested: EXPLODE_ALONG_AXIS and INITIAL_SPREAD. Systems can be exploded several times if desired, and the two EXPLODE commands can be used in together.

These commands alter the current system's orientation configuration. Therefore, if the user wants to retain the current configuration, the ORIENT, STORE command should be used. This will allow the current system configuration to be retrieved later, if needed.

The EXPLODE_ALONG_AXIS command will translate all of the entities in the parent system selected, along a single Cartesian axis. The entities are separated along a specified axis, with a user specified minimum spacing between each entity. This can be helpful in the development of exploded view drawings. For example, at GEA, this could be used to develop the product service parts catalog illustrations. This command only explodes the instances in the system selected. The default system is the entire current system. But, by selecting another VISIBLE_SYSTEM, or another system's label or number, subsystems can be exploded first. Once the desired degree of explosion of a system assembly is achieved, it can be stored as a configuration for future recall.

The INITIAL_SPREAD command explodes all entities in a selected system (the current system is the default) about the system's Z-axis. They are spread out radially, and are separated by dividing the 360° spread evenly by the total number of entities in the system selected. The user is prompted to input a "spread factor" (default is 1.0). This factor is multiplied times a calculation the minimum/maximum ratio of boxes that enclose each entity and the entire system, times the vector direction the entity that is being translated, when moving each entity.

ASSEMBLY MODELING'S CONSTRUCTION COMMANDS

Assembly Modeling contains several construction commands which can be used to cut and intersect objects of component instances in the current system assembly, and to check for interferences between its component instances.

The CUT, INTERSECT, and PLANE_CUT commands function in much the same way that they do in the Object Modeling Module. The differences between the operation of these commands as presented in Assembly Modeling are because of the differences in working on systems of component instances versus working on single objects. But, the same limitations and tolerancing issues exist in performing these operations on assembly models as they do on objects. Therefore, the reader may want to review the discussion on these commands and their limitations presented in Chapter V.

The construction operations which yield new objects in the Object Modeling Module, will yield new Object, Components, Instances, and Systems in the System Assembly Module. Therefore, when these commands are executed on the current system assembly, the user will be prompted to specify the Object bin within Object Modeling into which new objects are to be stored. They will also ask for the entity bin within Assembly Modeling into which new components and systems are to be stored. Then the software will automatically store any new objects, components, and sub-systems.

If any configurations had been stored for the current system prior to the execution of a construction operation, those configurations will automatically be updated and retained in the new current system which is created by these commands. But, the new system assembly will not automatically be stored. It will only exist as the working current system. If the user wishes to retains the results obtained with a boolean operation, he must execute the STORE operation. Likewise, the stored results will not be retained in the permanent model files unless the SAVE command is used.

Assembly Modeling also provides a CROSS_SECTIONS command that contains the ability to generate cross sections of the current system. These cross section will be made up of one or more 2D profiles constructed from the shapes generated. These profiles will automatically be named and stored in the Object Modeling profile directory. A collection of all of the profiles produced will also be saved in the System Assembly Module as a cross section. Cross sections are stored in a directory which can be accessed and managed.

AUXILIARY GEOMETRY MENU

The AUXILIARY_GEOMETRY menu present POINTS, LINES, and WIREFRAME menus. The commands in these menus can be used to add, orient, modify, delete, and list geometry, that can be created in the current assembly model.

The AUXILIARY_GEOMETRY menu also contains access to a NOTES menu. The NOTES commands allow the user to create text notes, and to position them for display in the current system's 3D space along with its geometry. These can be used to annotate entities that appear in the 3D displays. The NOTES commands allow the user to define, orient, modify, blank and unblank, delete, and setup the attributes for these notes.

LISTING ASSEMBLY MODELS' PHYSICAL PROPERTIES

Assembly Modeling provides commands in the LIST menu which can be used to compute and list the physical properties of the entire current system assembly. These properties are the combined inertial properties of all of the components, each with its own density, that exist in the current system.

The components' physical properties must exist before the System Task can calculate the current system's properties. The components' properties can be calculated in Object Modeling when the objects associated with the components are created. If properties are created in Object Modeling, then when a system assembly component is created, the associated object's properties are assigned to the component. If physical properties are not created in Object Modeling, calculations are made, based on the faceted model, when a component is created, and the results are associated with the component. Finally, the user can use the COMPUTE_PROPERTIES to recompute, or enter a complete set of properties explicitly, for each component.

Once a system's properties have been calculated, they are not automatically recalculated when changes are made to the system (e.g., adding or deleting components, or changing their orientation). Therefore, they must be recalculated. Once a system's properties are calculated, the system must be STOred to retain the data. When this is done, and new properties of the system will be applied to all other systems which use it as a sub-system.

APPENDIX X

I-DEAS DISPLAY OPTIONS OPERATIONS

DISPLAY_OPTIONS

The DISPLAY_OPTIONS menu contains commands which can be used to create line, hidden line, shaded, and surface boundary displays. It provides command menus used to control the display of entities, labels, and tasks. Options for the control of the viewport, and display colors are located in this menu. Finally, customized display options can be stored, activated, and managed from within this menu.

LINE DISPLAYS

LINE displays are very similar to those produced with the global DRAW command. Except, LINE displays allow the user to control the display of feature lines. These feature lines are effected by the parameters that are established with the DISPLAY_ATTRIBUTES, FEATURE_-ANGLE command. The Line display menu allows the user to turn the feature lines ON and OFF, with the FEATURE_LINE_SW. It allows LINE displays to be SET_AS_GLOBAL, so that any subsequent GLOBAL_DISPLAY (GD) executed with produce a LINE display. A STATUS command is presented which can be used to display the status of the parameters that as set for this display option. Finally, this menu provides an EXECUTE command that causes the display screen to be erased and the display to be redrawn using the LINE display parameters.

HIDDEN LINE DISPLAYS

Hidden line displays can be setup and produced with the commands located in the DISPLAY_OPTIONS' HIDDEN_LINE sub_menu. Like the LINE display menu, HIDDEN_LINE menu contains: a FEATURE_LINE_SW; a SET_AS_-GLOBAL command; and a STATUS command. This menu also contains a RESOLUTION command. This is used to control the sharpness of the display that is produced. It effects the number of raster display lines the program calculates. This is based on raising 2 to the power of the RESOLUTION value set (between 1 and 9), and multiplying that times 1024. The default RESOLUTION value is 6. The lower the value, the faster the display will be produced. But, the higher the value, the more accurate the display will be. Finally, as with all of the DISPLAY_OPTIONS, an EXECUTE command is provided. This HIDDEN_LINE command allows displays to be created with all lines that are hidden from view to be removed from the display. This allows models to appear to be solid objects.

SHADED IMAGES

SHADED_IMAGES are the most complex of the I-DEAS display options. This display option allows the solid models to have their surfaces shaded, such that they appear to be solid objects on the graphics screen. The SHADED_IMAGE menu provides several options for use in customizing shaded image displays, in addition to the same EXECUTE, SET_AS_GLOBAL, and STATUS commands that the other display options types provide.

The OVERLAY_HIDDEN_SW is used to toggle ON and OFF the display of hidden line images of the solid models over their shaded images. When turned ON, this can aid in distinguishing between the objects various surfaces. The COLOR_OF_OVERLAY command is a companion to the overlay's ON/OFF switch. It is used to specify the color of the hidden line overlay.

PALETTE calls up a sub-menu that can be used to display the preset colors that can be specified for use with any of the I-DEAS entities. It also contains a COLOR_CUBE command which yields a display of six side of a cube which is shaded with continuously varying colors that are labeled at its corners with the primary color combinations that it takes to create the colors. This can be used along with the ENTER and MODIFY user created colors, by mixing percentages on the primary colors (red, green, and blue). The PALETTE menu also presents DELETE, LIST, RENAME, and STATUS command which can be used to maintain user created colors.

The SHADED_IMAGE menu also provides a LIGHTING menu. This presents commands that can be used to CREATE, MODIFY, SKETCH and manage light sources. These light sources can then be made CURRENT, and the next time a SHADED_IMAGE display is EXECUTED, the shading will take the lighting into account.

The most complex part of creating shaded images that appear to be pictures of actual objects is the design and setup of the lighting used to illuminate the images. This can require many iterations of experimenting with various settings before the desired results are achieved. This is not difficult, and the default setting can give reasonable good shaded images.

When light sources are created, the user can use a light at an "INFINITE_DISTANCE", or BARE_BULB, SPOT_LIGHT, or FLOOD_LIGHT types of light sources. The user can store light sources by name and/or number. Lights at an infinite distance must be defined by a vector method, defining the direction in which the light will shine. Bare bulb, spot, and flood lights sources can be located in space through the selection of a VISIBLE_POINT, a TRANSLATED_POINT, or by keying in a KEY_IN_SCREEN location, or KEY_IN_MODEL location. If a light is created with its location referenced to the screen rather than to the models space, it will remain stationary when the object(s) are moved in the viewport. But, if they are tied to the model's reference frame, the will move with the objects. A bare bulb requires the user to "enter [the] distance from [the] source at which [the] intensity applies. The intensity is 100% by default. To change this, the user must MODIFY the light source. This is true for all light sources. If a spot or flood light is created or modified, in addition to the location of the light, a location is required for the TARGET of the light's rays, and a CONE ANGLE is required to limit the volume on space that the light will illuminate.

When a light source is been created it automatically becomes active in the current working viewport. Several light sources can be used in the same viewport. This is accomplished by executing the CURRENT command from the SHADE_IMAGE, LIGHTING menu. This command prompts the user for the name or range of numbers of the light sources that are to be used. This replaces the previous list of light sources, and the command will continue to prompt the user for additional lights until the user enters a <RETURN> without a name or number. The STAtus command can be used to list the current light source(s).

The LIST command can be used to list the settings of any stored light source. This includes its definition method (type of light), reference frame (screen or model), the position of the light source and its focus point is applicable, the light's color, intensity, at distance, and its angle of lighting is applicable. It will also list the viewports in which it is currently enabled.

Light sources' settings can be changed with the MODIFY command. This command will prompt the user for the modification of all applicable parameters for the light source that has been chosen for modification. These parameters include: COLOR, INTENSITY %, INTENSITY_AT_DISTANCE, FOCUS_POINT_POSITION, LIGHT_SOURCE_POSITION, LIGHTING_CONE_ANGLE, and REFERENCE_FRAME. Once the user has made the desired modifications, the default (DONE) can be entered with a D or simple by entering <RETURN> without any input. The command will then process the changes, store the modified light source, and then terminate the MODIFY command.

The SKETCH command will cause I-DEAS to display an "*" at the location and focus point of each of the currently active light sources, and a line between each light's *'s. This can aid the user in positioning light sources to accomplish the desired lighting effects.

In addition to using direct lighting techniques with specified light sources, the user can apply ambient light. This is specified by entering an AMBIENT_VALUE. This value is the percentage of the maximum ambient light that could be applied. This is entered as a whole number (i.e., 10.0 is entered for 10%) between 0 and 100.

Finally, once all of the lighting has been set up, the user can chose to filter of amplify the current lighting's effects. This is done with the EXPOSURE command, which prompts the user to enter a "fraction".

This value can be number between 0 and 100. The initial default value is 1. Any value less that 1 (i.e., a decimal fraction) will act as a filter, and any value greater than 1 will act as an amplifier.

The effects of the various lighting techniques are also effected by the DISPLAY_ATTRIBUTES that are specified for each stored object. These attributes are discussed in a following section in this chapter.

Other shaded image options presented in the SHADED_IMAGE menu. Several toggle switches are located in this menu. The TRANSLUCENCY_SW is used to turn ON and OFF the processing of translucency of objects whose TRANSLUCENCY DISPLAY_ATTRIBUTE has been specified as a non-zero value. The PRECEDENCE_SW is used to turn ON and OFF the processing of precedence of objects whose PRECEDENCE DISPLAY_ATTRIBUTE has been specified. Objects with higher precedence values will be displayed more visibly that those with lower values. The GRAYSCALE_SW is used to turn ON and OFF grayscale processing on color display terminals. finally, the REVERSE_HARDCOPY_SW is used to turn ON and OFF reverse grayscaling on non-color terminals.

The SHADED_IMAGE menu allows the user to specify the level of DITHERING that will be used when the program simulates a range of colors or grayscales on the display screen through the illumination of the display screen's pixels. For black and white displays this can be fairly easily understood in terms of the numbers of pixels that are processed in a group to create various levels of gray. But, it is more difficult to understand when dealing with color displays. In either case, the user can set this to values of: 0, to use only the device's available colors (i.e., no mixing); 1, for one pixel per group; 2, for 4 pixels grouped 2 by 2; 3, for 16 pixels grouped 4 by 4; or 4, for 64 pixels grouped 8 by 8. The user may need to experiment with the effect of various levels of dithering to yield the desired display results.

DEPTH_CUEING is an option in the SHADED_IMAGE menu that can be turned ON and OFF, and set for the intensity decay of depth cueing. This can be used to adjust the contrast in color and intensity of the background relative to the model to enhance the appearance of three dimensions. This requires the specification of a background color other than black. This is done with the COLOR_OR_BACKGROUND command presented in the main DISPLAY_OPTIONS menu, which is discussed later in this chapter.

I-DEAS offers several options in shading. This are specified by using the BASIC_SHADING, TERMINAL_SHADING, and ADVANCED_SHADING commands located in the SHADED_IMAGE menu. Basic shading is done by the host computer on which I-DEAS is being run. This will produce high resolution shading at a relatively slow speed. Terminal shading is available only on some terminals (e.g., Tektronix 4129). When it is available, it is faster, relative to basic shading, by may be at lower resolution. Advanced shading techniques include: SHADOWS, REFLECTIONS, ANTI_ALIASING, and PIXEL_FILTERING. The discussion of these techniques is outside the scope of this work. But, suffice it to say that I-DEAS is capable of producing computer generated images that can approach appearing to be photographs of actual objects. These techniques take a great amount of time to set up when attempting to produce desired effects, and can consume large numbers of CPU hours of computer time. But, because these advance shading techniques use ray tracing calculations, their use should be investigated by anyone that requires "industrial design" type displays. Also, some advanced engineering analyses involving radiation transfer (e.g., optics and heat transfer) have been able to take advantage of I-DEAS advanced shading techniques.

SURFACE BOUNDARY DISPLAYS

The final display type that is available in the DISPLAY_OPTIONS menu creates SURFACE_BOUNDARY displays. These displays are similar to hidden line displays. Except, SURFACE_BOUNDARY displays allow the lines that are hidden for view by the front faces of objects to be shown as dashed lines, or completely removed. In addition to the same EXECUTE, SET_AS_GLOBAL, and STATUS commands the other display options types provide, SURFACE_BOUNDARY displays also provide for: HIDDEN_LINE_STYLE; PRECISION; ACCURACY_OF_CLIPPING; HIDDEN_CHECK_INTERVAL; and COLOR_-OPTIONS.

HIDDEN_LINE_STYLE provides a sub-menu which allow the user to specify if the hidden lines are to be REMOVEd, DASHED, or SOLID. If DASHED is desired, this menu also provides a LENGTH_DASHED command for use in specifying the length of the dashes. This is based on the dashes size in relation to the display screen. The screen's width is considered to be 10 units wide. The LENGTH_DASHED command asks the user to specify the size of the dashes and the space between dashes, with an initial default value of 0.1 set for each.

The PRECISION command in the SURFACE_BOUNDARY display menu provides the user with the options to turn "precision" and SILHOUETTE_ADJUST_SW, ON and OFF. When precision is ON, surface boundary displays will be based on the precise (rational B-spline) surfaces intersections. When it is OFF, only the facet edges are displayed. Silhouette lines are view dependent lines. They are the lines that are displayed at the edge of curved surfaces when only part of the surface is visible from the current view point (e.g., the lines shown at the sides of a cylinder). This is set to ON by default, and is necessary for accurate data transfers of view-dependent displays made to I-DEAS DRAFTING and to IGES formatted output files. But, this requires additional computation time, and if this level of accuracy is not necessary, turning the SILHOUETTE_-ADJUST_SW OFF will speed up computer response time. The PRECISION menu also provides an option for specifying the CURVE_SMOOTHNESS, when precision is turned ON. This controls the number of line segments that are used to display each precise (non-uniform rational B-spline) surface. The default is for each facet of the model to be replaced with four (4) line segments. With this command, the user may specify any number between 0 and 100.

The SURFACE_BOUNDARY DISPLAY_OPTION's ACCURACY_OF_CLIPPING command gives the user to choice between SCREEN_DISPLAY, PICTURE_FILE, and IGES_OUTPUT as the basic levels for accuracy in computation of the visibility of points. SCREEN_DISPLAY will cause subsequent surface boundary displays to be clipped to within one (1) pixel. If PICTURE_-FILE is specified, the user will be asked how may times the image will be enlarged (e.g., for plotting). I-DEAS will then divide 1 by this number to determine the tolerance for clipping. Finally, if the user chooses IGES_OUTPUT, for writing view-dependent displays out in IGES format, I-DEAS will ask the user for the clipping tolerance that is to be used. This should be as small practical, and is input in the active model units (i.e., if the active units are "IN", then 0.001 is a 0.001" tolerance).

The HIDDEN_CHECK_INTERVAL can be used to subdivide the object's edges in order to detect small edges that should be hidden from view. The need for this options arises in cases where large simple surface lie in front of small details that should be hidden from view. Because of the was I-DEAS process hidden line displays, it may be possible that when the large surfaces vertices, and equally spaced points in between are established for deleting hidden lines, all of these points may not detect the small details. The check interval is entered in current model units. This allows the user to specify its value such that small details of known size can be used as the basis for setting a small enough check interval to be sure they will not be missed.

The last SURFACE_BOUNDARY display option is the COLOR_OPTIONS. This allows the display to be produced using either the OBJECT_COLOR or the SHADED_OVERLAY_COLOR.

GENERAL DISPLAY OPTIONS

The DISPLAY_OPTIONS menu provides several other options and sub-menus in addition to the four major display types.

Several of these options involve toggle switches. The ENTITIES_SW sub-menu provides the user to with toggle switches used to turn the display of working sets, profiles, skin groups, and systems ON and OFF for simultaneous display with solid models. The TASK_DISPLAYS display options sub-menu is very similar. It provides toggle switches for the display of objects, working sets, profiles, skin groups, systems, and components tasks on the same display screen. It also has a STAtus command which lists the ON/OFF status of each of these switches. The DISPLAY_OPTIONS LABEL_SWS sub-menu provides ON/OFF switches for the display of labels placed on points, facets, surfaces, and bodies. It also provides a SIZE command used to specify the size of these labels, and a STAtus command for listing the label switches status. Toggle switches are also provided in the DISPLAY_OPTIONS VIEWPORT sub-menu. These switches allow the user to turn ON and OFF the display of: the X, Y, Z, triad; the X at the origin of the model space; the viewport's outline; the viewport's label (number). This menu also provides a LOCK_SW with can be used to "lock" a viewport. This prevents the contents of the viewport from being modified when it is locked. A similar command is provided, call PICKABILITY. When this switch is OFF, the program will not offer the ability to pick entities with the screen cursor in subsequent command operations.

The DISPLAY_OPTIONS menu provides a RESET command which can be used to reset all of the all of the display option parameters to the values set when a new model file is created.

The DISPLAY_OPTIONS menu provides A COLOR_OF_BACKGROUND command, which can be used to change the default background color of black to any other standard or user created color. A black background often yields the highest contrast between entities being displayed on the graphics screen. Also, a black background will allow plots to be produced using no ink for the background, because white will be substituted for black, requiring no color on white paper. But, if the goal is to produce graphics displays for viewing on the display screen alone, or for photographs of the screen, other background colors may be desirable. Some care must be taken in choosing the background color. Because, the colors of the menus will not be changed, and they can become impossible to see on some backgrounds.

The DISPLAY_OPTIONS menu also provides the same PALETTE color options sub-menu that is provided in the SHADED_IMAGE menu. These commands are identical to those that were discussed along with the shaded image commands.

The AFFECTED_VIEWPORTS command in the DISPLAY_OPTIONS menu is used when multiple viewports are active on the display screen. It is used to specify which viewport will be affected by changes in the display options. When multiple viewports exist, this can be used to specify either WORKING_VIEWPORT_ONLY or PICK_VIEWPORT. IF PICK_VIEWPORT is specified, each time a display option is executed, the program will prompt the used to pick the viewport to be affected. If only one viewport exists, of if WORKING_VIEWPORT_ONLY is specified, only one viewport will be affected by subsequent display option changes.

The DISPLAY_OPTIONS menu presents four display option management commands. CURRENT_DO is used, like the GET command for entities, to specify which of the stored display options is to be made current. If multiple viewports are active, the user will be prompted to pick the viewport(s) in for which this display option is to be made active. This allows a different display option to be applied in each viewport. The STORE command is used to store the current display options as a store configuration for latter recall and use. The MANAGE command calls up the standard I-DEAS manage menu, which allows the user to DELETE, LIST, RENAME, COPY, and list the DIRECTORY of all stored display options. It also allows the user to change the OLD_STACK_LENGTH. This will automatically store display options as OLD1, OLD2, etc., if set to a value greater than zero, in the event that they have not been stored, and are about to be destroyed by a command (e.g., CURRENT_DO).

SPECIAL DISPLAY_OPTIONS OPERATIONS

The DISPLAY_OPTIONS menu contains a sub-menu called SPECIAL_OPERATIONS. This menu contains presents that allow the user to specify several operations that are available on selected terminals only.

The BLANKING_SW is a switch that can be used on all terminals. It allows the user to turn blanking OFF with a single command. This will cause all blanked entities in the current storage bin to be make visible in subsequent displays. When it is turned ON, normal blanking will be followed (i.e., blanked entities will be invisible).

If the terminal being used offers the ability to manipulate the view locally, the DYNAMIC_VIEW_SW can be turned on to enable this capability. If the terminal being used does not support dynamic viewing, and this command is used, the program will notify the user that this not possible. But, when this is available (e.g., on Tektronix 4129 terminals) it can be used to manipulate the views much more rapidly than when done by the main CPU (although at some what lower resolution). The DYNAMIC_ORIENTATION_SW is a similar ON/OFF toggle switch for dynamic orientation.

The LOCAL_SHADING_SW can be described in much the same way as the DYNAMIC_VIEW_SW. On some terminals, like a Tektronix 4129, the terminal is capable of performing the shaded image calculations. This, again at slightly lower resolution, can produce shaded images more rapidly that the main CPU. But, the real power of a terminal with local graphics display and shading capabilities is when these two special operations are used together. This will allow objects to be shaded in one orientation, and then viewed from another direction, and the object will automatically be displayed shaded in the new orientation.

The SPECIAL_OPERATIONS command named TERMINAL_DISPLAY_LIST is used to turn ON and OFF the storage of the display list in intelligent graphics terminals' memory. This option can be turned OFF if the terminal's memory capacity is exceeded.

The DISPLAY_OPTIONS, SPECIAL_OPERATIONS menu provides a FACET_-SHRINK_SW can be used to shrink all of an object's facets. This is like the element shrink operation in Engineering Analysis in which the elements can be shrunk to examine an FEA model of errors in element construction. In solid modeling, by shrinking an object's facets, the user can examine objects faces to the sure that no facers overlap. The facets are shrunk proportionally by the percentage set with the FACE_-SHRINK_% command in the DISPLAY_ATTRIBUTES menu.

The TRACE_LINE_STYLE command is used to specify either SOLID or DASHED trace line styles. The author was not able to determine which display options require trace lines, and was further confused by the fact that the opposite style was listed from that which was specified with this command when the SPECIAL_OPERATIONS STATUS command was executed. If a user is interested in using traces lines, this would require further investigation.

The special operations menu allows the user to request EVALUATED_-SURFACES_DISPlays of the current object. These produces displays showing lines at constant parametric intervals along the object's precise surfaces. The lines displayed are parallel to the sides of each surfaces edges. The user can control the number of lines drawn across each precise surface with the DISPLAY_LINES command, and the number of line segments used to depict each display line, or curve, is controlled by the INCREMENT command. This display option can be SET_AS_GLOBAL.

The final SPECIAL_OPERATION presented in this menu is use to display the CONTROL_NET. The control net is used by the program to mathematically control the shape of an object's precise surfaces.

DISPLAY_ATTRIBUTES

In I-DEAS Rev. 4.1, the DISPLAY_ATTRIBUTES menu contains commands used to control the parameters that affect the appearance of objects displayed on the graphic screen. In Rev. V, these are located in the DISPLAY_OPTIONS, SURFACE_APPEARANCE menu.

The COLOR_OF_OBJECT command can be used to change the color of the current object. The user can specify any stored color by mane of number. If the user want to see a list of the available colors, a "?" can be entered to display the color DIRECTORY The default color will be the object's current color. This can be useful if several objects have been used to construct the current object using Boolean operations, which has resulted in the object having surface colors.

The PALETTE command presented in the DISPLAY_ATTRIBUTES menu is identical to that which was described earlier in this chapter along with the SHADED_IMAGES commands.

The SHADING_TYPE is used to control the variation of colors, or shades, across surfaces when shaded images are processed. There are four shading types offered, they are: SMOOTH, FLAT, UNIFORM, and CONTINUOUS. The on-line help provided by I-DEAS describes each of these levels of shading. But, experimentation with these options will more easily show the user the best shading type for an given circumstance.

The RADIANT_LIGHT command presented in the DISPLAY_ATTRIBUTES menu allows the user to specify the minimum amount of light that each surface of an object will radiate when a shaded image is processed. This value will be added to the ambient light that has been set in the DISPLAY_OPTIONS menu. But, the values set for radiant light will be store with the surfaces of the current object. This allows the radiant light to vary between stored objects, while the ambient light affects any object displayed.

The DIFFUSE_LIGHT command is used to specify the amount of the diffuse light, light that comes from the light source(s), that the current object's surfaces will reflect. The intensity of light that the user will see is dependent on the angle at which the surfaces are viewed relative to the position of the light source(s).

The combined effects of two commands, GLOSSINESS_(SPECULAR) and BRIGHNESS_(SPECULAR), control the apparent shininess of objects when they are displayed as shaded images. The GLOSSINESS command is used to specify the "specular coefficient". Specular light is reflected off of an object without picking up the objects color, like a mirror's reflections. Glossiness controls how much the specular reflection spreads as it is reflected. The higher the value of the glossiness parameter, the closer the reflections come to being reflected in a single line. The BRIGHNESS command is used to specify, as a percentage, the amount of specular light the object will reflect.

The appearance of LINE and HIDDEN_LINE displays are affected by the angular value set with the FEATURE_ANGLE command, when the FEATURE_LINE_SW is turned ON. This parameter is used by the program to determine which facets' edges are to be display. The edges between separate surfaces are always shown. But, the edges of facets that make up a singe surface will only be shown if the angle between the planes of adjacent facets is greater than the FEATURE_ANGLE specified. The default angle is 30^{o}, which will show no facets around a cylinder, because their common angle is 22.5^{o}.

I-DEAS offers the capability of displaying objects in shaded form which are transparent, allowing other objects to be seen through them. This is controlled by the value stored for each object's TRANSPARENCY_%, in the DISPLAY_ATTRIBUTES menu; and by turning ON the TRANSPARENCY_SW in the DISPLAY_OPTIONS, SHADED_IMAGE menu.

A similar effect to transparent objects is provided by turning the SHADED_IMAGE PRECEDENCE_SW ON, and storing a PRECEDENCE_% DISPLAY_ATTRIBUTE for each object. This does not actually make objects transparent. It make the objects appear in various degrees of visibility. They will appear in order from highest to lowest visibility based on their percentage precedence. Objects which are in front of other objects will block the view of the objects behind them. But, objects with higher precedence will appear brighter. This can be used to yield an increased sense of three dimensional depth to shaded images produced of multiple objects.

The DISPLAY_ATTRIBUTE menu provides three command that can be used to select bodies and surfaces from those displayed to and place them into a list of entities which can then be operated on with subsequent display attribute commands (e.g. blanking, and making attribute changes). This list will remain active until it is replaced by a new list, or until a non-DISPLAY_ATTRIBUTE command is executed. These commands include BLANK_SURFACE/BODY, BODY_SELECT, and SURFACE_SELECT.

PICTURE FILES

One of the powerful features offered by the I-DEAS program is the creation of "picture files". These are used to store display images in memory disk files for future display. This allows the user to setup and create display images, store them and them call them up in subsequent interactive sessions. When the images are recalled, no computer computations are required, and they will be displayed are quickly as the terminal and the host computer can transfer the data to the display screen. This allows the user to prepare images for review by others, at a later time, and to display them at a faster rate.

This also allows a user to create images on terminal, and them to display them on another terminal. This might be necessary if a color graphics copier is available only on one terminal for producing (paper) hard copies of the images. Or, this might be desirable if most work is done on an inexpensive terminals or PCs with low graphics resolutions, and a higher resolution graphics terminal is available for displaying images.

Three picture FILE_TYPEs can be created: Replay, Binary, and Formatted. Replay files can only be replayed on the same terminal type that was used to create them. But, they are the smallest of the three file types, and can be redisplayed the fastest. Binary files can be replayed on other terminal types than just the terminal used to create them. But, they are larger than, and slower than Replay files. Finally, Formatted files are the largest, and slowest of the three types. But, not only can they be redisplayed on any valid terminal type, they can also then as the input to drive a plotter. This requires an external program to act as a plot driver.

WRITE_FILE_SW is the toggle switch for turning the creation of picture files ON and OFF. Thus, if the user wishes to create a picture file, after specifying the desired FILE_TYPE, this switch is used to turn this feature ON. Then, any subsequent displays executed will cause the program to prompt the user for a file name in which the picture file is to be stored. The AUTO_NAME_SW command can be used to turn on a feature which will prompt the user for a base name with which subsequent picture file will be name automatically. If this is used, and the base name entered is "PICTURE", the files will be named PICTURE1, PICTURE2, etc. If a name is entered, either manually or automatically, which already exists, the user will be asked if it should be over written. If the user answers NO, the program will ask for another name. If automatic naming is in effect, the next name in the sequence will be tried. The default name is always NONE, which will yield no picture file. Once the desired picture files have been created, the WRITE_FILE_SW can then be used to turn this feature OFF. This will allow execution of displays being prompted for picture file names.

Once picture files have been created, they can be redisplayed by using the DISPLAY_PICTURE_FILE command. This will prompt the user for the name of the file which is to be displayed. If picture files have been created in the current I-DEAS session, the default file will be the last file created during the session. The user can also choose the DIRECTORY option, help in the selection of file names from those stored during the session. This requires the user to turn file tracking on with the ON_TRACKING command presented in the menu of DIRECTORY options.

APPENDIX XI

CAD DRAFTING OPERATIONS

INTRODUCTION

As discussed in Chapter 2, this work was done with I-DEAS REV. 4.1. But, as of this writing, GEA had installed I-DEAS REV. V. The DRAFTING Family has had at least one major change introduced between these two revisions. The version that will be discussed here requires that the geometry created in Solid Modeling be transferred into DRAFTING for the creation of CAD drawings based on solid models. If any changes are made to the solid models after drawings have been created, the user must manually update the drawings with the commands provided in DRAFTING. Or, the geometry must be deleted from the drawings, and then the new geometry must be transferred into the drawings, and re-detailed. But, with Rev. V, I-DEAS provides the capability for automatically incorporating the solid models' changes into the drawings. This requires study of the Rev. V system. But, this change appears to be very useful, and should be investigated by future users.

GRAPHICS TERMINAL REQUIREMENTS

I-DEAS DRAFTING requires the use of a mouse or tablet (even though the Tektronix terminal may have a joy pad, that works in all other I-DEAS Families). But, if other I-DEAS Families are used before entering DRAFTING, a non-mouse code can be used, and the user will be prompted to change the code when entering DRAFTING. Therefore, if Solid Modeling is used first, T4107 might be used, and then switched to T4207A when entering DRAFTING.

INTERFACING WITH I-DEAS DRAFTING

The interface to DRAFTING is menu-drive, similar to the menu system used in the other I-DEAS Families and Modules. But, unlike the other Families' menus, which scroll vertically over the graphics display, DRAFTING's menus are confined to a short box that spans the width of the bottom of the display screen, and the graphics views are displayed in a large area above the menus (see Figure 61.). Also, rather than scrolling, DRAFTING's menus "pop" into the display's menu box, as the user picks various commands. The menu box is subdivided into three smaller boxes, containing sub-menus. The bottom, single text line, sub-menu box is the Global Menu Area. The largest box is the Prompt Area, or command menu area, in which the various levels of command menus will be displayed. During the execution of most commands, this large menu box will be divided, creating an Option or Selection Area at the right side of the screen. This sub-box will present the options that become available during the execution of DRAFTING's commands.

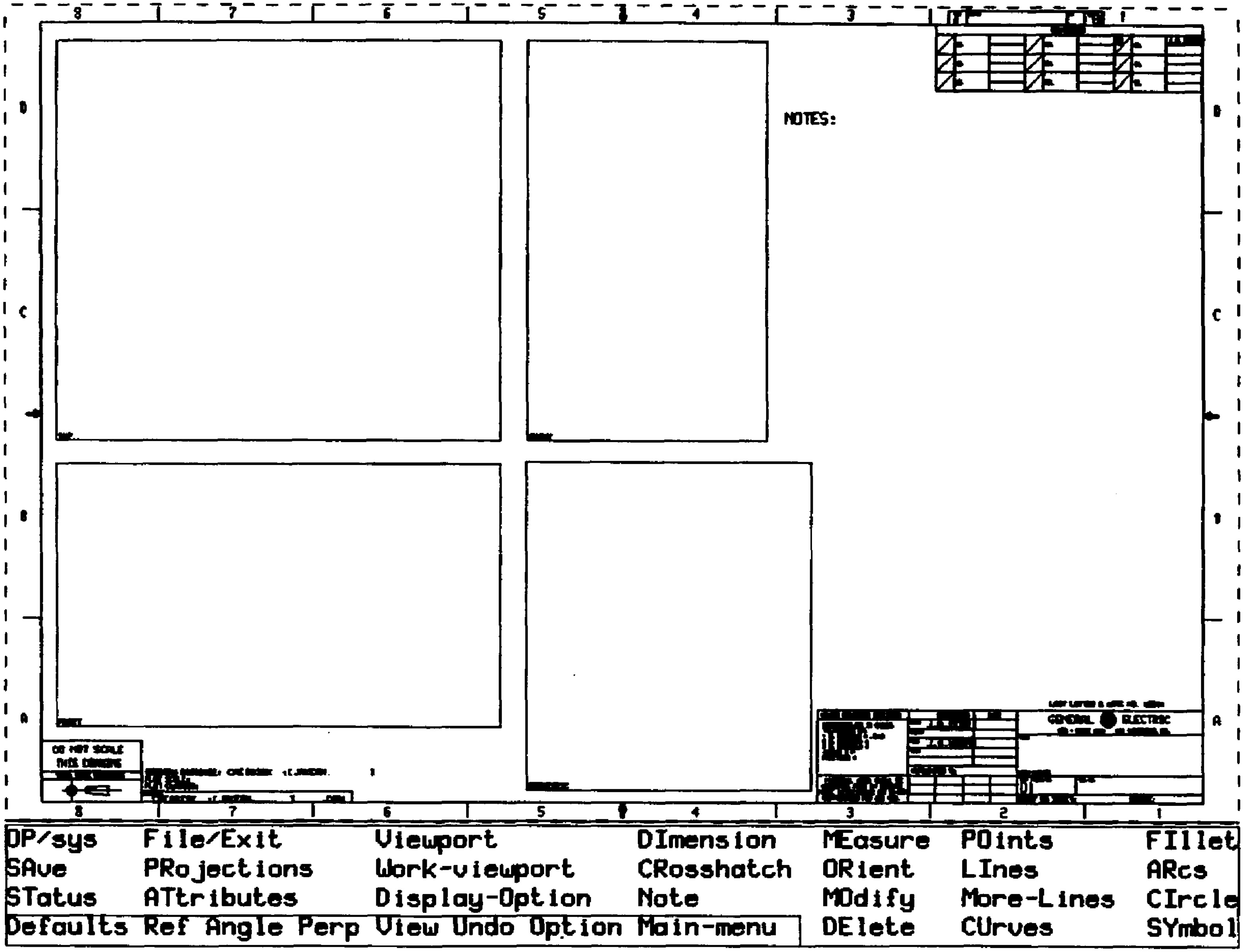


FIGURE 61 . - I-DEAS DRAFTING DISPLAY

In DRAFTING, most commands can be executed by typing in the appropriate mnemonic identifying the desired command. But, unlike discussed in Chapter 2 for the other I-DEAS Families, the commands executed in DRAFTING are generally "picked" or "digitized". This is done using the terminal's digitizer tablet or mouse to move the screen cursor to point at the desired selection, and then pressing the appropriate key to "<DIG>" (digitize) the selection.

In addition to <DIG>ing commands, the DRAFTING system often requires the user to respond with the <RETURN> key, and to abort a command's execution, the user must use the <ESC> (escape) key. Because many I-DEAS users will be using a Tektronix terminal with a Tektronix digitizer table that users a four key "puck" to move the screen cursor, DRAFTING provides the convenience of programming three of the puck's keys to serve as <DIG>, <RETURN>, and <ESC> keys.

DRAFTING COMMANDS

DRAFTING's commands are so similar to those that are discussed in APPENDICES VIII, IX, and X, that few need to be described here. But, there are a few very powerful capabilities that should be mentioned. At most steps in the execution of DRAFTING commands, and at any level within its menu structure, the user can pick the options shown in the Global menu area. This allows the user to change views, use an on screen calculator, change defaults, and access the MAIN-MENU, etc. without terminating a command, or actually returning to the MAIN-MENU. Once the desired functions are completed, <ESC> can be used to return to the same step in a command's execution, or position in the menu structure.

If the user enters an <ESC> during the selection process while executing a command, the selection process is terminated, and the command is executed. For example, if a user is selecting entities for deletion and uses <ESC> after having <DIG>ed the wrong entity, that entity will be deleted. But, DRAFTING provides an option that make it more forgiving of mistakes like this than the other I-DEAS Families do. The UNDO command located in the Global Menu area allows the user to undo the last command's effects after it has been terminated. This command can also be used repeatedly to undo several of the last commands executed. DRAFTING even allows the user to undo the effects of UNDO. This is done by pressing <CONTROL> together with the "R" key. Thus, UNDO allows users to experiment with DRAFTING's commands, without risking making mistakes that become permanent, freeing them from constantly referring to a user's manual.

VIEW COMMANDS

The VIEW commands used in DRAFTING operate like they do in the other I-DEAS Families. Except a few are specific to DRAFTING. These include ZOOM-ALL, which fills the entire graphics area with the drawing viewport (i.e., the entire drawing). The ZOOM-VIEW command is used to fill the graphics area with one of the drawing's viewports. The ZOOM-WINDOW command enlarges a specified portion of the visible part of the drawing to fill the graphics area. The ZOOM-IN and ZOOM-OUT commands work like the other I-DEAS Families' MAGNIFY command. But, rather than using a decimal fraction to specify a magnification that will shrink the current display, making more of the drawing visible, the ZOOM-OUT command is used to do this with values greater than 1. Therefore, in DRAFTING, ZOOM-OUT with a factor of 2, is equivalent to using a magnification factor of .5 in Solid Modeling. DRAFTING also has an added PAN command. This can be used to maintain the current level of magnification, and to move the drawing across the graphics display area, by specifying two screen locations from which to pan from and to.

VIEWPORTS

I-DEAS DRAFTING utilizes multiple viewports on a drawing to separate the entities created for each view of the geometry. When a drawing size is specified, one "drawing viewport" is automatically created, which covers the drawing's entire "sheet of paper". But, the user should create smaller viewports on this overall viewport into which views, and notes, etc. can be created. When entities are created in DRAFTING they are placed in the "working viewport". The WORK-VIEWPORT command located in the MAIN-MENU is used to pick the current working viewport.

In order to see the borders of viewports, the commands located in the DISPLAY-OPTIONS menu can be used to turn ON and Off the boarders, and their names. The current working viewport's boarder will be shown with a different line style then the other viewport's. Thus, when the boarders are visible, the user can see which viewport is current.

CLASSES, COLORS and GROUPS

Like any good CAD package, DRAFTING provides multiple "classes" in which entities can be created. The user can separated different types of entities, dimensions, notes, etc. into different classes, and to selectively make classes visible or invisible. DRAFTING provides 256 transparent classes from which to choose. Any new entities created are placed on the "current" or active class. This is determined by the DEFAULT, CLASS command. The visibility of a class is determined by CLASS command located in the DISPLAY-OPTIONS menu.

In addition to using classes to segregate information placed on a drawing, different colors can be used. Unlike some CAD software, I-DEAS DRAFTING allows the use of multiple colors on any given class. It also allows the user to turn ON and OFF the visibility of colors on all visible classes. These options are controlled by the DEFAULT, COLOR, and DISPLAY-OPTIONS, COLOR commands like those described above for classes.

DRAFTING also allows entities to be created with a GROUP-NAME attached to them. This also groups of entities to be specified when making selections from within DRAFTING commands.

PREPARATION OF SOLID MODELS FOR TRANSFER TO DRAFTING

Because this thesis used the solid models' geometry for the creation of CAD drawings, some preparation of the solids in Solid Modeling was needed before they were transferred into DRAFTING. This preparation involves setting up the views of the geometry that are desired in the drawings, and setting the display parameters desired for these views. The DRAFTING system is 2D. The geometry transferred for creation of drawings is "flattened" onto a 2D plane, and therefore, it can no longer be manipulated in three dimensions. Thus, the views must the prepared to appear in the standard drafting style. This generally requires top, front, side, and isometric views of the parts. In these views, any hidden lines are usually shown as dashed lines, except for in isometric views, in which they are usually removed from view. I-DEAS

Solid Modeling allows the user to use the software to make this happen, rather then transferring all lines into DRAFTING as solid lines, and then manually changing the hidden lines to dashed or removed lines.

SETUP OF VIEWS

The views can be setup one at a time, using a single viewport on the graphics screen. But, a more efficient method is to utilize I-DEAS multiple viewport capabilities to prepare all views of a complete solid on the screen simultaneously. This allows them all to be transferred to DRAFTING together. Then if any section views are needed, they can be setup and transferred separately.

Solid Modeling's multiple viewport options are discussed in Chapter 2. Multiple viewports are setup with the VIEW, LAYOUT command. Four layouts are provided by I-DEAS, containing from one to four viewports. The user can also create custom layouts.

For this thesis, all four standard drafting views were used in creating CAD drawings of the new surface unit support's two support pieces, its center trim, and the assembly of the complete support. Thus, the I-DEAS standard layout named "FOUR, EQUAL SIZED [VIEWPORTS]" was used when setting up the views for transfer of geometry to DRAFTING. To cause I-DEAS to prompt the user to select the affected viewport for use with any subsequent display option or view changes, the VIEW, AFFECTED_VIEWPORTS, PICK_VIEWPORT command was executed.

After the four viewport display layout was established, each of the views were created. This was done by applying the VIEW, EYE_DIRECTION and ROTATE_ABOUT_MODEL commands to establish top, front, side, and isometric viewports. After each viewport's view was setup as desired, they were stored with the VIEW, STORE command using these four names, for future use.

SETUP OF DISPLAY OPTIONS

After the views were setup, the next step in preparing the solid model geometry for transfer to DRAFTING was to setup the Display Options that are needed to produce the desired hidden line effects. This is accomplished by using the DISPLAY_OPTIONS, SURFACE_BOUNDARY_IMAGE commands. These are discussed in Appendix X. But, the specific parameters that were set, were done by: 1) setting the HIDDEN_LINE_-STYLE to DASHED for all four viewports, and then to REMOVED for the isometric viewport; and then, 2) turning the PRECISION, SILHOUETTE_-ADJUST_SW, ON for all viewports.

WRITING OUT TRANSFER FILES

The transfer to DRAFTING is made by writing IGES formatted files out of Solid Modeling, and then reading then into drawings in DRAFTING. This is done once the viewports and display options are setup as desired. It is accomplished by using the MANAGE, WRITE, DRAFTING command. This command prompts the user for specification of: the curve error factor; IGES file units (INCHES); file identification text, optional (20 lines of 72 characters maximum); and the IGES file name. If the use specifies a new file name, I-DEAS will ask if it is "OK to write new file"?. But, if the user enters an existing file's name, I-DEAS will ask if it is "OK to overwrite existing file?". It will also ask if it "OK to display?". If the user answers YES, I-DEAS will then ask if it is "OK to erase screen?". This allows the user to preview the displays that are written into the IGES files. This enables the user to see if the hidden line style, and level of precession looks OK. Thus, if the displays look unacceptable, the user may be saved from spending time reading a bad IGES file into DRAFTING.

PREPARING DRAFTING DRAWING TO RECEIVED GEOMETRY TRANSFER

When DRAFTING is entered, the user is asked to "Enter drawing name or select new drawing size:". When preparing a new drawing for receipt of the geometry from Solid Modeling, the user will specify a drawing size at this point. The drawing size options are: the ANSI Standard English drawing sizes: A, B, C, D, and E; standard Metric drawing sizes: A4, A3, A2, A1, and A0; and EXPLICIT. If an English drawing size is used all dimensions entered will be in inches (e.g., 3 used as the value for a line's length will be 3.0 inches long). But, if a Metric size is used, the drawing's size will be the same as the English size. But, when dimensions are read in from Solid Modeling transfer files, or when geometry is created from within DRAFTING, they will be in millimeters. The user must use this cautiously, because once millimeter have been established, they can not (easily) be changed to inches. Actually, the drawing viewports can be SCALED back into inches. But, the system will continue to think millimeters are being used. The EXPLICIT option allows the user to directly specify a custom drawing size by entering its boarders' dimensions in inches.

For this thesis, all four of the drawings created were made "D-size". This large drawing size was not actually needed for all of these drawings. But, as will be discussed later, only a D-sized GEA drawing format (i.e., boarder, title block, revision blocks, etc. on the

CAD screen, same as on drafting paper) was available for use in I-DEAS DRAFTING at the time this work was done. So, by specifying D-sized drawings, no drawing format had to be created, and the drawings could easily be plotted, or duplicated in smaller sizes using the equipment available at GEA if desired.

Once a new drawing size is specified, DRAFTING sets up the display screen with a "drawing viewport" that is proportional the size of the drafting paper size that was specified. The next step the user must complete in preparing to read IGES files into a drawing is to setup viewports on the drawing viewport into which the transfer files' views can be read..

DRAFTING's viewports can be modified once they are established. But, it is helpful if they are setup as near to the ideal as possible up front, to avoid additional work later. Because the drawing format requires additional time to display each time different views of the drawing are displayed, generally it will save the user time if the drawing format is not added to the CAD drawing until one of the last steps in creating drawings. But, this means that the user must visualize were the drawing format will require space on the overall drawing viewport, as well as were the user will need to place notes, etc., so that no viewports will be setup for the geometry and dimensions etc. in these areas. An alternative to this, is to bring the drawing format into the drawing, on a class by itself, and then making that class invisible. Or, the drawing format's entities can be BLANKED.

The viewports are setup with the VIEWPORT, DEFINE_VIEWPORT command. This prompts the user to define the view's boarders. This is usually done by picking diagonal corners for the viewport using the screen cursor. But, this command also presents other options in the sub-menu that is display for use during the execution of this command. Once the boarders are defined, this command prompts the user to specify the view's origin. Again this is usually done by picking a location on the screen with the cursor. Once the origin is established for a viewport, the user is asked for a "view angle" for the viewport. This becomes a label that can be displayed in the corner of the viewport. The user is provided with the following view angles from which to choose: TOP, ISOM, LEFT, FRONT, RIGHT, and BACK. Having made this last selection, the execution of this command is terminated. If additional viewports are to be defined, this procedure must be repeated.

After the geometry viewports have been established on the drawing viewport, their boarders and origins can be aligned to present the geometry in the standard drafting style in which the top, front, and side views are projections of one another. This is accomplished with the use of the VIEWPORT, ALIGN, MATCH-BORDER, and ALIGN-ORIGIN commands. The MATCH-BOARDER command allows the user to select the viewport whose boarders are to be changed, and the viewport whose boarders are to be matched. The user then specifies whether the boarders are to be matched HORIZONTALLY (moving the horizontal boarders vertically), or VERTICALLY (moving the vertical boarders horizontally). The ALIGN-ORIGIN command works in much the same way, aligning one viewport's origin with another's, either VERTICALLY or HORIZONTALLY. But, this is generally done after the geometry has been read into them, so that it can be best viewed.

STORING DRAFTING DRAWINGS

DRAFTING operates like all other I-DEAS Families. All work is done with a "current" (temporary) drawing that exists only in computer RAM. It is not saved permanently in a disk file unless the user commands it to be written to a file. Thus, the user should SAVE the current drawing, before retrieving another drawing, or before leaving DRAFTING, if he want to save the current drawing.

Once the basic drawing has been prepared, it is a good idea to save the drawing at this point. In fact, if several drawings are going to be created for similar parts (e.g., the two pieces of the new support design for this thesis), by saving the drawing at this point, a generic drawing can be setup that can be saved under two names, one for the specific part that is being detailed, and one for use with other parts. This can also be done by completing one drawing, and saving it in two files, and then deleting the geometry from one and transferring in another part's geometry. This will require the user to create the drawing format, notes, tolerances, signature blocks, etc. only once, saving a lot of time. Drawings are saved with the SAVE command located in DRAFTING's MAIN-MENU.

Once drawings have been created in DRAFTING, they can be gotten from the user's disk file directories. When entering DRAFTING this is done with the GET-DRAWING-FROM-FILE option when asked to "Enter drawing name or select new drawing size:". If the user wants to change to another drawing from within DRAFTING, the FILE/EXIT menu's GET-DRAWING command is used.

READING THE TRANSFER FILES INTO DRAFTING

When the user has established a drawing's viewports as desired, the geometry that was written out from Solid Modeling can then be read into DRAFTING. This is done with the FILE/EXIT, FROM-GEOMOD command. When this command is invoked, the user is asked for the Solid Modeling transfer file which is to be read. Like all DRAFTING commands which require alphanumeric input, DRAFTING presents a set of the alphanumeric characters along with RET and ESC in the Options menu area. This allows the user to use the mouse or tablet to drive the screen cursor to <DIG> the characters. This is fairly tedious. So, the users can use the keyboard to type in the file name. The full name, including the extension is required (e.g., FILE.IGE).

Once the file name is entered, the user is prompted to specify the viewports into which each of the views in the transfer file are to be placed. This can be done by <DIG>ing each of the viewports, in order of the Solid Modeling viewport numbers, to place the geometry into the corresponding DRAFTING viewports into which the geometry is to be placed. The user must be careful to <DIG> only once in each viewport. Otherwise several of the Solid Modeling views may be placed together in the same viewport. If this happens, the user will have to call up the empty drawing again, or deleted the geometry in the viewports, and then transfer the geometry in again.

Once the geometry have been transferred into the drawing (and the drawing has been SAVEd, if it is acceptable) the user may need to align the origins of the viewports are described previously.

DETAILING DRAWINGS

After a drawing has been setup and its geometry has been transferred in from Solid Modeling, DRAFTING's commands to detail the drawing. The users can easily apply DRAFTING's menu driven entity creating, dimensioning, crosshatching, and note and symbol creation commands to complete the creating of a drafting style engineering CAD drawing. The majority of this commands are very easy to use, and are located by selecting command menus from DRAFTING's MAIN-MENU.

GEOMETRY CREATION

Any additional geometry which the user must add, or create from scratch if desired, can be created by picking one of the following create command menus: POINTS, LINES, MORE-LINES, CURVES, FILLET, ARCS, and CIRCLE.

The user must remember to establish the desired viewport as current with the WORK-VIEWPORT command in order to create any entities in a specific viewport. If this is not done, the entities created will most likely end up in the overall drawing viewport. This may appear OK, but if the user tries to MODIFY, ORIENT, or change their ATTRIBUTES, problems will arise. The user must also remember to utilize the DEFAULTS command if CLASSES, COLORS and GROUPS are to be used to segregate the entities created.

Very few of the create commands require any description do to their straight forward operation. But, first time and infrequent DRAFTING users may have difficulty locating the DRAFTING-LINES commands. These are found by selecting MORE-LINES, followed by DRAFTING-LINES. This will yield an options menu which allows the user to create: CIRCLE_-

CENTERS; CENTER_LINEs; CUT-PLANE lines, with either EVEN- or UNEVEN-SPACING; and LONG-BREAK-MARK lines.

DRAFTING provides all of the functions that can be accomplish on a drawing board with a drafting machine and standard drafting tools, like a compass, ruler, templates, and an eraser.

The POINTS command allows users to create points at explicit point locations. This can be done by specifying absolute coordinates (X and Y) for the points. DRAFTING also allows the points to be created at locations relative to REFerence points, either with incremental coordinates (IX and IY), or in polar coordinates (LNG and PA).

POINTS also allows the user to create points at implicit point locations. These include <DIG>ing: a screen location (DIG); the END of an existing entity; the center of a line or arc (CTR); and existing point (PNT); a point on an existing line or arc (PNT-ON); at the intersection of and existing line or arc with another line or arc (INT-OF); and by interpolation between two existing or implied points, or at a distance along a line or arc, from the end picked (%OF). Each of these point specification options is also available when specifying implicit points for the creation of other entity types from within the other creation commands.

The general LINES creation command allow the user to create lines from one point (either explicit or implicit) to another. The command allows multiple lines to be created consecutively, each starting at the end point of the previous line. The command continues until it is terminated with an <ESC>, or a <RETURN> to the MAIN-MENU.

With MORE-LINES commands, the user can create HORIZONTAL, VERTICAL, and ANGLED-LINES. Lines can be created PARallel and PERPendicular to other lines. Also, lines can be created that are TANgent to: two arcs or circles (TAN/TAN); a point and an arc or circle (TAN/POINT); an arc or circle at an angle (TAN/ANGLE); an arc or circle and perpendicular to an existing line (TAN/PERP).

The CURVES menu provides create commands that are capable of producing: OFFSET-CURVES; MULTI-LINES; SLOTS; n-sided polygons inside an arc (POLYGON-IN-ARC); hex-heads based on their center point and their bolt head size (HEX-HEAD-BY-SIZE); hex-heads based on their center point and their bolt head's flat length per side (HEX-HEAD-BY-FLAT); GENERAL-ELLIPSEs; curves fitted smoothly through any number of user selected points (B-SPLINE-CURVE); n-sided polygons based on their center point and the number of sides (GENERAL-POLYGON); BOXES-BY-WIDTH/HGT; and rectangles based on two diagonal corner points (BOXES-BY-TWO-POINTS). The SLOT command can be used to create slots of specified length, width, and placed at a specified angle. The resulting slots will have parallel sides with semicircles at the ends (their diameter being the slots' width).

The ARCS and CIRCLE commands provide several options when creating these entities. They can be based on radius or diameter dimensions, depending on whether the RADIUS or DIAMETER command has been specified (which affects all subsequent create commands). The RADIUS and DIAMETER commands also allow the user to enter the dimension directly; or, to select an existing arc's dimension (EXISTING-RAD(DIA)), to add to add to an existing arc's dimension (RAD(DIA)-PLUS-INC). Arcs and circles can be based on center (CTR) points, or through three points (PNT). ARCS can be created in either clockwise (CLW) or counterclockwise (CCLW) directions, depending on: picking either CLW or CCLW; the center point selected; or the sign of the angle specified (which affects all subsequent create commands). The CIRCLE commands allow the user to create circles that are tangent to another existing entity. Multiple circles can be created concentrically around a single center point with different edge points (CENTER/POINTS(S)); or with the same edge point (from the first circle created) around different center points (CENTER(S)/-POINT). Similar commands can be used to using RADII or DIAMETERS in place of the edge points. Circles can also be INSCRIBED-IN-THREE-LINES.

Most create commands can be used repeatedly after the execution of their operation is complete, without the need for executing the command again. This is done by entering a <RETURN> when a commands operation is finished. If the user wants to terminate a command, an <ESC> should be entered. This will return the user to that create command's menu. To return to the main menu, the user must select MAIN-MENU.

ANNOTATING DRAWINGS

DRAFTING has three separate menus that are accessed from the MAIN-MENU for the annotation of drawings. These are: DIMENSION, NOTE, and CROSSHATCH. The DIMENSION menu contains commands that allow the user to place dimension on a drawing's geometry using standard drafting practices. DRAFTING provides a dynamic dimensioning system (which can be turned off if desired with the DYNAMIC-ON/OFF command). This allows the user to <DIG> geometric locations between which dimension are desired, and the location in which the dimension should be placed. Then the program will display the resulting dimension. Until the user enters an <ESC> or <RETURN> is entered, the location of the dimension, and its parameters can be changed simply by <DIG>ing the new dimension location, and selecting commands from the active options menu. Dimensions are created and stored as single entities with multiple parameters associated with each. In addition to allowing dimensions to be modified dynamically while they are being created, this also allows them to be changed at any time, with this same level of flexibility. This is done with the EDIT-DIMENSION command.

DRAFTING provides DIMENSION commands for the creation of ANGULAR, RADIAL, and DIAMETRICAL dimensions. It also allows linear dimensions to be created horizontally (HOR-LINEAR), vertically (VER-LINEAR), and between two points at any angle, displayed with their dimension lines shown parallel to the line between the points (PAR-LINEAR). Linear dimensions can be created by continually selecting pairs of points (explicit or implicit), between which dimensions are to be placed. This is the case when the STANDARD command is in effect (default). The user can select one point to act as a datum, and then all other points that are to dimensioned from this point. This is the case when the DATUM command has been applied. Finally, when CHAINED has been used, after one dimension is set-up its end point becomes the start point the next dimension.

In addition to providing commands for the creation of dimensions, the DIMENSION menu also provides commands used to created, LABELS, BALLOONS, and FEATURE_CONTROL-SYMBOLS. LABELS can be created with any desired text, with a leader and arrow pointing to any location on a drawing. If dynamic dimensioning is on, labels can be moved and modified just like any dimension. BALLOONS are similar to LABELS, except their text is surrounded by a circle, and their leads do not include arrow heads. I-DEAS DRAFTING provides three FEATURE-CONTROL-

SYMBOLS sub-menus, which are: GEOMETRIC-TOLERANCES, DATUM-TARGETS, and SURFACE-FINISH-MARKS. Each of these up sub-menus contain commands that can be used to place a large number of predefined symbols at any desired locations within the WORK-VIEWPORT.

The NOTE command allows DRAFTING's users to easily added notes, and titles, and general text anywhere desired in the active WORK-VIEWPORT. The NOTES command provides an options menu which allow the user to control the HEIGHT, SLANT, justification (JUSTIFY), WIDTH-RATIO, DOWNSPACING, and TEXT-ANGLE for each note created. Once one line of a note is enter to the user satisfaction and the <RETURN> key is depressed, the program will place the note at that location. It will then drop the reference location down by the value of the DOWN-SPACING parameter, and prompt the user for input of the note's next line of text. This process of defining, placing, and then entering <RETURN> to add additional lines will continue until the user issues an <ESC>.

The CROSSHATCH command can be used to crosshatch any areas on a drawing which are bounded completely by closed loops of curves. These areas can include "holes" which are not to be crosshatched. DRAFTING provides standard crosshatch patterns for: IRON, STEEL, LEAD, ALUMINUM, GLASS/SLATE, RUBBER/PLASTIC, and BRASS/COPPER. The user can also create USER-DEFINED crosshatching patterns, including their angle, font, color spacing, and offset. The standard patterns can also be easily used with a REVERSE crosshatching angle, or user defined DISTANCE between its repeat patterns. Once the user has identified the pattern that is desired, an <ESC> is entered, and DRAFTING will prompt the user to select the entities for the outer boundary and optional interior holes. The the boundaries have been identified, a <RETURN> will cause the program to crosshatch area defined. If the results are unacceptable, the user can use the UNDO command to erase the crosshatching, and then redefine the pattern and try again. Note that crosshatching works pouring paint into a container, if there are any breaks in the container (boundaries), the paint (crosshatching) will spill out! This can result in the entire WORK-VIEWPORT being crosshatched. Remember UNDO if this occurs.

CREATING AND USING SYMBOLS

Symbols are collections of geometry and/or text which are stored together. They can be used repeatedly in the drawing in which they are created, and they can be written out to external files for use in other drawings. DRAFTING's symbol capabilities are accessed with the SYMBOL command. It presents a menu that allows the user to: CREATE-SYMBOLs; CREATE-INSTANCEs of the symbols created; change-all-instances of a symbol; STORE, RETRIEVE, and LIST symbols; and to "SMASH" symbols.

The CREATE-SYMBOL command asks the user to specify an alphanumeric name for the symbol. It then prompts the user to select the entities that are to be included in the symbol. When the last entity has been picked, <RETURN> is entered to terminate the selection process. Then the program asks the user to specify the location which is to be uses as the symbol's origin. Once a symbol is created, it can only be selected by its origin. Therefore, a permanent point or feature of the symbol should be specified as the symbol's origin. the entities selected in the creation of a symbol does not become an instance of the symbol. If the user wants to convert it to a symbol, it can be deleted, and then the symbol can be instanced at the same location.

Once symbols have been created or RETRIEVEd into a drawing file, they can be placed at any location in the work-viewport desired through the use of the CREATE-INSTANCE command. The user can use the commands presented in this command's options menu to set an angle and scale for the instances of the symbol, and to mirror the symbol across its X or Y axis (which is done before any scaling and rotation). Once these options have be set as desired, the user has two options for placing the symbol that is specified for instancing. With the ORIGINS(S) command the user can specify screen locations for the instances consecutively, by continuing to specify instance origin locations without entering a <RETURN>. Once a <RETURN> is entered, the instances of the symbol will be created, placing the symbol's origin at the instance origins specified. If the user enters an <ESC> before a <RETURN>, the command's operation is aborted without creating any instances. The other option DRAFTING presents for the placement of symbols (if supported by the terminal being used) is to DRAG-TO-ORIGIN. This allows the user to move a copy of the symbol around the drawing dynamically with the mouse or tablet. Once the symbol is located as desired, <RETURN> will cause an instance to be created at that location, and <ESC> will abort the command's operation.

The CHANGE-ALL-INSTANCES command can be used to change the entities that comprise a symbol, and then replace all current instances of the old version of the instance with the new version. Other than the facts that this command requires the user to specify the name of an existing symbol, and that it replaces existing symbol instances, this command operates just like the CREATE-SYMBOL command described above.

The STORE command is used to store symbols externally from the active drawing file. The RETRIEVE command is used to bring symbols into the active drawing file which have been created and stored when they were created in other drawing files. The D-sized drawing format (FORMAT-D.), that was used for this thesis is a good example of a symbol that has been created and stored for use in other drawing files. Other examples of commonly used symbols that the user might want to create and store are: "critical-to-quality" symbols; electrical schematic symbols, flow chart control symbols, ladder logic symbols, etc. Some of the mechanical engineering drawing type symbols have been provided by I-DEAS in the DIMENSION, FEATURE-CONTROL-SYMBOLS menu.

The SYMBOLS, SMASH command can be used to convert symbol instances back into individual entities for use as points, lines, curves, etc.

ADDING THE DRAWING FORMAT TO DRAWINGS

A D-sized drawing format had be created be GEA's CAD department for use in I-DEAS DRAFTING as a symbol. It contains the GEA drawing format used to add a title, drawing number, signatures, revision marks, etc. to drawing created at GEA. To use this format for this thesis, the symbol file, named "FORMAT-D." was copied into the disk directory in which the DRAFTING drawings were being created for this thesis. Then the symbol was RETRIEVEd into each of the drawings' file, and then it was instanced into the main drawing viewport with the CREATE-INSTANCE command. The figures, appearing later in this chapter, displaying the drawings created for this thesis show the inclusion of this drawing format.

TRANSFER FILES

Geometry is transferred to Calma (or any other external CAD system) by writing out IGES formatted files. I-DEAS provides three types of IGES files from which to choose. The first is a standard IGES file (IGES). The other two have been defined specifically for use by: Calma DDM Rev. 3.1.1 or higher or, COMPUTER VISION.

WRITING OUT TRANSFER FILES FOR USE IN CALMA

The transfer files are written up using the commands in Solid Modeling's MANAGE, WRITE menu, which can be assessed by typing:

/MA W <RETURN>.

Once the WRITE menu has been accessed, the CALMA_IGES command is selected (C).

There are four types of files that can be written out for use by Calma DDM. FACET_EDGES can be written out resulting in a file containing lines that lay on the current object's facets, and no curves or surfaces are written. VIEW_DEPENDENT files can be written which will result in transferring the view dependent geometry that are visible based on the current view and surface boundary display parameters. The geometry written out will look like a SURFACE_BOUNDARY_DISPLAY of the current object (see Chapter VII). The WRITE, CALMA_IGES command will also allow the user to select and write out IGES formatted files of any PROFILE stored in the current model file. The file type that is desired for transferring complete solid models geometry is named CURVES_AND/OR_-SURFACEs.

The CURVES_AND/OR_SURFACEs file type allows the user to write out the 3D curves from the intersections of the current object's precise surfaces, and/or its surfaces themselves. If Calma DDM is going to be used to create Calma solid models, the surfaces might be used. This has never been attempted by the author. But, if the geometry is being transferred for use in producing production engineering drawings only, the surfaces will not be used, and may actually cause problems for the CAD Designer. Therefore, when I-DEAS solid models are being transferred to GEA's CAD Department, only curves should be transferred.

The specification of the desired entities for output are set with the use of the SET_IGES_PARAMETERS (P) command. This allows the user to set the OUTPUT_GEOMETRY to: CURVES, SURFACES, or BOTH_CURVES_AND_-SURFACEs. The user must also specify the desired UNITS, which can be either INCHES or MILLIMETERS. The final IGES parameter that must be set is the CURVE_ERROR_FACTOR. This factor can be set to any integer between 0 and 10, with the default being 5. Before issuing the EXECUTE command, the user can use the STAtus command to be sure that these parameters have be set to the desired values.

Once the user has issued an EXECUTE command to write out an IGES transfer file, I-DEAS will prompt the user for input of text for the start section of the file, if desired. Then the user is asked for the IGES file name (e.g., CTRIM.IGE) that is to be used. If no file exists with the name specified, the user will be asked if it is OK to write a new file. If a file already exists with the name specified, the user will be asked if it is OK to overwrite that file.

APPENDIX XII

I-DEAS PREPROCESSING OPERATIONS

I-DEAS FINITE ELEMENT ANALYSIS

The I-DEAS Engineering Analysis Family contains commands which can be used to build and preform analyses on finite element, plastics mold filling and cooling, and frame models. For this thesis neither the plastic molding or the frame analysis Modules were used, and they will not be discussed any further. But, the finite element Pre/Post Processing Module was used extensively for building FEA models of the new and old supports based on the solid models produced of them in I-DEAS Solid Modeling, Object Modeling Module. Also, the Pre/Post Processing Module was used for display of the model solutions which were run with ANSYS programs.

TYPE OF FEA MODELS AND ANALYSES TO BE CONSTRUCTED AND PERFORMED

Before the finite element models are constructed, it is necessary to decide what type of FEA models and what type of finite elements are be best suited for the analyses that are to be performed.

ANSYS is capable of performing: THERMAL, STATIC, LINEAR BUCKLING, MODE-FREQUENCY, FULL HARMONIC RESPONSE, NONLINEAR TRANSIENT DYNAMIC, REDUCED LINEAR TRANSIENT DYNAMIC, REDUCED HARMONIC RESPONSE, and SUBSTRUCTURE GENERATION or STRESS PASS analyses. The specification of the analysis type is done in ANSYS with the "KAN" command.

The ANSYS finite element software contains an extensive finite element library from which the element type(s) could be chosen. These include POINT or matrix, LINEAR or line, AREA, and VOLUME elements, which are either 2D or 3D elements. The ANSYS^{R} Engineering Analysis User's Manual[18] lists the elements that are available, and the input and output data, theory, assumptions associated with each. The specification of the ANSYS element type (which is done with its "ET" command) is discussed in Appendix XIII. In I-DEAS the default element type is setup by specifying the family, order, and topology for the element type desired. Once the default element type is specified, all subsequent elements created are of this type.

MESH AREAS AND MESH VOLUMES

Mesh areas are bounded planner areas in 3D space which the I-DEAS program can automatically mesh, by dividing them up and creating finite elements on them. When models are created with VOLUME elements, rather than AREA elements, MESH_VOLUMES are required. Mesh areas (and mesh volumes) can be created directly. But, in order to take full advantage of a common database throughout this "Mechanical Computer Aided Analysis", I-DEAS allows the geometry created in Object Modeling to be

[18] Gabriel J DeSalvo, Ph.D., Robert W. Gorman, ANSYS Engineering Analysis System User's Manual, Swanson Analysis Systems, Inc., Houston, Pennsylvania, 1987, Appendix C.

used in Engineering Analysis. It allows mesh areas to be constructed through the selection of closed-loops of geometric curves.

Two types of mesh areas are available within I-DEAS Pre/Post Processing for use in automatic mesh generation. The first are called FREE_MESH areas. They require a large amount of computer computation during automatic meshing. But, free mesh generation automatically determines the transitions between the elements created in adjoining mesh areas, and can create irregular mesh patterns. It requires little segmenting of models for automatic meshing. But, care must be taken in planning the mesh areas to prevent the creation of overly distorted elements, which can lead to errors in the FEA solutions.

The second mesh area type is called "MAPPED_MESH". Mapped mesh generation requires less computer computation during automatic mesh generation than free mesh generation. It can not produce irregular mesh patterns; and it requires manual creation of transitions between mesh areas. It also requires considerable segmentation of the models.

GEOMETRY DEFINITION

Since mesh areas are constructed through the selection of closed-loops of geometric curves, it was necessary to define the geometry from which these curves were selected. The geometry of the I-DEAS solid models are often too complex for direct use in the construction of FEA models. SDRC anticipated these types of problems, and therefore, I-DEAS uses working sets, rather than solid models, for the geometry used to aid the user in constructing mesh areas and volumes, used to automatically mesh FEA models. It also allows working sets to be shared by Solid Modeling, System Assembly, and Engineering Analysis.

Working sets can be created in I-DEAS Solid Modeling Family for use in Engineering Analysis. But, they can also be created and modified within Engineering Analysis. Engineering Analysis' commands are capable of creating new working sets based on the edges and vertex points of complete solid models, or on their surfaces.

The first step used in the construction of working sets from existing solid models is the creation of the geometric framework for the sets. For this thesis, this was done by using MANAGE_FE_MODELS, CREATE command to name the basic FEA models, and the working sets that were created for each. Then the CREATE_WIRE, FROM_OBJECT, GET_OBJECT command was used to have I-DEAS create a working set from the solid models that were created of the entire supports (see Chapter V).

Once initial working sets are created, the CREATE_WIRE, POINT, POSITION, BETWEEN VISIBLE command was used to create points, to create end points for the mesh areas needed for automatic meshing. In this process all unnecessary geometry from the solid modes (e.g., for this thesis, the tines used to crimp the supports to the heating coils) can be eliminated.

After end points have been created, lines can be created connecting them. These lines were eventually selected when the boundaries of the mesh area were created. They were constructed with the CREATE_WIRE, LINE, POINT_TO_POINT command. After the lines are created for the boundaries of the mesh areas (or volumes), any unneeded geometry can be deleted from the working sets.

PHYSICAL AND MATERIAL PROPERTIES TABLES

Before the mesh areas and finite elements can be created, MATerial and Physical properties tables needed to be constructed. This involved the specification of the desired properties for materials being modeled.

I-DEAS provides a command called "ANALYSIS_PROGRAM_MASK". This command can be used to specify the analysis program which will be used to solve the FEA models. Once this is done, I-DEAS will only make available those material and physical properties used by the specified analysis program. It allows the user to select from: NEUTRAL; MODEL_-SOLN/OPTIM (I-DEAS); FRAME (I-DEAS); NASTRAN_(MSC); ANSYS_SASI; and SUPERB. ANSYS_SASI was selected for this thesis; and then the material properties tables were created.

After the materials tables were created, physical properties tables needed to be created to describe physical properties (e.g., thickness) the elements being used in the FEA models. For the shell elements that were applied in this thesis, the only physical property that the elements required was their thickness.

CREATION OF MESH AREAS

Once the material properties were defined, mesh areas can be constructed. I-DEAS Engineering Analysis provides commands required for the creation, orientation, modification, and management of mesh areas and mesh volumes. It also contain the commands for the automatic meshing of the mesh areas and volumes, and for the direct creation, orientation, modification, and management of the active FEA models nodes and elements.

In order for the mesh areas or volumes to be visible on the screen, the MESH_SWS, MESH_AREA_SW and/or MESH_VOLUMES_SW switch(s) must be turned ON. Also, when it was desirable, the mesh area (and volumes) labels can be displayed with the LABELS, MESH_SWS, ON command.

Before creating mesh areas, the following options can be selected or confirmed as the desired default values. The SURFACING command is used to confirm that the default value of AUTOMATICALLY_ATTACHED surfaces is set. This caused I-DEAS to automatically create and attach a surface to each mesh area as they were created. The COLOR command is used to change colors for each set of mesh areas that are created. Finally, the TYPE_OF_MESH_AREA command is used to specify which mesh areas type is desired.

Once the defaults are set, mesh areas can be created through the manual graphics cursor picking of the curves needed to create each mesh area, with the MESH_AREA, CREATE command. The create command requires the input of the mesh area start label (applying the default of 1 for the first mesh area), and the label increment (which was left as the default of 1). Next it calls for the confirmation of the element type which is to be used for all subsequent mesh areas created. After the element type is specified, the CREATE command asks for the selection of the Physical Property Table. This is followed by the confirmation of the material type (default is ISOTROPIC). Finally, the Material Property Tables required. After all of these specifications are made, the CREATE command asks for the selection of sets of three to one hundred curves that formed a continuous loop around each mesh area. The curves were generally selected by using the screen cursor to pick the VISIBLE curves. I-DEAS highlights each of these curves as they were picked. But, in the case of relatively small lines and the arcs at the supports' bends, the SCREEN_AREA, DIAGONAL_POINTS screen selection method can be used. When this was done, I-DEAS did not highlight those curve. If a curve was selected which does not share a common end point with the previously selected curve, I-DEAS notes the error and asks for a new selection. If a pick as ambiguous, due to curves which are laying on top of each other, I-DEAS presents a directory of their labels, and one can then be selected. Once the last curve in the loop around a mesh area is picked, D (Done) is keyed in. This may causes the program to state that no common surface was found related to all of the selected curves (because no areas existed in the models), and it asks if one is to be picked. Once NO is selected, a surface and the mesh area are created. The CREATE command then proceeds with the prompts needed to create additional mesh areas, until Done (D) is typed, or a <RETURN> is entered without a cursor pick.

MULTIPLE FEA MODELS

Once mesh areas have been created I-DEAS automatic mesh generator can be used to create the nodes and elements on them. In order to create several models (i.e., several mesh sizes for each) from the same mesh areas, temporary models can be copied into other models using the commands located in the MANAGE menu.

DISTINGUISHING MODELS' PIECES WITH COLORS

In order to be able to distinguish the nodes and elements associated with each piece of an FEA model, different colors can be used for each piece. This is accomplished by setting the default colors for the nodes and elements, and then meshing one piece at a time. This scheme doesn't always work exactly as it should have. It is not known why. But, is some cases the default colors are only applied at the boundaries of the mesh areas. When this occurs, the colors can be modified after the nodes and elements are created. The entities that required modification can be specified by the mesh areas they were related to. If the mesh areas for each piece are created with different colors, the mesh areas' colors can be used to specify which mesh areas were desired.

SPECIFYING MESH SIZES

In order to use the automatic mesh generator, the MESH_SIZE needs to be specified. This determines the nominal size of the elements that the automatic mesh generator will produce. I-DEAS provides commands which can be used to specify element sizes for localized areas, to scale the global and local element sizes, and to determine the number of elements generated by dividing up the lengths of the selected curves.

Whenever I-DEAS meshes mesh areas or mesh volumes, it freezes them. This allows I-DEAS to remember the locations of the nodes, and it prevents them from being deleted if nodes and elements exist in these areas. The MESH_SIZE, FREE_MESHING, UNFREEZE command is used to unfreeze mesh areas when nodes and elements had been created that needed to be deleted (e.g., due to application of the wrong global mesh size).

AUTOMATIC MESH GENERATION

Once all of the preparations for the automatic generation of the nodes and elements, I-DEAS GENERATE_MESH command can be used to mesh the entering the color of the mesh area; and then entering a <RETURN>, mesh the specified mesh areas. If the nodes and/or elements display are turned on, they will be displayed on the screen as they are generated. This is desirable, because after I-DEAS generates the mesh, it prompts the user to answer if it is "OK to keep nodes and elements?" (Yes or No). Once this question is answered, I-DEAS terminates automatic mesh generation.

MERGING NODES

When desired, nodes in FEA modes can be merged together, as was done for this thesis's "welded" nodes. In I-DEAS Preprocessing, nodes can be merged with the NODE_COINCIDENCE command. This command is used to check an FEA model for nodes which reside in coincident locations, to merge them if desired. The command allows the user to specify which nodes are to be tested for coincidence, and the distance between nodes that is to be considered coincident. This command can be used to select nodes to be tested for coincidence using the SCREEN_AREA, DIAGONAL_-POINTS method to limit the testing to only those sets of nodes that are to be merged. Once the appropriate nodes were specified, entering D (Done) or <RETURN> was entered without picking any additional nodes, tells I-DEAS that no more nodes are to be tested. Then I-DEAS asks the user for the distance between nodes that is to be considered coincident. After this is specified, I-DEAS determines which, if any, nodes are coincident, and prints out either the coincident node labels only, or the coincident nodes labels and the element which use those nodes. Finally, the NODE_COINCIDENCE command asked if it is "OK to merge coincident nodes?". When the correct listing of nodes for "welding" is obtained, this final question can be answered with yes. This caused

I-DEAS to redefine the elements which used the pairs of coincident nodes, eliminating the use of one of the nodes. This process doesn't deleted the unused nodes. That can be accomplished with the NODES, DELETE, ALL command. The delete command can not delete any node which is used in an element's connectivity. This allows the DELETE, ALL command to be used, because only those nodes which are unused after the merging operation are deleted.

RESTRAINING FEA MODELS

Once FEA models have been constructed, it is necessary to restrain them so that they could be loaded. The restraints needed to model the manner in which the objects being modeled will be supported in application.

At this stage in an analysis, it may be necessary to convert the FEA model's nodes' definition and displacement coordinate systems from one coordinate system to another (e.g., from Cartesian to Cylindrical coordinates). This will allow the model's restraints to be created using different global coordinate directions then were used in the creating of its nodes.

I-DEAS displays the labels x, y, and z for all coordinate values, regardless of which global coordinate system is active. Thus, when the nodal displacements were restrained, and when the restrains were listed for verification, the "x" values represented the radial dimension; the "y" values represented the angular dimension; and the "z" values represented the vertical dimension, in the active Global Cylindrical Coordinate System.

The RESTRAINTS command presents a menu of commands which are used to create restraint cases and to create the restraints applied in those cases. Multiple restraint cases can be created for a single FEA model. The set which is being viewed and operated on is made active with the RESTRAINTS, CURRENT command. Thus, the first step in creating the restraints is to use the CREATE_SET command to establish the name (or number) under which the set was stored.

Having establish the active restraint set, the RESTRAINTS, NODAL_-DISPLACEMENT command can be used to specify the nodes for restraint. I-DEAS allows restraints to be specified as constants, setting the displacement of each node selected's translations and rotations to specified values (most often zero), or to set them "FREE".

LOADING OF FEA MODELS

After the models' are restrained, the next step in the I-DEAS preprocessing of an FEA model is the application of static loads. This involves the creation of LOAD_SETs, which contain the desired loads. For this thesis, FACE_PRESSUREs, and NODAL_FORCES were used.

ANALYSIS CASE SETS

The next step in I-DEAS preprocessing of FEA models is the creation of analysis case sets. A case set contains at least one load set, and one restraint, and/or one constraint set. Case sets are created with the CASE_MANAGEMENT, CREATE command. The CREATE command prompts the user for a case set name or number under which it will be stored. It then asks for the selection of the load set(s), and a restraint set, and a constraint set. Responding to any of the load, restraint or constraint selection prompts with a <RETURN> and no name or number tells I-DEAS that all of the load sets have been specified, or that no restraint or no constraint set is to be used respectively. I-DEAS also provides the standard MODIFY, DELETE, LIST, RENAME, DIRECTORY and STATUS commands which can be used to manage the cases sets once they have been created.

WAVE FRONT OPTIMIZATION

The computer requirements for an ANSYS solution run can be reduced by optimizing the "wave front size". The ANSYS program solves the system of simultaneous linear equations developed from an assembled finite element model using an "in-core wave front" procedure. The wave front is the number of equations which are active in the computer's memory after an element has been processed during the solution procedure. This method places a restriction on the size of the wave front for any given model definition. The restriction is based on the amount of computer storage available in the system being used.

The size of the wave front is determined by the sequence in which the elements are arranged. The size of the wave front will expands and contracts as the simultaneous equations are solved. The total system of equations is assembled from the element matrices into an assembly matrix. The first time the equations for a node are required by an element, they are added to the assembly matrix, expanding the wave front. Each time a node is used by the elements for the last time, its equations are algebraically solved, and then its equations are eliminated from the assembly matrix in the computer's memory, contracting the wave front. As the solution of an FEA model progresses, the total number of nodes equations in the assembly matrix determines the "instantaneous wave front size". The amount of active computer memory is therefore dependent on the instantaneous wave front size. The computer solution time is also dependent on the wave front size. It is proportional to the square of the mean wave front size.

Therefore, to minimize the computer memory and computation time, it is desirable to minimize the wave front size. I-DEAS provides commands which can be used to optimize (i.e., minimize) the wave front [size]. It also provides for the management of the "band width", which is important when other FEA programs that use "band solver" routines are used. This involves optimization based on node numbering, rather than element numbering. But, since ANSYS was used for this thesis. Band widths will not be discussed any further.

For wave front size optimization, the user first executes the OPTIMIZE_WAVEFRONT command. This causes I-DEAS to automatically optimize the wave front size. It presents the total number of elements processed, total length of the node adjacency array, and the maximum node degree. Then it displays a table of the current ("BEFORE") and optimized ("AFTER") element "frontwidth" statics. This includes the calculated maximum, average, and root mean square (RMS) values of the wave front. If the optimization results in an improved (i.e., smaller frontwidth), then the RELABEL_ELEMENTS command should be executed. To verify that this has been executed properly, the CHECK_WAVEFRONT_SAT command can be executed. This should show that the current frontwidth values match those displayed in the "AFTER" column in the OPTIMIZE_WAVEFRONT command first executed.

TRANSFER OF THE MODELS TO ANSYS

Once an FEA models is constructed, a case set is created, and its frontwidth is optimized, it can be transferred to ANSYS for final preprocessing, and submittal to the ANSYS solution program. This is accomplished with the WRITE, ANSYS command. This command tells I-DEAS to write an (ASCII) ANSYS input file. It will be named MODEL_FILE_-NAME.DAT (where MODEL_FILE_NAME is the name of the current I-DEAS model file) by default. Or, the user can enter any appropriate name in response to this command's prompt for a file name.

After the WRITE, ANSYS command asks for the file name, it asks if the problem type is STRUCTURAL, NORMAL_MODES, or HEAT_TRANSFER. The user is them prompted for the selection of the case set(s) that are to be written to the ANSYS input file. Finally, the WRITE, ANSYS commands asks if starting and/or ending "JCL" are needed. The JCL stands for Job Control Language cards, which may be require for input of ANSYS input files into some installations of the ANSYS software. At GEA neither are needed. Therefore, both JCL prompts were answered as "NO" (the default answer), when each of the files were written out for this thesis.

Having written out ANSYS input files to the VAX computer disk directory, the models are ready for the next step, which is file editing and ANSYS preprocessing. This is discussed in Appendix XIII.

APPENDIX XIII

ANSYS PREPROCESSING OPERATIONS

ANSYS SOFTWARE

At GEA, Swanson Analysis Systems Inc.'s ANSYS finite element programs are available on select VAX computers in the AP1 and AP35 VAX clusters. At the time this thesis was being completed, ANSYS was available only on the "AP35V3" VAX. This was determined by typing:

```
WHEREIS ANSYS <RETURN>,
```

at the VAX system level prompt.

Once a user is logged on to AP35V3, the ANSYS programs can be called up by entering:

```
ANSYS <RETURN>,
```

at the VAX system level prompt. If this is done, the program prompts the user for the selection of the version of ANSYS which is desired. At the time of this writing, at GEA the following ANSYS versions were available: 4.1C, 4.2B, 4.3, 4.3A, 4.4, and 4.4A. Version 4.3A was used for this thesis, because it is the version that the author had been trained to use at an "Introduction to ANSYS" seminar course taught by an ANSYS Support Distributor.

ANSYS INTERACTIVE COMMAND FILE

The ANSYS software creates and makes use of multiple files in the user's VAX disk directory, some of which are described in one of the next sections of this chapter. ANSYS automatically creates the names for these files, unless the user assigns logical names to these files prior to calling up the ANSYS program. ANSYS uses names like FILE12.DAT. If the ANSYS program is entered from a VAX disk directory that already contains these files from a previous ANSYS session, they may be overwritten, and destroyed. Therefore, if the user does not preassign other names for the ANSYS files, or at least rename those files that are to be saved before entering the next ANSYS session, it is recommended that the user perform different FE analyses from different disk directories to prevent losses. But, a better approach is to write, or make use of a copy of an existing, command file which will assist the user in assigning logical names to the ANSYS files.

For this thesis, an interactive command file, written in 1988 by Dr. Mike Day at the University of Louisville, called "ANSYS_INTER.COM" was utilized. This file contains command lines that prompt the user for a logical file name, and then it assigns this name to all of the require ANSYS files. With this command file, it was a simple matter of using the same names as the ANSYS PREP7 input files that were discussed in Chapter X. For example, when the PREP7 input file was N116.DAT, this command file allowed the name "N116" to be used for the ANSYS files. It then made the file name assignments like N116.F12, where the file with the .F12 extension matches the standard ANSYS FILE12.DAT file.

In addition to making the logical file name assignments, the ANSYS_INTER.COM command file also calls up ANSYS version 4.3A. It reminds the user that /INT must be entered within ANSYS to run it in the interactive mode. Also, when the user exits ANSYS, this command file assists the user in maintaining the ANSYS files. Without this command file, ANSYS would leave any file generated during the session in the users disk directory after exiting the program. But, with the command file, the user is asked if he would like to delete all ANSYS files except the preprocessing and postprocessing files. The user can enter Y, for Yes, and all other files will be deleted by the command file. Or, the user can answer N, for No, and no files will be deleted. Finally, the user can enter P, for Prompt, and the user will be prompted for the deletion of each of the existing ANSYS files. A copy of Dr. Day's command file is listed in Appendix IV.

ANSYS' PROGRAM LEVELS

When ANSYS is entered (with, or without the command file), the first input is usually:

/INT <RETURN>.

This tells ANSYS the user wants to operate in an interactive mode. Since the interactive command file was used to activate ANSYS for this thesis, it reminded the author that /INT was needed to work interactively.

Once the interactive mode is specified, the program enters what ANSYS refers to as the "Routine Begin Level". Then the program displays an introductory banner message, followed by a "BEGIN -INP=" prompt. This is followed by a cursor, presented for the input of a program command line. At this level in ANSYS, the ANSYS files can be managed, on-line documentation can be reviewed, and the user can select a "Routine Level" of the ANSYS program.

ANSYS uses different terminology for the structure of its commands then I-DEAS. Rather than Families, Modules, and Tasks; ANSYS refers to Phases, Routines, and Modules. The Phases do not actually exist as a level of commands within ANSYS. They are simply the logical steps followed in performing a finite element analysis with the ANSYS software, which are: preprocessing, solution, and postprocessing. The Routines are the first step down from the Routine Begin Level. Each Routine is a collection of commands which are related by their general function in performing the operations required for a Phases of an analysis. Modules are collections of commands within an ANSYS Routine, that are grouped together for the purpose of retrieving on-line documentation on them.

ANSYS COMMANDS

Each of the ANSYS command can usually be input with the first four letters of their complete names (more if they are necessary to distinguish it from the other commands in the current Routine). In this text the first reference to a command will be shown in a form like:

COMMand.

Then any reference to that command will appear as:

COMM.

Unlike I-DEAS, ANSYS is not a menu driven system. Each operation is performed by inputting a command line. Within a Routine any of the commands present in the Routine can be executed, without the need to move into different levels of the Routine's structure. Therefore, unlike I-DEAS which requires movement in its Task and menu structure to locate the desired commands, all ANSYS commands are present anywhere within a Routine. But, because there are no menus presented of the available commands, the user must either rely on memorization of the commands, crib notes, hand books, or on-line documentation to know which commands are available.

ON-LINE DOCUMENTATION

ANSYS provides extensive on-line documentation, which can be used at any time during an interactive ANSYS session, reducing the need for external documentation. The on-line documentation for each Rountine's commands are organized in Modules. To view documentation on a specific command, the user must enter a command line specifying the Module desired, followed by a command requesting documentation. By typing:

HELP <RETURN>,

ANSYS will respond with a general explanation of how the on-line documentation can be obtained. But, ANSYS does not use the word HELP to obtain any further information, it uses the word "DOCUmentation".

At any time during an interactive ANSYS session, DOCU can be requested. Without any qualifiers,

/DOCU <RETURN>,

will yield a description of the commands present in the current Module. If the qualifier "ALL" is used, that is, if

/DOCU, ALL <RETURN>

is entered, all of the commands present in the current Module will be listed including the format for their use with appropriate parameters.

If a command is used as a qualifier, that is, if

/DOCU, COMMAND-NAME <RETURN>,

is entered, where COMMAND-NAME is a command in the active Routine, ANSYS will display that command's purpose, arguments, and their definitions.

The "current Module" is determined by the last command that was issued. Once a command is issued, ANSYS moves into the Module containing that command, and it becomes the current Module. Therefore, if a command is issue, and then DOCU is used, all of the commands in the same Module as that command will be listed.

But, Modules can also be entered by typing the name of the Module. ANSYS makes it easy to determine what Modules are available in the current Routine. By simply typing:

MODUle <RETURN>,

ANSYS will respond with a list of all of the current Routine's Modules, and a brief description of what each Module's commands are used for. Likewise, it is equally easy to determine which Module is current. This is done by typing:

STATus <RETURN>.

STAT will yield the current Routine, and Module; and in addition, it will list the status on the parameters that are pertinent to the current Module.

SLASH COMMANDS

ANSYS uses several commands which are proceeded with a slash (i.e., a "/"). These commands provide general control functions, and many can be used at both the Routine Begin Level and a Routine Levels. Typing:

DOCU,/CMD <RETURN>,

will yield documentation on the commands which are available for use at any current level.

The /COMment command can be used to input and list comments which can be written into the ASCII coded ANSYS files. Any line starting with /COM will be treated as a comment, and the entire line can be used. When an ASCII coded ANSYS file is being created, or edited with a file editor, additional comments can be placed in any line, starting with a "*" placed in any column other than the first column in a line.

The /EOF command is used to tell ANSYS that the End Of the current File has been reached. When this command is used at the Routine Level, work in that Routine is terminated, and control is returned to the Routine Begin Level. When /EOF is used at the Routine Begin Level, operation in the ANSYS program is terminated, and the program is exited.

The /INPUT and /OUTPUT commands are used to switch the input or out of the next commands to a file in the current disk directory. The format for each is:

```
/INPUT (or /OUTPUT),FNAME,EXT <RETURN>,
```

where FNAME is the name of the disk file (eight characters maximum), and the EXT is the file name's extension (three character maximum). For example:

```
/INPUT,N116,DAT <RETURN>,
```

would be used to transfer the ANSYS input file named "N116.DAT" into ANSYS from the current disk directory.

The title of the FEA model (or analysis case if more than one analyses is being done on similar models) is input by typing:

```
/TITLE,NEW SU SUPT W/ 1/16" MESH - J.A.KERN <RETURN>,
```

as an example. When I-DEAS created the ANSYS input files it wrote a title in each file which consisted of the I-DEAS database description. This resulted in the same TITLE in all of the ANSYS input files. So, when the files were edited, the TITLEs were changed to better describe each analysis case.

ANSYS FILES

As discussed earlier, ANSYS creates and works with multiple files which are stored in the users current disk directory. Some of these files are written in binary code and can not be edited with a screen editor or word processor, while others are written in ASCII code and can be edited. Each file serves a different purpose, and some store different information before and after the solution routine has been run. All of the ANSYS files are described in the ANSYS Engineering Analysis System User's Manual [19]. Plus, ANSYS includes a screen as part of the on-line DOCUmentation which can be reviewed during an interactive ANSYS session by typing:

DOCU,FILE <RETURN>,

or simply

FILES <RETURN>.

Table 7. lists some of the ANSYS files that are most often used. These file names are the default name declarations made by ANSYS. For this thesis, the interactive command file was run when accessing ANSYS, giving all files logical names like "N116.F02" in place of "FILE02.DAT".

[19] Gabriel J. DeSalvo, Ph.D., Robert W. Gorman, ANSYS Engineering Analysis System User's Manual, Swanson Analysis Systems, Inc., Houston, Pennsylvania, 1987, Appendix C.

TABLE 7.

ANSYS FILES

FILE	TYPE	CONTENTS
FILE2	BINARY	MODEL GEOMETRY AND PREVIOUS ITERATION SOLUTION
FILE3	BINARY	ANALYSIS DATA INPUT FROM SFWRITE COMMAND
FILE5	ASCII	INITIAL INPUT DATA
FILE12	BINARY	POSTPROCESSING DATA AFTER SOLUTION ROUTINE IS RUN
FILE14	ASCII	ELEMENT DEFINITIONS ONLY (WRITTEN ONLY UPON REQUEST)
FILE15	ASCII	NODE DEFINITIONS ONLY (WRITTEN ONLY UPON REQUEST)
FILE16	BINARY	FILE FOR SAVING PREP7 WORK FOR LATER RESUME (SAVE/FINI)
FILE17	BINARY	TEMPERATURES WRITTEN BY SFWRIT FOR KTEMP=-1 ONLY
FILE18	ASCII	LIST OF ALL USER INPUT COMMANDS
FILE19	ASCII	ERROR LOG
FILE21	BINARY	PLOT DATA BASE
FILE23	ASCII	BOUNDARY CONDITIONS ONLY (WRITTEN ONLY UPON REQUEST)
FILE27	ASCII	COMPLETE ANSYS ANALYSIS INPUT DATA FROM AFWRITE COMMAND
FILE28	ASCII	REAL CONSTANTS ONLY (WRITTEN ONLY UPON REQUEST)
FILE29	ASCII	MATERIAL PROPERTIES ONLY (WRITTEN ONLY UPON REQUEST)

The first Phase of ANSYS finite element analysis is preprocessing. At ANSYS' Routine Begin Level, the command:

```
DOCU,PREP <RETURN>,
```

was entered in order to determine which Routine would be best suited for the PREProcessing of the FEA models for this thesis. ANSYS responded with an explanation that two preprocessor Routines are available. The PREP7 Routine is used to begin new analyses. Its modules contain commands useful for defining FEA models, and their case sets. The PREP6 Routine is use to define additional load steps for existing FEA models.

Therefore, the PREP7 Routine was used for the preprocessing Phase of these analyses. That Routine Level is reach from the Routine Begin Level by typing:

/PREP7 <RETURN>.

When PREP7 is first entered, ANSYS displays several of the most commonly needed commands, with a brief description of each.

SAVE AND RESUME COMMANDS

ANSYS PREP7 contains the capability to save all work done in preprocessing an FEA model file at any point during an interactive ANSYS session, and to resume work at the same point that the work was last saved. This is needed if the user wishes to leave an ANSYS preprocessing session and return to it later. It allows the user to protect his work by creating a back-up file which can be retrieved in the event that the system "crashes", or the data line is interrupted. Also, this allows the user to save the work that has been completed, and then execute a command which might yield unacceptable result. If the result are OK, then the user can go on with his work. But, if they are not OK, the data that is in memory can be replace with the data which was save to a disc file. This is accomplished by entering:

SAVE <RETURN>,

at any time during a PREP7 interactive session. Each time the SAVE command is executed, the data in memory is overwritten in FILE16. In ANSYS terminology, FILE16 is "rewound" and then the new data is written into the same FILE16 replacing the existing resume data. To retrieve this data, replacing the data in active memory, the user types:

RESUme <RETURN>.

The SAVE command is automatically executed each time the user exits PREP7 by typing:

FINIshed <RETURN>.

FINI is the normal command used to exit from any ANSYS Routine. The /EOF command is only used to exit a Routine when the user does not want the data in active memory to be saved.

PROCEDURE

ANSYS includes a very helpful form of documentation, which is viewed by typing:

PROCedures <RETURN>.

The PROC command presents sixteen (16) steps which can be followed in setting up a finite element analysis with ANSYS. The steps are numbered 1 through 16. The PROC command is used by typing:

PROC, N <RETURN>,

Where N is either: the NUMBER for one of the sixteen steps; PREVious, for the step prior to the last step that was reviewed; SAME, to review the same step as the last step that was reviewed; or, NEXT, for the next step in the process.

STEP 1. - SPECIFICATION OF THE ANALYSIS AND ELEMENT TYPES

PROC,1 suggests that the next step in performing an ANSYS analysis, after creating a TITLE, is to specify the ANALYsis type. For this thesis, all of the analyses performed were static structural analyses. Therefore, the ANSYS input files written by I-DEAS needed to contain a command to specify this ANALYsis Type. Entering DOCU,ANALY in ANSYS PREP7 yields a display of the KAN command, which is used to specify the analysis type, and the parameter values for each type of analysis available. For this thesis, the input files needed to contain the command line:

```
KAN,0,
```

which specifies a STATIC analysis.

PROC,1 also indicates that DOCU,SELEct should be used to select the Element Type that is to be used. Then switch to the PREP7 ETYPE Module, and use the ET command to associate the elements with a type number. Therefore, DOCU, SELE was entered to review the element types. (Note that in I-DEAS Preprocessing, the elements for the supports were specified as quadrilateral thin shell elements, and the elements for the load plates were specified as triangular thin shell elements.)

GEOM can be one of the following geometric element types: POINT, LINE, AREA, or VOLUme. DIM can be either 2D or 3D for the space that FEA model must occupy. For this thesis, the following was typed:

```
/SELE,AREA,3D <RETURN>.
```

ANSYS responded to this with a list of several element types from the ANSYS "JSTIF" library, all of which were area elements which occupy 3D space. From this list, the element type with JSTIF number 63 was chosen (quadrilateral flat shell elements).

Then following the directions from DOCU,SELE, the ANSYS command STIF, JSTIF was used. STIF provides documentation on each of the available element types, by element type number. This is a brief summary of the information that is printed in the ANSYS User's Manual [1]. But, STIF did provided the: number of nodes; degrees of freedom per node; real constants; properties; pressures; temperatures; and "KEYOPTions" which can be applied to the specified element type. Each of these were noted, and this command was revisited when each of these parameters were reviewed to determine if I-DEAS had properly written them into the ANSYS input files.

The section of the ANSYS User's Manual [2] covering STIF 63 was read, and it was discovered that the triangular elements used for the load plates could also be defined using STIF 63 elements. The difference being that the supports' quadrilateral elements would be defined by four nodes, referred to as nodes I, J, K, and L; where the load plates' elements would be defined by three nodes, with the third node repeated (i.e., nodes I, J, K, and K). Therefore, when the ANSYS input files written by I-DEAS were reviewed and edited, they were checked to see that all elements were defined using the STIF 63 element type, and all of the load plates' elements were defined by three nodes with the third node repeated as a fourth node.

Next, still following the directions from PROC,1, the command ET was reviewed. ET is used to define an element type. For this thesis,

[1] Ibid. pp. 4.63.1 - 4.63.5.

[2] Ibid.

each model contained one element type for the supports, and a second element type for the load plates when used. The format for this command is:

ET,ITYPE,JSTIF,KOP1,KOP2,...,KOP9,INOPR.

The ITYPE argument identifies the element type. It can be an integer between 1 and 20. For this thesis it was 1 for the supports, and 2 for the load plate. JSTIF is the stiffness library number for the element type, which was 63 for both ITYPE 1, and ITYPE 2. The nine possible KEYOPTions (i.e., KOP1, KOP2,...) were reviewed in the STIF,63 on-line documentation, and it was determined that none of the six options for STIF 63 were needed for the analyses to be performed, and thus all were reviewed to be sure they were set to their default value of zero. INOPR was also confirmed as set to its default value of zero. If it had been set to 1, the print out of all element solutions of this ITYPE would have been suppressed.

STEP 2. - ANALYSIS OPTIONS

PROC,2 identifies the second step in an ANSYS analysis as the time for specifying any analysis options that are desired. Following its instructions,

OPTIons <RETURN>

was typed, making the OPTIONS Module active. Then STAT, DOCU, and DOCU,ALL were all used to determine if any analysis options were desired. This lead to the use of DOCU, KAY. The KAY command is used to set special analysis options. These include: interpolated/extrapolated displacement calculations; large deflection solutions; stress stiffening; and the use of Newton-Raphson Procedures. None of these options were desired. Thus, all were left with their default values of zero, and no KAY command lines were expected in the ANSYS input files written by I-DEAS. Also, because the material properties for the supports and the load plates were linear, the KNL (Key Nonlinear) command was not used. If nonlinear material data tables (NL command) had been used, KNL,1 would have been used.

STEP 3. - MATERIAL PROPERTIES TABLES

PROC,3 leads the ANSYS PREP7 user to the application of the MATErial Properties Module, and the nonlinear properties set with KNL, and the NL command if needed. For this thesis, the MATE Module was entered, and then DOCU and DOCU,ALL were used to apply the command line:

LABEL*,MAT,C0,C1,C2,C3,C4.

In this command, LABEL* can be any valid material property label. MAT can be any arbitrary integer between 1 and 100, serving as the material identification number. For this thesis, a MAT of 1 was used for the surface unit supports, and a MAT of 2 was used for the load plates.

From STIF,63, it was known that the valid material properties for the quadrilateral flat shell elements were: EX, EY, ALPX, ALPY, NUXY, DENS, and GXY. Only those properties which were required for the static loading case analyses needed to be specified. Therefore, when the ANSYS input files written by I-DEAS were reviewed, they were check for the proper Moduli of Elasticity (i.e., EX,1,0.28E+06, and EX,2,10.0E+06); and Poison's Ratios (i.e., NUXY,1,2.9E-02, and NUXY,2,3.34E-02). Because EY values were not defined, they default to the EX values. Because they were input in I-DEAS, the coefficients of thermal expansion (ALPX), and mass densities (DENS), values for the stainless steel alloy support material and aluminum load plates were expected to appear in the ANSYS input files written by I-DEAS. But, because no thermal or acceleration loads were analyzed, they would not be used in the ANSYS solutions.

STEP 4. - REAL CONSTANTS DEFINITIONS

PROC,4 calls for the definition of the elements' Real Constants, which are set with the commands in the RCONstants Module. These are specified with the command:

R,NEST,R1,R1,R3,R4,R5,R6 <RETURN>,

in which the NEST value is real constant set's (arbitrary) identification number. The six real constants' values R1 through R6 are values specified to satisfy the requirements of the JSTIF element type(s) being used. If more than six values are needed, the RMORE command is used, and if the user needs to modify real constant values during an interactive ANSYS session, the RMODIFy command is used. Both following the same argument format as the R command.

From STIF,63, it was known that the valid material properties for the quadrilateral flat shell elements were: TK(I), TK(J), TK(K), TK(L), EFS, and THETA. The thickness values (i.e., TK(I), TK(J), etc.) can be used to vary the elements thicknesses smoothly over the area of the elements, with the thickness specified at the elements nodes. But, because the supports' and load plates' thickness were constant, only TK(I) values were needed. ANSYS applies this same value as a default to the other node thicknesses. Therefore, when the ANSYS input files written by I-DEAS were reviewed, they were checked for the proper thickness values (i.e., R,1,0.05, and R,2,0.10).

STEP 5. - GEOMETRY DEFINITION

PROC,5 presents the Modules used to define the model's "geometry" and "mesh". This can be accomplished through direct definition of nodes and elements (i.e., creating the FEA "mesh"), and due to the assembly of the elements in space, the "geometry" is also defined. But, ANSYS also provides the capability of defining the "geometry" for the models, which will serve as the boundaries for the FEA mesh, and then ANSYS provides automatic mesh generation, which allows the program to generate the nodes and elements.

ANSYS calls its automatic mesh generation "solid modeling". ANSYS solid models are not constructed like I-DEAS solid models, and they do not contain complete physical properties like I-DEAS solids. ANSYS solid models are constructed through the creation of "KEY POINTS", and then lines, arcs, areas, and volumes based on the key points. ANSYS literature shows examples of solid models which have been displayed using graphics workstations which contain terminal-dependent light source shaded image capabilities. The author has never explored these capabilities. But, it appears that if I-DEAS solid modeling is available, its capabilities in this area would be more easily applied to the creation and display of solids, then ANSYS' method of solid modeling. The exception may be in the fact that ANSYS provides the further capability of defining the solids based on parametric geometric coordinate locations. This gives the user the ability to design models with the ability to optimize the design by changing the values of parameters, rather than constructing complete models when iterations are needed.

For this thesis, the models had already been constructed in I-DEAS. At this step in the analyses, the command lines in the ANSYS input files related to the definition of the nodes and elements need to be understood, so that they could be reviewed for accuracy. The method used in the input files for "mesh definition" is the direct creation technique. Therefore, those modules, and commands referred to in PROC,5 relating to direct definition of nodes and elements were needed.

PROC,5 leads the user into the CDSYS Module. This Module is used to define the coordinate system in which the nodes (and key points if used) are defined. For this thesis, Cylindrical coordinates were desired. Therefore, the command:

```
CDSYS,$DOCU <RETURN>
```

was entered interactively. This caused ANSYS to display the on-line documentation that is available for this Module. Note that two commands were combined on one command line by separating them with a "$". This technique can be used to combine several commands on one command line. From the resulting display,

```
DOCU,CSYS <RETURN>
```

was entered to review the coordinate systems which are predefined by ANSYS for user selection. This informs the user that the CSYS command can be applied with one of the following arguments: 0 for Cartesian, 1 for Cylindrical, 2 for Spherical, or N for a Number the user has defined for a customized coordinate system. Thus, for this thesis, it was then known that the ANSYS input flies written by I-DEAS should contain the command line:

```
CSYS,1.
```

After the coordinate system was defined, PROC,5 calls for the creation on nodes. This is accomplished with the commands documented in the NODES Module. Nodes are directly created with the N command, whose format is:

```
N,NODE,X,Y,Z,THXY,THYZ,THXZ.
```

In this command, NODE is the node number that is to be assigned to the node being created. X, Y, and Z are the nodes locations in the current coordinate system. Therefore, in cylindrical coordinates these parameters are R, Theta, and Z. The values of THXY, THYZ, and THXZ are rotation angles, in degrees, of the nodal coordinate system relative to the global Cartesian coordinate system.

Finally, when the nodes have been created, the elements can be created. This is done with the commands in ELEMent Module. Before an element is created, ANSYS must be told which of the: element types that were defined with the ET command; real constant sets that were defined with the R command; and material properties sets that were defined with the MATERial Modules commands are to be applied to the elements. This is done with the following commands:

```
TYPE,ITYPE <RETURN>,
REAL,NSET <RETURN>,
```

and

```
MAT,MAT <RETURN>;
```

in which ITYPE, NSET, and MAT are the integer numbers assigned to the element type, real constant set, and material properties set desired for all elements created after issuing these commands. When I-DEAS wrote out the ANSYS input files, these three commands were combined into single command lines. Therefore, when the input files were reviewed, they contained the command line:

```
TYPE,1 $ REAL,1 $ MAT,1
```

before the elements for the supports were defined, and the command line:

```
TYPE,1 $ REAL,2 $ MAT,2
```

before the elements for the load plates were defined. If these commands were not used, all elements would have been defined with the default value of 1 for each of the sets, making all elements in the models with a single combination of element type, real constant set, and material.

The command used to create the elements was the E command which creates elements one at a time. It has the format:

```
E,I,J,K,L,M,N,O,P <RETURN>,
```

in which I,J,K,L, etc. are the nodes needed to define each element. Because quadrilateral shell elements were used, only four nodes were need for each element. Therefore, when the input files were inspected, they contained lines like:

```
E, I, J, K, L,
```

for the supports, and:

```
E, I, J, K, K,
```

for the triangular shell elements for the load plates.

PROCEDURE STEPS 6 THROUGH 9

The next four steps outlined in the on-line documentation for the ANSYS analysis PROCedure were not used for this thesis. PROC,6 explains the module and commands that can be used to define coupled degrees of freedom (DOF). PROC,7 is followed if linear constraint equations are to be used in an ANSYS analysis. PROC,8 describes the creation of "master degrees of freedom", which are needed in analysis types (KAN) 1, 2, 5, 6, and 7. Finally, PROC,9 is applied when a linear transient analysis (KAN,5) is being performed with gap nonlinearities.

STEP 10. - LOAD STEP OPTIONS

PROC,10 identifies the need to apply the load step options module, LOPTions, to specify the control, print, and data file options to the models that are to be solved. The LOPTions Module also contains the commands needed if "body loads" are to be applied, such as the loads placed on a model due to acceleration (e.g., gravity) and rotational velocity. Neither of which were applied to the analyses performed for this thesis. Which of course implies that the assumption was made that the static forces applied to the surface unit supports were significantly greater than the effects of their own weight, to make the acceleration body loads negligible.

ANSYS allows the user to define several load steps to a single finite element model, and to have then each solved in a single computation "run" of the ANSYS solution routine. For this thesis, each model was only solved for a single load step. But, this procedural step was still required for each.

To access the on-line DOCUmentation for the LOPT Module,

LOPT <RETURN>

must first be entered to access the LOPT Module. Then the DOCU command will yield a listing of the six DOCUmentation screens that are available, each of which can be reviewed by typing:

DOCU,N <RETURN>,

in which N is an integer between 1 and 6.

LOPT$DOCU,1 describes the commands that are used to control the load options. For this thesis, the only command appearing in this DOCU screen that was used for this thesis was ITERation command, and its use will be discussed in Chapter XV.

LOPT$DOCU,2 is used for modal analyses only, and thus was not used.

LOPT$DOCU,3 displays commands that can be used to define calculation options. From this list, the KRF command was used to control the nodal force and reaction force calculations and printout. The ANSYS PREP7 input files each contained the command line:

KRF,1 <RETURN>,

which tells ANSYS that the nodal forces are to be calculated and printed for each element, and the reaction forces are to be calculated and printed for the nodes which have had their displacements constrained. The input files also contained the command line:

KSE,1 <RETURN>,

which informs ANSYS that the strain energy is to be calculated and printed for each element in the model.

LOPT$DOCU,4 discusses the use of commands that will control the printing of nodal displacements and forces, and element stresses. The ANSYS PREP7 input files written by I-DEAS did not contain any of these commands.

LOPT$DOCU,5 displays the commands which can be used to control the frequency of displacements, stresses, and forces into the solution file. None of these commands were written into the input files by I-DEAS.

LOPT$DOCU,6 indicates which commands can be applied to control the load files and data in those files. Again, I-DEAS did not apply any of these commands when the PREP7 input files were written.

STEP 11. - APPLICATION OF CONSTRAINTS AND LOADS

PROC,11 indicates that the boundary conditions should be applied to the model at this step in the procedure. The models were to be constrained as discussed in Chapter X. The loading of the models is discussed in Chapter XIII for the mesh size optimization and model verification, and in Chapter XV for the load test simulation.

The nodes at the ends of the supports were to be assigned displacement values of zero in the Theta and Z directions, for all of the analyses performed. To verify that I-DEAS wrote the proper displacement constrains in the ANSYS PREP7 input files, the LOADS Module was enter, and the available commands were reviewed by typing:

LOADS$DOCU <RETURN>.

This showed that the Displacement command was need to constrain the end nodes. The D command has the following format:

D, NODE, LAB, DISP, CDISP, NEND, NINC, LAB2, ...LAB6.

NODE is the node number for the node being displace (or constrained by a zero displacement). LAB is any valid displacement direction or degree of freedom, chosen from ALL, UX, UY, UZ, ROTX, ROTY, and ROTZ. DISP is the value of the displacement. CDISP is the imaginary part of the displacement, used in harmonic response analyses. NEND and NINC can be used to name additional nodes which are between NODE and NEND in increments of NINC. LAB2,...LAB6 are other LABels for which the same value of displacement are to be specified. Therefore, the command lines in the input files were check to be sure that each contained three nodes constrained with:

"D, NODE, UY, 0., , ," and "D, NODE, UZ, 0., , ,";
or, "D, NODE, UY, 0., , , ,UZ".

STEPS 12 THROUGH 14 IN THE ANSYS ANALYSIS PROCEDURE

Step 12 in the ANSYS analysis procedure was not applied to this thesis. It involves the definition of temperatures for analyses involving thermal loads, and internal heat generation for heat transfer analyses. Steps 13 and 14 are used to merge nodes and to reorder elements to reduce the wave front which can reduce the computer requirements for the analysis solution phase. The nodes that needed to be merged were taken care of in the I-DEAS preprocessing phase of this thesis, as was the wave front optimization.

STEP 15. - WRITING THE ANALYSIS FILE

PROC,15 explains that after the analysis has been completely defined the MIScellaneous Module's AFWRITE commands is used to write out an analysis file. The format for this command is:

AFWRITE, NOUT, DLIST, NOCHK.

NOUT is the file into which the analysis data are to be written. By default, the analysis file will be FILE27. DLIST allows the analysis data to be summarized in tables during the solution phase. If DLIST is 0 (default), no summary will be printed. But, if DLIST is set to 1, the data will be summarized during the solution printout. NOCHK if set to 1 tells ANSYS that the data is not to be check before the analysis file is written. By default, NOCHK is set to 0, and the data will be checked for completeness before the analysis file is written. If ANSYS finds any fatal errors during the check, the analysis file will not be written, and error statements will be printed. Therefore, when the analysis is completely set up (which for this thesis also requires the addition of the loads that are discussed in Chapters XII and XIV), the user will issue the AFWRITE command, and then exit PREP7 with the FINIshed command.

When the ANSYS PREP7 input files written by I-DEAS were checked for accuracy and completeness, their last four lines were AFWRITE, FINISH, /INPUT,27, and FINISH. This would have caused ANSYS to write out an analysis file, exit PREP7, and then submit the analysis file to the ANSYS solution routine. These lines (except one of the FINISHEDs) were deleted from the input files. Because, it was decided that the models should be examined with the ANSYS graphical display commands before the analysis files were written and solved, to be sure that all of the model data was transferred out of I-DEAS and into ANSYS without any "bugs".

TRANSFER OF PREP7 INPUT FILES FROM I-DEAS

After the input files were edited to make the changes needed to have then conform to the steps outlined above, XIII, they were ready to be transferred into ANSYS. The first command line in each PREP7 input file was /PREP7. Therefore, these files were setup to be input from ANSYS' Routine Begin Level. This could have been in an interactive ANSYS session by typing:

```
/INPUT,INPUT_FILE,EXT <RETURN>,
```

in which INPUT_FILE is the name of the input file that was written by I-DEAS (e.g., N116), and EXT is the files extension (e.g., DAT). If the file is an ANSYS file, the file can be identified by number. For example,

```
/INPUT,27 <RETURN>,
```

is entered when submitting the analysis file FILE27.DAT or N116.F27 to the solution routine.

But, if this is done in an interactive session, the terminal will be tied up during the entire reading of the input file, and it may slow down the work of other users on the same computer network. Therefore, the PREP7 input files were transferred into ANSYS in the batch mode.

ANSYS BATCH COMMAND FILE

To make the batch transfer of the input files into ANSYS easier, a second command file written by Dr. Mike Day, at the University of Louisville, was used to submit the input files to batch processing. It is named "ANSYS_BATCH.COM". This command file is listed in Appendix V.

This command file contains command lines that prompt the user for a MODEL_NAME that is to be submitted to ANSYS. This is to be a file named MODEL_NAME.DAT, where MODEL_NAME can be any logical file name. If the file specified does not exist in the current disk directory, the command file assumes that a MODEL_FILE.F27 file exists, and it will proceed to create a .DAT file. The user is notified that this has occurred, and can abort the command files execution if this is not what was intended. The batch command file then assigns this name to all of the required ANSYS files, in the format MODEL_FILE.FNN, where NN is the file number (from 02 to 40).

After making the file declarations, the command file prompts the users for the selection of one of GEA's VAX computer nodes, and then for the batch queue on this node that is to be used. Next it asks if the batch job is to be run "NOW" or "TONIGHT". If the user responds with NOW, the job will be submitted to the batch queue immediately. But, if the job is long, and the results will not be reviewed until the following day, TONIGHT should be chosen. This will place the job in the chosen queue such that the job will not be started until midnight.

Finally, the user is asked if all the ANSYS files that are created during the run should be deleted upon completion, except for the results file (MODEL_FILE.F12), the preprocessing (model) file (MODEL_FILE.F16) and the error listing file (MODEL_FILE.F19). If the user responds with N for No, all files created during the solution run will be retained. If the user enters Y for Yes, all files created with a extension of .FNN will be deleted except for these three files.

GRAPHICAL DISPLAY OPTIONS

ANSYS provides extensive graphical display capabilities when accessed with a color graphics terminal, workstation, or PC with Tektronix terminal emulation software. To make use of these capabilities, the first thing ANSYS requires before the user can call for any "plots" (which is how ANSYS refers to it graphics displays), is to specify the graphics DEVICE-NAME which is being used. ANSYS helps remind the user of this by including a note in the banner which is displayed upon entry into /PREP7 (and the postprocessor routines which are discussed later), that shows the user the he must enter:

```
/SHOW, DEVICE-NAME
```

to enable graphics display. The DOCU,/SHOW lists the commands format:

```
/SHOW, FNAME, EXT, VECT, NCPL.
```

The FNAME can be input as a graphics device code when used to produce plots interactively on the display device. The on-line DOCUmentation lists all of the valid device codes. For this thesis Tektronix terminals and PCs with Tektronix emulation were used. So, the device codes: 4107, 4205, and 4129 were used depending on which device was being used at the time.

FNAME can also be used as a disk storage file name, with an extension of EXT. This allows the plots to be diverted to a file for later processing with one of ANSYS' external plot programs, such as PLOT33 which is available on GEA'S VAX computer cluster system. This option was not utilized for this thesis and will not be described any further.

VECT is a switch (1 or 0) which is used to specify either raster plots, if available, or vector plots. In VECT is set to 0 (default), the plots will be displayed in raster mode with color fill. This will cause the plots to be display with each entity which has been set for coloring to be displayed with colors filled in. NCPL is used to set the number of color planes (terminal dependent) available for color filling, which can be between 4 and 8. But, when VECT is set to 1, the plots will be displayed in vector form, without color filling. The user has to experiment with these setting to get optimum use of the graphics capabilities. In some cases color filling is desirable. But, in others, the colors can obscure entities which are being displayed. Color filling also requires more data transfer time, and can slow down the users progress.

COMBINING TEXT AND GRAPHICS DISPLAYS

ANSYS is capable of displaying both interactive text and graphical plots simultaneously on the same video screen. When a Tektronix terminal is used, this involves the control of the "dialog" and the graphics display areas on the screen. When the user wish to view the dialog alone,

/T <RETURN>

can be entered, which will set the dialog area to the full number of dialog lines that are specified in the terminal's setup (e.g., *DAL,32). If a plot has been displayed, the dialog will scroll over the graphics area, or the user can erased the graphics with the terminal's "G Erase" key.

If the user wishes to view a plot without seeing the text,

/G <RETURN>

can be entered, and the dialog will be turned off. This has the same effect as toggling the dialog off with a Tektronix terminal's "Dialog" key.

If the user wishes to see both text and graphics at the same time,

/B <RETURN>

can be entered. This will turn the dialog on, and set the number of dialog lines to a reduced number from the number set in the terminal's setup, and place the text area below the graphics area on the video screen. The text will scroll in the lower area without overlapping the plots when further commands are input.

PLOTTING THE FEA MODELS

The models can be viewed in one or more windows on a graphics terminal's display using the commands in ANSYS' PLOT Module. By typing:

PLOT$DOCU <RETURN>,

and

```
PLOT$DOCU,ALL <RETURN>,
```

the user can review the on-line DOCUmentation for the plot options. In PREP7 there are two basic plots available: NPLOT for displaying the model's nodes, and EPLOT for displaying the model's elements.

The display of these plots can be customized to suit the user's needs by applying the options that are presented in the other DOCUmentation screens that are suggested by the PLOT Module's DOCUmentation. These include the display commands that can be review by typing:

```
DOCU,GRPH <RETURN>.
```

The graphical display control commands are all "slash" commands. They give the user the ability to: rotate the display (/ANGLe); automatically scale the plot to fill the display window (/AUTO); specify the distance from which the entities are viewed (/DISTance); distort the display of displacements for clarity (/DSCAle); specify the location of the focal point (/FOCUs); apply grids, log scaling, and axis labels (/GRAPH); set color specifications (/NUMBer); display boundary conditions (/PBC); control the display of the system triad, border around the window(s), and title (/PLOF); apply numbering to the displayed entities (/PNUMber); distort the display along the X and/or Y screen axes to assist in the visualization of obscure entities (/RATIo); shrink the elements to examine their boundaries (/SHRInk); write text directly on the display at a user specified location (/TLABel); create section and hidden-line displays (/TYPe); apply user specified scaling (/USER); create perspective displays using a view cone angle (/VCONe); define the viewing direction along a line normal to the display screen (/VIEW); define, activate, and deactivate display windows (/WIND); and if the graphics terminal has the capability to perform screen picking (i.e., with a joy pad, mouse, tablet, or screen cursor control wheels) the user can pick the center and a radius point on a window to zoom in on the entities displayed (/ZOOM). There are several other display control commands presented in the DOCU,GRPH screen, but these were used most often. Several of these options will only take effect when the /USER command is applied. These are noted in the on-line documentation which is available for each by typing:

```
DOCU, COMMAND <RETURN>,
```

in which the display control command's name is used for the COMMAND.

In addition to using the /ZOOM command to display selected portions of a complete node of element plot, the use of commands in the RSELect Module can also be applied to limit the number of entities displayed. These commands are documented for on-line review, which can be seen by typing:

```
RSEL$DOCU <RETURN>.
```

There are commands for selecting nodes, elements, keypoints, line segments, areas, and volumes. Also, entities can be grouped into components with the commands in the COMPonents Module.

Various combinations of these display options were used to view the FEA models that were transferred in from I-DEAS to be sure that all of the data were transferred without errors. Some of the displays are shown in Chapters XII and XIV. Once the author was satisfied that the models were constructed as desired, analysis files were written for input into the ANSYS solution Routine.

ANSYS ANALYSIS FILES

The analysis files were written out to ANSYS "FILE27" (actually named MODEL_FILE.F27 by the interactive command file that was used), for input into the solution routine, using the AFWRITE command. Thus, completing Step 15 in the ANSYS analysis procedure discussed earlier.

After the analysis file was written, the PREP7 module was exited with the FINIshed command. This returned control to the Routine Begin Level of the ANSYS program, and also wrote a PREP7 RESUME file (FILE16), that could be used to reload the models for further preprocessing.

BATCH SOLUTION OF THE ANSYS ANALYSIS FILES

The analysis files were submitted to ANSYS' solution routine in the batch mode. This was accomplished with the same batch command file that was used to transfer the models into ANSYS from I-DEAS. To do this, the model input files that had been transferred into ANSYS were renamed, so that they could be save. Then new files were created with the same MODEL_NAME.DAT file names. These files served as solution phase input files. They contained the following command line:

```
/INPUT,27
FINISHED
/EOF.
```

This allowed the batch command file to be used to submit the input files to ANSYS for batch processing, using the logical file names that were declared when the models were input with this batch command file. Because of the file declarations made by the command file, the /INPUT,27 command was used to input the analysis files with the MODEL_NAME.F27 file names (e.g., N116.DAT), rather than the standard ANSYS analysis file name FILE27.DAT.

The batch command file was also used to deleted all of the files ANSYS created during solution of the analysis files, except for the solution results (MODEL_NAME.12), the preprocessing resume file (MODEL_-NAME.16), and the errors file (MODEL_NAME.19), when the solutions were completed.

Three additional files are created during the solution phase. They are MODEL_FILE.LOG, MODEL_FILE.LIS, and MODEL_FILE.COM. The LOG file contains information on the system control commands that were issued in during the batch "run", and at the end of the file any warnings and errors will be listed. Since the MODEL_FILE.F19 error file was not deleted, any warnings and errors will be listed in this file, and so the LOG file does not necessarily need to be saved. The LISt file contains a summary of the input and analysis results. Because ANSYS and I-DEAS postprocessing can be used to more conveniently investigate the analysis results, this file is seldom use. The COM file lists the system level commands that were executed in SETting system parameters, ASSIGNing and DEASSIGNing the logical file names, RUNning ANSYS, and DELETing the undesired files. This file would only be reviewed if a processing error occurred, and the user was trying to locate any problems. Therefore, one might consider modifying the batch command file to delete the three files from the user's disk directory along with the other ANSYS files that it can delete for the user automatically at the conclusion of the solution run.

APPENDIX XIV

POST PROCESSING OPERATIONS

ANSYS POST PROCESSING OF THE SOLUTION RESULTS

From ANSYS' Routine Begin Level,

DOCU, POST <RETURN>

can be entered. ANSYS responds to this with documentation describing the post processing phase of an ANSYS analysis. ANSYS provides five post processing routines. Two of the routines are dedicated to the summation of harmonic loads, /POST29 and /POST30, for solid and shell models respectively, which do not apply to this thesis. /POST27 is used to combine the results from several load sets for comparison and display of the results. Since only one load set was run for each of the models constructed for this thesis, this Routine was not used. /POST26 is used to print out and plot displacement, stress, or temperature time histories from the results of time dependent analysis (i.e., creep analyses, and dynamic analyses). But, since only static load cases were analyzed, this Routine was not used either. The postprocessor that was used is named /POST1.

/POST1 is ANSYS' general analysis results data base postprocessor. It provided module of command that all the user to select result, sort, list, print, graphically display, and plot analysis results. Therefore,

/POST1 <RETURN>,

was entered to access this routine in interactive ANSYS sessions.

Unlike the preprocessing routine (/PREP7), /POST1 does not provide the user with a procedure to follow in its on-line DOCUmentation, since there is no step by step procedure that would be applicable to all post processing projects. But, the can enter

HELP <RETURN>,

to show the user how to access on-line documentation that is available. The documentation includes the MODUles command, to determine where the user can located on-line DOCUmentation on the commands that are provided in /POST1. HELP also shows the user that a command called PSTIF is provided for use in determining what stress data is available for retrieval from the results file. A command named SLCT is provided which discusses the processes available for selecting the nodes and elements for which use in subsequent post processing. Similarly, a SRTG command discusses how the data can be sorted into ascending of descending order prior to listing or printing the results for the nodes or elements.

The /POST1 Modules are: DEFINE, SELECT, SORT, PRINT, PLOT, CDSYS, NODE, CALCULate, LOCALCulate, and FATIGUE. For this thesis the first five MODULES in this list were applicable, and were used for the preliminary post processing of the results of the loading cases that are discussed in Chapters XII and XIV. The DOCU command was used to display the documentation available on-line which describes the command available in each of this Modules.

DEFINE MODULE PROCESSES

The DEFINE Module is used to define the stress and force data that is to be retrieved from the results file for processing, and to read the data into the interactive post processing session for processing. Before the stresses were defined for retrieval,

PSTIF,63 <RETURN>

was entered to review that data that were available for the thin shell elements that were used in these analyses. This yielded a listing of the "LEVEL 1-3 DATA FOR STIF 63 [ELEMENTS]".

Section 4.0.4 of the ANSYS Engineering Analysis System User's Manual [22] was reviewed to understand the various levels of data that are available. This documentation (not available on-line) explains that, in order to conserve computation time and disk storage space (i.e., to save analysis time and the cost of run time), all of the stress data that could be generated during the solution phase are not automatically calculated. ANSYS divides the available stress data into six "KED LEVELS". These levels are numbered 1 through 6, representing: 1) force components; 2) basic stress components (centroidal and nodal); 3) principal stresses (centroidal and nodal), temperatures, etc.; 4) additional surface data; 5) nonlinear centroidal data; and 6) linear integration point data.[23]

Not all element types are capable of producing data for all six of these levels. The description for a selected element type must be reviewed to determine which data are available for calculation during

[22] Ibid., pp. 4.0.10 - 4.0.11.

[23] Ibid., Table 4.0.4.1, p. 4.0.11.

the solution phase. ANSYS calculates all data that are in levels 1 through 3 by default. If the user needs to have items calculated that are in levels 4 through 6, these must be specified during the preprocessing phase, in order to tell ANSYS to produce those data during the solution run. The ANSYS preprocessing command which is used to specify the solution of data from KED levels 4 through 6 is "KEYOPTion". The format for this command is:

KEYOPT, ITYPE, KNUM, KEYOPT(KNUM).

In this command, ITYPE is the number of one of the defined element types being used, KNUM,is the number of the KEYOPTion that is being specified, and KEYOPT(KNUM) is the argument value desired for the key option being set. The STIFF,JSTIFF on-line documentation command lists the key-options that are available for the JSTIFF element type that is being used.

For this thesis, no additional calculations were required beyond the data that are produced by default from KED levels 1 through 3.

SPECIFYING THE STRESS DATA THAT ARE TO BE STORED FOR PROCESSING

ANSYS automatically store the some "convenience" stress results when the analysis results are read into memory for post processing. The convenience labels are in two groups: COMPonent nodal stresses: SX, SY, SXY, and SXZ; and PRINcipal nodal stresses: SIG1, SIG2, SIG3, S.I. (Max. Shear), and SIGE (Von Mises). The user must declare any other stress values that are desired from the results file. This is accomplished with the ANSYS POST1 STRESS command. Also, the sorting commands (discussed below) are only available for data that are declared with the STRESS command.

The STRESS command was used to identify the stress components that were to be processed. The format for the STRESS command is:

STRESS, LAB, JSTIF, ITEM.

In this, LAB, is a user specified label (four characters maximum), which can be used with subsequent commands to identify this stress component for processing. ANSYS provides several "convenience" labels for use with strain energy, kinetic energy, element volume, centroidal thermal and magnetic vectors. These can be obtained from the on-line DOCUmentation of STRESS. ANSYS cautions the user against using predefined ANSYS labels, like SX, DISP, UX, SIG1, etc. JSTIFF is the element library stiffness type number for the elements whose stress data are to be stored. ITEM is used to identify the solution data that are to be stored with this LABel, as identified by PSTIFF,JSTIFF or the post data listing provided in the ANSYS Engineering Analysis System User's Manual [24] for each element stiffness type. JSTIFF and ITEM are not used if any of the convenience labels are used.

STORING ANALYSIS RESULTS FOR POST PROCESSING

Once the STRESS command was used to specify that the principal stresses were to be stored for processing, the ANSYS POST1 SET command w as used to tell ANSYS to store the geometric, stress, force and displacement data for processing. The SET command is used to specify which load step and iteration's data are to be stored. The format for the SET command is:

SET, LSTEP, ITER, NSECT, KIMG <RETURN>.

[24] Ibid., pp. 4.0.10 - 4.0.11.

LSTEP, is used to tell ANSYS which load step is to be used. The default is load set 1. For this thesis, only one load step was solved for each of the static cases that was solved. ITER is used to indicate which iteration from within that load step's solution is to be used. Only one iteration was made in the analyses used to determine the optimum mesh size. For the simulated load test analyses, iterative solution steps were required. This is discussed in Chapter XV. NSECT is used with cyclic analyses to specify which sector of the solution data is to be read. This did not apply to this thesis. Similarly, KIMG, which is used to store the imaginary part of complex analysis, was not applicable for this thesis. Therefore, for each of the interactive ANSYS post processing sessions that were run of this thesis, the following command line was issued:

```
SET, 1, 1 <RETURN>.
```

SORTING AND PRINTING ANALYSIS RESULTS IN ANSYS POST PROCESSING

All analysis results that are read into memory with the SET command can be reviewed as numerical data with the commands documented in the PRINT module. These commands can be reviewed by entering:

```
PRINT$DOCU,(ALL) <RETURN>.
```

For this thesis, the maximum nodal displacements, and the maximum principal stress where desired, to determine which mesh size was best suited for these analyses, and to determine if the new support's strength is acceptably in comparison to the old support design. The nodal displacements can be printed out using the PRDISP command, and the values of stress that have been stored can be printed out with the PRSTRS command. But, before these commands were used, the data was sorted to place the maximum values of the desired displacements and stress at the top of each listing.

Sorting is accomplished with the commands that are documented in the SORT module. The data is actually sorted based on the nodes or elements in the model.

For this thesis, the nodes needed to be sorted based on their displacement in the vertical direction. This was accomplished by applying the NSORT command. Its format is:

NSORT, LABEL, ORDER, KABS, NMAX.

LABEL is the sorting parameter. It must be one of the following: X, Y, Z, DISP, ROT, UX, UZ, UZ, ROTX, ROTY, ROTZ, TEMP, PRES, VOLT, MAG, SX, SY, SXY, SXZ, SIG1, SIG2, SIG3, S.I., or SIGE. Therefore, to sort the nodes based on their vertical displacements, UZ was used as the LABEL. The ORDER can be specified as either 0 or negative for descending order, or as 1 or positive for ascending order. Thus, descending order was specified. KABS is used to specify that the nodes are to be sorted on the parameter data's absolute values if it is set to 1 or positive, or their absolute values if it is set to 0 or negative. Since the vertical displacements were in the negative Z direction, absolute values were desired. NMAX is used to specify the maximum number of nodes that are to be sorted. The default value of NMAX is all nodes that are active. If a set of nodes has been specified with the commands documented in the SELEct module, which is less than the total number of nodes in the compete FEA model, only those nodes will be sorted. If this has been done prior to sorting the node (e.g., to view a sub set of nodes), the

NALL command can be used prior to running the NSORT command to reset the node set to all of the nodes in the model. The NSORT command was entered as:

NSORT, UZ, 0, 1 <RETURN>.

Once the nodes were sorted, the PRDISP command was used to display the displacements of the nodes. The node with the maximum vertical displacement appeared at the top of the listing.

Next, the principal normal stresses were sorted and printed. This required the elements to sorted based on the stress data that was stored with the label "S1". Element sorting is done with the ESORT command, which has the format:

ESORT, LABEL, ORDER, KABS, EMAX.

The parameters for this command work just like those used with the NSORT command, except LABEL must be a label that was defined with the POST1 STRESS command (e.g., "S1"). Therefore, the command was entered as:

ESORT, S1, 0 <RETURN>.

Once, the elements were sorted, the PRSTR command was used to print out the stress data. PRSTR has the format:

PRESTR, LAB1, ..., LAB10.

The valid LABles are: (BLANK) for the first 10 stress items; GRP1, for items 1 to 10; GRP2, for items 11 to 20; GRP3, for items 21 to 30; GRP4, for items 31 to 40; or vector quantities TG, TF, EF, JS, HMAG, BMAG, or FMAG.

In addition to the PRESTR command, the PRNSTR command was used to print out the COMPonent and PRINcipal stresses. Before this command was used, the MIDdle command (located in the DEFInition module) was used to specify that the results were to be printed for the middle (average) nodal stresses.

In addition to printing out the vertical displacements and stresses, the reaction forces were also printed out to be sure that their totals were equal to fifteen pounds (15 lbs) for the mesh optimization runs, and twenty-five pounds (25 lbs) for the load test simulation runs. This was done with the POST1 PRRFOR command.

PLOTTING DISPLACEMENT AND STRESS RESULTS IN ANSYS POST PROCESSING

The POST1 routine's command which are used to produce stress and displacement plots are documented in the PLOT module. Like the plot commands that were discussed in Chapter XI, it is necessary to use the /SHOW command to specify the graphics device and plot parameters prior to issuing any plot command in post processing the analysis results. Likewise, the commands described in the on-line DOCU,GRPH (e.g., /PNUM, /NUMB, /PBC, and /PSYM) are used in POST1 just like they are in PREP7 to setup the details on the numbering and color filling of the entities displayed, and the boundary and coordinate systems displayed, etc.

The models' nodes and elements can be displayed with the same NPLOT, and EPLOT commands that are used in PREP7 preprocessing. If they have been sorted, or if sub-sets of total model's nodes and elements have been setup, only those entities will be display. The complete model can be restored by using the NALL and/or the EALL command.

The displacements resulting from the applied forces and pressures can be displayed with the PLDISPlacements command. The format for this command is:

PLDISP,KUND <RETURN>.

In this, KUND can be either: 0, which will display the displaced structure only; 1, which will overlay the displaced plot with the UNDisplaced structure; or, 2, which will overlay the displaced structure with an edge plot of the UNDisplaced structure. The last of these KUND options is preferred by the author, and therefore, for this thesis, displacement plots were made during interactive ANSYS post processing sessions by typing:

PLDISP,2 <RETURN>.

PLDISP is also controlled by the /DSCALE command. It is used to apply a scale factor to the distorted displacement plot being displayed. The format for this command is:

/DSCALE, WN, DMULT <RETURN>.

WN is the display window in which this is to be applied, which has a default of 1. The DMULTiplier parameter can be either 0 (or blank), 1, any user specified numerical value, or OFF. If 0 (default) is applied, the displacements will automatically be scaled such that the maximum displacement is displayed as five percent (5%) of the window. If 1 is used, the displacements will be scaled the same as the true geometry. If the user specifies a numerical value, that value will be used to scale the distorted display. Finally, if OFF is specified, the displacement plot will show no displacements.

The analysis' stress results are display with the PLNSTR plot command. The format for this command is:

PLNSTR, SLABL, KAVG <RETURN>.

The SLABL label can be any valid stress label that has been stored with the STRESS command, or from the following list: SX, SY, SZ, SXY, SYZ, SXZ, SIG1, SIG2, SIG3, S.I., SIGE, UX, UY, UZ, ROTX, ROTY, ROTZ, PREC, TEMP, VOLT, MAG, BX, BY, BZ, or BSUM. KAVG can be set to either 0 or 1. If it is set to 0, nodal stresses (SX, SY, ..., SIGE) will be averaged, but STRESS command items will not be averaged. If KAVG is set to 1, the opposite will occur (i.e., nodal stress will not and STRESS items will be averaged).

The appearance of the stress plots is effected by /NUMBer command, and by the selection of either vector or raster plotting, with the /SHOW command. The /NUMB command must be applied with either 1, for color and numbering, or 2, for color only. If it is applied with -1, for no color or numbering, or 2, for numbering only, no stress contours will be displayed. If vector plotting is specified (e.g., /SHOW,4107,,1), the stress contours will be displayed as line segments at each contours line, and will be marked with letters (A,B,C,...). If raster plotting is specified (e.g., /SHOW,4107,,0), the stress contours will be color filled between the contour lines. In either case, as long as the plots are made with a legend (i.e., /CLABEL,1,0 or 1, and not /CLABEL,1,-1), the contours will be labeled and a legend listing the stress value along each contour will be displayed.

It is impossible to display with grey scale copies of the color graphics displays, but, the ANSYS color displays of the stress contour plots uses color to very vividly show the gradients in resulting stress. The lowest values of stress are show in deep blue colors. The highest values of stress are show in red. The range from lowest to highest stress are displayed with blue, light blue, green, yellow, orange, and finally red. This allow the analyst to display the entire structure. The once the hot spots, that is the red areas of highest stress, are seen, the /ZOOM and sort commands can be used to display more limited areas showing the problem areas in the design.

In hind sight, if the author had defined different materials (with the same properties) for each of the individual pieces of the surface unit supports analyzed for this thesis, it would have been easier to view the resulting stresses in each piece of the supports. This could have been done by applying the ERSELect command with different MATerial numbers. This would have been by inputs in the form:

ERSEL,MAT,1 (or 2,3,4,...) <RETURN>.

This would have allowed the overlapping portions of the supports to be separated for display.

LOADING ANSYS ANALYSIS DATA INTO I-DEAS

In order to apply I-DEAS to the post processing of FEA results from an ANSYS solution run, the Engineering Analysis Family's Data Loader Module must be used to read the results into an I-DEAS universal file. This is done by entering the I-DEAS program, either with a new model file (e.g., RESULTS.MF*) or with an existing model file. Then Engineering Analysis, Data Loader is selected when prompted for the input of the desired I-DEAS software Family (EA DL <RETURN>).

Once the Data Loader Module is active, ANSYS (AN) is given as a response to the prompt for the analysis type. I-DEAS next prompts the user for the Data Loader processing options. These options are listed in Table 8. When this prompt was presented, AL was input as the response.

Following the input of the processing option, the user is asked to input the name of the POST data file. This is the name of the ANSYS results file, file 12. This will either be FILE12.DAT, if no command file was used; or it will be MODEL_NAME.F12 (e.g., N116.F12) if the batch command file was used to access ANSYS for solving the FEA models. Having responded to this prompt, I-DEAS will request a name for the universal file which is to be created. The default name of either FILE12.UAL or MODEL_NAME.UAL (e.g., N116.UAL) will be provided.

TABLE 8.

I-DEAS DATA LOADER PROCESSING OPTIONS

AL - BOTH GEOMETRY AND ANALYSIS DATA

GD - GEOMETRY DATA ONLY

NG - NODE GEOMETRY ONLY

EC - ELEMENT CONNECTIVITY ONLY

AD - ANALYSIS DATA

ND - NODAL DATA

ST - STRESS DATA

EF - ELEMENT FORCE DATA

After the specifications of the results and universal file names are made, I-DEAS asks the user if the data is to be loaded during the present interactive session, or if batch processing is desired. For this thesis, because the files that were to be loaded were relatively small, and because this work was being done after normal business hours, interactive data loading was selected. Having input this response, I-DEAS the performs the data loading task.

READING RESULTS DATA UNIVERSAL FILE INTO I-DEAS POST PROCESSING

The data loading task does not actually read the results data into the I-DEAS model file data base. It creates an I-DEAS universal file, into which the data are loaded. For the data to be brought into the desired I-DEAS model files, using the Engineering Analysis Pre/Post Processing Module to read the universal file. Therefore, once the Data Loader completes its creation of the universal file, control is returned to the menu selection level for the specification of a Engineering Analysis Module. The user then selects Pre/Post Processing. Note, that if a different model file data base was used for the data loading function then is desired for the final reading of the results data, the user should access that desired data base before proceeding with the reading of the universal file.

Within Post Processing, MANAGE_FE_MODELS, READ, UNIVERSAL is input. Then the name of the universal file (e.g., N116.UAL) is specified. I-DEAS then reads the universal file, which contains the ANSYS FEA model solution results into the model file data base (scratch files). During the read processing, the progress will be listed on the terminal's display screen. Once the data have been read, the user is asked if it is OK to load the graphics database. Once the user responds with YES (default), the graphics data is loaded into active memory, and display options can be processed.

ACCESSING ANALYSIS RESULTS WITH I-DEAS

In order to access the analysis results that have been transferred into I-DEAS for ANSYS, a GROUP of elements and/or nodes (depending on the results desired) must first be created. In the case of displacements, the group must consist of elements. As a first pass at displaying the displacement results, a group can be created containing all of the elements in the active FEA model. This can be done be entering:

/GR NE EL * Y <RETURN>.

This group can the be stored as a permanent group named "ALL ELEMENTS". This same method can be used to establish permanent group containing all of the model's nodes by substituting N and NODES in place of EL and ELEMENTS in the two command lines.

If the user wants to investigate the results based on smaller subgroups of nodes and/or elements, other groups can be created by using the other commands in the GROUP menu. These include ADD, REmove, or NEW, with the use of graphic cursor picking of entities. The picking can be based on VISIBLE entities, SCREEN_AREAs, or ENCLOSED_VOLUMES. Also, ATTRIBUTES can be used, and the entities can be RELATED_TO: ELEMENTS, CURVES, MESH_AREAs, MESH_VOLUMES, etc.

Once a GROUP is created and made CURRENT, the results can be interrogated. To do this, an ANALYSIS_DATASET_SELECTION must be made. This is done with the commands in the menu with this name. When the data was read into I-DEAS from the universal file, several datasets were created. Additional datasets can be CREATED, DELETED, RENAMED, and have their DESCRIPTIONs Listed and MOdified, with the commands in this menu.

Requesting a DIrectory of the existing datasets will reveal the sets that were read in from the ANSYS results. The analyses made for this thesis resulted in datasets containing the displacements, stresses, and reaction forces. Once one of these datasets is made CUrrent, the options presented in the POST_PROCESSING Module's main menu will change, based on the results that can be displayed from each dataset.

When a dataset containing displacements is made current, a DEFORMED_GEOMETRY option will be presented in the POST_PROCESSING main menu. Displacement data sets, as well as all other data sets, will also present command menus for ARROW_PLOTs, CONTOUR plots, XY-PLOTs, and if beam elements are being analyzed, BEAM_SHR_&_MOM_DIAGRAMS can be used to create shear_and_moment diagrams from stresses or element forces.

PRINTING ANALYSIS RESULTS WITH I-DEAS

In most cases, if the analysis has been solved using ANSYS, any listing or print out of the raw data results will have been performed with ANSYS during initial interactive post processing. But, it is possible to review this information using I-DEAS as well.

In the case of displacement results, the DEFORMED_GEOMETRY command menu contains a REPORT_DATA menu of commands which can be used to print the data to the computer terminal's screen, a disk file, or a printer. This command has several options which are controlled by the commands in this menu.

The RAW_DATA_SW is used to toggle on and off the printing of the complete data file for the nodal displacements. If this switch is set to OFF, only the magnitude of the complete displacement for each nodes displacement will be printed. If it is ON, all of the X, Y, Z, and rotational displacements will be printed.

The SUMMARY_SW is used to print a summary of the data's Maximum, Minimum, and Average values when it is ON.

The SORT_COLUMN command is used to sort the displacement data based on the data column specified. The columns are number from 1 to 7, in the following order: Node Numbers, Translations in X, Y, Z, and Rotations about X, Y, and Z. For this thesis, because translations in the Z direction were the most important, the data was sorted based on column 4. This placed the node with the largest translation in the Z direction at the top of the print out, followed by the rests of the nodes listed in descending order of magnitude.

The TITLE command allows the user to specify a title to be printed in the header information that is printed before the data is listed. The header information also contains the: revision of I-DEAS being used; data and time; model file's title; Group ID name; analysis dataset; report type (Deformed Geometry); units; dataset type (Displacements); and load set.

If the data is being sent to an attached printer, the LINES_PER_-PAGE_(FILE) command can be used to specify the number of lines that the printer is compatible with.

The STATUS command can be used to verify the settings of each of these report options commands.

PLOTTING DISPLACEMENT AND STRESS RESULTS WITH I-DEAS

The analysis results can be display graphically through the use of several display options that I-DEAS makes available in the Engineering Analysis Families Post Processing Module. As stated above for the printing of the results data, the options presented in the POST_PROCES--SING Module's main menu will change, based on the results that can be displayed from each dataset.

The DEFORMED_GEOMETRY command calls up a menu that presents the commands and options that can be used to control and execute the display of the displaced geometry of an FEA model for which the displacements dataset is active. The displacements can be displayed as LINE, FREE_FACE, CONTINUOUS_TONE, HIDDEN_LINE, SHADED_IMAGE, or CUTTING_PLANE_DISPLAY (if the model is constructed with solid elements) formats. Also, ANIMATION sequences for the displacement results can be created, if the terminal being used supports animation. In each case, there are several options that can be set to customize the display results.

PERCENT_DEFORMATION and SCALE_OF_DEFORMATION are both used to exaggerate the displayed displacements, like ANSYS' /DSCALE command. The two commands act a toggle switches, when one is specified, the other is turned off. PERCENT_DEFORMATION is used to scale the displayed deformation as a percentage of the screen. For example, 10.0 would be input for 10% of the viewport's size. SCALE_OF_DEFORMATION is used to specify a scale factor which all the displacements will be multiplied by. If the scale factor is specified as 10.0, all displacements will be multiply by 10 and then displayed.

RELATIVE_DISPLACEMENT allows the displacements to be displayed showing all displacements relative to the position of a specified node or location in space. This allows the user apply NODE_OFFSET, and selecting a node, using any of I-DEAS' standard entity specification or picking options, relative to which the other nodes displacements will be displayed. Or, MANUAL_OFFSET can be selected, and then a set of X, Y, and Z values can be input to specify the point in space about which the displacements will be displayed. This will yield a display with all displacements appearing to be displaced the amount specified as the offset values.

The DEFORMED_ATTRIBUTES and UNDEFORMED_ATTRIBUTES menus each provide the same options, effecting either the deformed or undeformed geometry as their names imply. They allow the user to specify that the geometry be displayed either as BOUNDARY_ONLY or as ELEMENTS. In the case of the undeformed geometry, its display can be turned off by selecting NO_UNDEFORMED. Each of these menus also provides for user specification of the LINESTYLE (solid of dashed) and COLOR for the geometry displayed.

ARROW_PLOT provides a menu of that presents the commands and options that can be used to control and execute the display of selected analysis results in a format in which they are displayed graphically as arrows placed at each node. The arrows will be displayed in a range of colors representing the magnitude of the data being displayed. A color key is displayed on the screen labeling the magnitude of the data at the ends of the range the colors represent. Arrow plots can be created for either FREE_FACE or CUTTING_PLANE_DISPLAY formats, similar to the DEFORMED_GEOMETRY plots. The user must specify the results dataset that is to be used. This is done with the ON_DEFORMED_GEOMETRY sub-menu's ANALYSIS_DATASET_SELECT, USE command. The plot can be created with or without displaying the displaced geometry in the deformed format. This is specified by either turning the ON_DEFORMED_GEOMETRY ON or OFF. The other commands presented in this sub-menu allow the user to customized the display of deformed geometry with the same options a discussed for the DEFORMED_GEOMETRY plots. The user may specify the FRAME_OF REFERENCE for the displacements from the available coordinate systems (GLOBAL, LOCAL, DISPLACEMENT, ELEMENT, MATERIAL, or SKETCH_REF_FRAME); SHELL_SURFACE for shell elements (TOP, MIDDLE, BOTTOM_SURFACEs, BENDING_VALUES, or LAMINATE_PLY); and the data that are to be plotted, all from within the ARROW_PLOT, DATA_COMPONENT sub-menu. The data that is available for selection from the DATA_COMPONENT menu depends on which ANALYSIS_DATASET_SELECTION, CURRENT specification has been set. ARROW_PLOT also provides for the specification of the LOCATION_OF_ARROW, ARROW_LENGTH, COLOR_OF_ARROWS, to further customize the display of arrow plots. It also allows the element boundaries to be toggled On and OFF with an ELEMENT_BOUNDARY_SW.

CONTOUR plots can be created to display displacement, force, and stress results yielding contour plots similar to ANSYS' PLNSTR plots. The options available for customizing I-DEAS' CONTOUR plots are the same as those described for the ARROW_PLOT commands. FREE_FACE CONTOUR plots yield results are similar to ANSYS' PLNSTR plots which are produced in vector form (i.e., without color filling). In addition to the formats that the DEFORMED_GEOMETRY and ARROW_PLOT commands provide, the CONTOUR plot options also present a FRINGE_CONTOUR menu. This option will yield results similar to ANSYS' PLNSTR plots which are produced in raster form (i.e., with color filling). Due to the difficulties faced in reproducing color plots in black and white for report duplication, it is desirable to be able to produce contour plots with "grayscale" shading. I-DEAS provides for this by toggling the GRAYSCALE_SW to ON, it is presented in several menus (e.g., CONTOUR, FRINGE_CONTOURS). This allows <u>all</u> formats of the contour plots to be produced in grayscale.

Experimentation with this shows that this works well when applied with CONTINUOUS_TONE CONTOUR plotting. But do to the limited number of gray tones available, and their similarity, it was still necessary to produce contour plots in color to be sure that the grayscale plots would be properly interpreted, and to add some annotation to the black and white plots for this test.

I-DEAS Post Processing Module provides a unique command menu for the graphical display and printout of results based on user specified criterion. This is presented in the menu accessed with the appropriate command named CRITERION. This menu allows the user to produce reports and plots in the same formats as discussed for the DEFORMED_GEOMETRY, ARROW_PLOT, and CONTOUR commands, with the additional ability to specify criterion with which the data is interrogated during the plot or report processing. The criterion is specified with the commands presented in the CRITERION, DATA_RANGE menu. This menu contains options for: ABOVE_DATA_VALUE, ABOVE_DATA_PERCENT, BELOW_DATA_VALUE, BELOW_DATA_PERCENT, INSIDE_DATA_VALUE, INSIDE_DATA_PERCENT, OUTSIDE_DATA_VALUE, and OUTSIDE_DATA_PERCENT.

I-DEAS Post Processing Module also provides commands useful for producing reports and plots in XY_PLOT and BEAM_SHR_&_MOM_DIAGRAM formats. These options were not required for this thesis, and thus there use will not be discussed here. But, users of the I-DEAS FEA analysis system should be aware that they are available, and their use should be explored where appropriate.

BIBLIOGRAPHY

Bauld, N. R. Jr., Mechanics of Materials, Brooks/Cole Engineering Division, Monterey, California, 1982.

Boresi, A. P., Sidebottom, O. M., Advanced Mechanics of Materials, Fourth Edition, John Wiley & Sons, New York, 1985.

DeSalvo, G. J., Gorman, R. W., ANSYS Engineering Analysis System User's Manual, Swanson Analysis Systems, Inc., Houston, Pennsylvania, 1987.

Materials Selector, Materials Engineering magazine, Penton Publishing, Inc., December, 1989.

VITA

The author, Jeffrey Alan Kern, is the son of Roy W. Kern and Julia (Easterday) Kern. He was born February 5, 1962, in Fort Wayne, Indiana.

His elementary education was obtained in the Fort Wayne Community School System, Fort Wayne, Indiana, and secondary education at R. Nelson Snider High School, Fort Wayne, Indiana, where he graduated in 1981.

In August, 1981, he enter Purdue University at the Indiana University - Purdue University campus in Fort Wayne, Indiana (IPFW). In June, 1983, he was married to Lee (Langlet) Kern. In June, 1985, he received the degree of Bachelor of Science in Mechanical Engineering from Purdue University, on its main campus in West Lafayette, Indiana.

In June, 1985, he joined GE Appliances as an engineer, and began his studies toward a Master of Science degree at the University of Louisville, in Louisville, Kentucky. In January, 1987, Lee Kern gave birth to their first daughter, Mercedes Lynn Kern, and in November, 1990, to their second daughter, Emerald Jane Kern. At the time this thesis was published, he had received two United States Patents, assigned to the General Electric Company: Temperature Sensor Assembly for an Automatic Surface Unit, Patent Number 4,812,624; and Reversible Turbine Pump, Patent Number 4,923,366.