Feature-Based Geometric Modeling using B-Spline Surfaces
and a Natural Language Approach

by

Ashit R. Gandhi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Mechanical Engineering

APPROVED:

Dr. A. Mykleblast, Chairman

Dr. M. P. Deisenroth

Dr. R. H. Fries

Dr. J. R. Mahan

Dr. C. F. Reinhart

April 14, 1989

Blacksburg, Virginia
Feature-Based Geometric Modeling using B-Spline Surfaces
and a Natural Language Approach

by

Ashit R. Gandhi

Dr. A. Myklebust, Chairman
Mechanical Engineering

(ABSTRACT)

Traditionally, design geometries are represented using orthographic views which require a human being to interpret them and recognize geometric features to evaluate the design. Solid modeling systems have made the task somewhat easier, but they often require tedious and complex operations using simple geometric primitives. This has led to investigation of modeling systems which allow the creation of assemblies the way engineers conceive them - as features.

To be able to efficiently describe models in terms of features, a rich set of feature descriptors is necessary. An exhaustive study of English words describing form, shape, feature, shape altering transformations and surface conditions was done to establish a wide vocabulary for feature based description. Classification based on topology and form was done and prototype constraint relations were implemented to differentiate between some features. A feature is created from a topological group by computing points and interpolating them with uniform rational B-spline patches. Control points needed to compute the patches are computed from the interpolation points by an inverse relation. A designer-oriented modeling language, based on simple English syntax, was devised to specify procedures to be executed by the modeler in order to create features using minimal user input.
I would like to extend my deepest gratitude to the chairman of my supervisory committee, Dr. Arvid Myklebust, for his guidance, encouragement and support throughout my graduate studies.

I wish to express my sincere appreciation to Drs. Deisenroth, Fries, Mahan and Reinholtz, who served as members of my committee and offered valuable advise and support.

For her infinite support, I wish to thank my wife, , without whom this work would have taken longer. Her devotion and sacrifices have helped me overcome many hard and trying moments during the last two years.

Last and foremost, I am indebted to my parents and family members for their unfailing love and support which allowed me to make my dreams come true. I know, that no one will be happier than them to see this work completed.
Table of Contents

Chapter 1: Introduction ................................................... 1

1.1 Overview ........................................................................ 1

1.1.1 Need to Step-up Automation ......................................... 1

1.1.2 Conventional CAD/CAM systems ..................................... 2

1.1.3 Trends in CAD/CAM system development ......................... 3

1.1.4 Advantages of Feature Based Modeling ............................. 5

1.2 Literature Review ...................................................... 6

1.2.1 Survey Papers in Modeling .......................................... 6

1.2.2 Improvement Techniques in CAD ................................. 7

1.2.3 Feature Based Geometric Modeling ................................. 7

1.2.4 Surface Modeling Techniques ........................................ 9

1.3 Problem Definition ................................................... 9

1.3.1 A Feature Dictionary and Library ................................. 11

1.3.2 Feature Parameterization .......................................... 11

1.3.3 Feature Definition through Incomplete Specification ............ 12

1.3.4 Geometric Modeling of Features .................................... 12

1.3.5 Designer-Oriented Modeling Language ............................ 13
Table of Contents

Userid Requirements ................................................. 304
File Requirements .................................................... 304
Hardware Requirement ............................................... 304
Running FeatureMod .................................................. 305
FeatureMod Functions ................................................ 306
Overview ................................................................... 306
Starting A New Model .............................................. 307
Reading Existing Model Files ....................................... 308
Defining An Object ................................................... 309
Defining An Assembly ............................................... 310
Adding To an Object or Assembly ......................... 311
Deleting an Object or Assembly ............................... 312
Displaying an Object or Assembly ............................ 313
Model Inquiry ....................................................... 314
Displaying Multiple Views ....................................... 315
Displaying Single Views ........................................... 315
Displaying A Hidden Surface Image ............................ 316
Displaying A Shaded Image ...................................... 317
Performing Three Dimensional Viewing Transformations 318

Appendix D - Feature Inquiry Program ....................... 319

Vita ................................................................ 320

Table of Contents
## List of Illustrations

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Parameters defining Major Topology Groups - I.</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Parameters defining Major Topology Groups - II.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Parameters defining Major Topology Groups - III.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Parameters defining Major Topology Groups - IV.</td>
<td>24</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Parameters defining Major Topology Groups - V.</td>
<td>25</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Parameters defining Other Features - I.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Parameters defining Other Features - II.</td>
<td>27</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Common Profiles and Parameters Defining Them.</td>
<td>28</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Feature Classification by Shape Form.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Examples for General Form Features.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Examples for Special Form Features.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Examples for Superficial Form Features - I.</td>
<td>40</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Examples for Superficial Form Features - II.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Examples for Superficial Form Features - III.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Example Features Based On Suggested Guidelines - I.</td>
<td>54</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Example Features Based On Suggested Guidelines - II.</td>
<td>55</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Some Acceptable Dimensions for a Rectangular Bar.</td>
<td>62</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Input Processing by UML.</td>
<td>71</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Example of Forward Modeling.</td>
<td>82</td>
</tr>
</tbody>
</table>
Figure 20. Example of Inverse Modeling. ........................................ 85
Figure 21. Examples of Patches Using Inverse Solution. .................... 87
Figure 22. Examples of Features Using Inverse Solution. .................... 88
Figure 23. Examples of Features Using Inverse Solution. .................... 89
Figure 24. Data Flow Diagram of FeatureMod. .............................. 94
Figure 25. Reserved Areas on Display Screen. ............................. 96
Figure 26. Representative Object Hierarchy in FeatureMod. ................ 98
Figure 27. Menu Hierarchy in FeatureMod. ................................. 99
Figure 28. Data Storage in FeatureMod. ..................................... 101
Figure 29. Templates for Creating (a) Counter-bore and (b) Counter-sink. .. 103
Figure 30. Sample Templates for Creating Assemblies. ..................... 104
Figure 31. Features Describing a Pulley Mount - I. .......................... 108
Figure 32. Features Describing a Pulley Mount - II. ......................... 109
Figure 33. Features Describing a Flow Controlling Valve. .................. 111
Figure 34. A Pulley Mount Arrangement Created by FeatureMod - Front View. 112
Figure 35. A Pulley Mount Arrangement Created by FeatureMod - Top View. 113
Figure 36. A Pulley Mount Arrangement Created by FeatureMod - Right Side View. .................................................. 114
Figure 37. A Pulley Mount Arrangement Created by FeatureMod - Isometric View. .................................................. 115
List of Tables

Table 1. List of Features - I. ....................................... 17
Table 2. List of Features - II. ...................................... 18
Table 3. Feature Classification by Topology - I. ......................... 31
Table 4. Feature Classification by Topology - II. ........................ 32
Table 5. Feature Classification by Topology - III. ........................ 33
Table 6. Feature Classification by Topology - IV. ........................ 34
Table 7. Feature Classification by Topology - V. ........................ 35
Table 8. Feature Classification by Shape Form - I. ...................... 43
Table 9. Feature Classification by Shape Form - II. ........................ 44
Table 10. Feature Classification by Shape Form - III. ..................... 45
Table 11. Feature Classification by Shape Form - IV. ..................... 46
Table 12. Feature Classification by Shape Form - V. ...................... 47
Table 13. Feature Classification by Shape Form - VI. ...................... 48
Table 14. Suggested Mathematical Relations For Example Features - I. ........... 51
Table 15. Suggested Mathematical Relations For Example Features - II. ........... 52
Table 16. Suggested Mathematical Relations For Example Features - III. .......... 53
Table 17. Shape-Altering Words And Their Effect On Mathematical Rules. ........ 58
Table 18. Command Verbs - List and Classification. ...................... 68
Table 19. Commonly used Prepositions. .................................. 70
Table 20. List of Separators and Filler Words. ........................... 73
Chapter 1: Introduction

1.1 Overview

1.1.1 Need to Step-up Automation

Increasing competition throughout the world has put tremendous pressure on manufacturers to step up automation in order to make better products - and to make them more quickly and less expensively than before. This means reduced lead times for both design and manufacturing cycles and an increase in required levels of productivity and flexibility. This has led to an attempt to develop computer tools that will help automate, and hence speed up, the cycles. One way of achieving this is by integration of CAD and CAM. Yet a great many computer tools purporting to serve these ends automate only a small subset of production. What are touted as "computer-aided design" (CAD) systems, for example, usually are systems that offer dazzling graphics but few tools to improve the product that is on the drawing board.
"Computer-aided manufacturing" (CAM) often amounts to using a computer to create NC instructions and to run machine tools that are not integrated with other factory operations. Most importantly, CAD/CAM has so far done little to bridge the long-standing chasm between those who design the products and those who make them. The new generation of CAD/CAM will have to bring computer power to bear on far more of the design and production cycles.

1.1.2 Conventional CAD/CAM systems

One of the primary aims of geometric modeling is to provide a computer shape model that is as useful in design and manufacture as any physical prototype. Hence, one vital component of this stored description is a geometric model which contains information about the shape of the object under consideration. The richness and the completeness of the geometric model have a profound effect on the capabilities of the system.

In the earliest systems, data stored were only two-dimensional graphical entities which made up engineering drawings. Dimensions and notes on such drawings describe the geometric model, but their meaning is usually subjective and difficult for a computer to interpret. Such systems still predominate the field and lack the necessary 'intelligence' to review the engineering drawing, interpret it, and extract a set of useful information.

More recent systems have attempted to store descriptions of physical objects themselves. This description can be stored by using wire-frames, surface models or solid models and can be used for many different purposes such as generation of
engineering drawings and analyses and generation of NC instructions. An object is best modeled using solid models, since they contain maximum information that would help supporting engineering functions like mass property calculations, interference checks, etc. Of the several methods available for solid model construction, three methods are commonly used. These are sweep representation, B-Rep (boundary representation) and CSG (constructive solid geometry). B-Rep requires the creation of surface models and their conversion to solids. However, the methods by which surface models are created require a very high level of human interaction. A major disadvantage of using CSG is the limited number of primitives that act as the building blocks for the model. It is difficult for a designer to visualize complex objects in terms of these primitives and even harder to model. Hence, CSG often requires complex, tedious and time-consuming operations.

1.1.3 Trends in CAD/CAM system development

The ease with which geometric models can be created governs the speed of design. In trying to develop more sophisticated applications algorithms delivering higher levels of automation, it becomes clear that better schemes for geometric modeling and geometry representation must be devised. These schemes must overcome the deficiencies of many current modeling systems. Because design is iterative in nature, the topology, geometry or dimensioning of a geometric model must be modified many times during the design cycle. However, the effectiveness of future CAD/CAM systems will depend not only upon the ease with which geometric models can be created and modified, but also on the availability of convenient user
interfaces for automatic analyses, design evaluations, re-design suggestions and generation of manufacturing information.

This has led to research activity in feature-based geometric modeling systems which will contain more effective procedures for definitions and modification of geometric models. It is generally agreed that commonly used geometric primitives (like spheres, cylinders, cubes) are not a sufficiently rich vocabulary with which to describe engineering components. Designers use a richer vocabulary of higher level descriptors (like slots, keyways, chamfers, fillets) to convey information about geometry and topology of the component. It is the aim of feature-based modeling to allow engineers to design parts and assemblies by specifying readily understandable features or to extract feature information from an existing model. Thus common entities like holes, grooves, notches and bosses can be called for and manipulated without referring to individual geometric elements of which they are composed. A set of commonly used engineering shapes and features could be used as building blocks in creating parts or assemblies. The use of such higher level descriptors to convey information about the geometry and topology of a part allows both designers and production engineers to perform their separate tasks using a common language (of feature-based vocabulary) and to use the same solid model from conceptual design to the finished part. Ultimately, a feature model should provide sufficient information to drive NC tools such as milling machines, lathes and drill presses that make the part.

In addition to easing and speeding the modeling process, feature-based modeling systems are expected to have the highest level of integration of CAD and
CAM as yet. It will only be reasonable that such systems, in coming years, may replace solid modeling as a tool for geometric modeling.

1.1.4 Advantages of Feature Based Modeling

A major disadvantage of conventional CSG solid modelers is the availability of only a limited number of primitives that can be used in creating a part model. It is not easy to visualize complex shapes and even harder to model using these primitives. The availability of a large number of pre-defined generic features in a feature modeler will not only make the task of modeling easier but also faster, since they will eliminate several steps that are generally required to create a feature from basic primitives. Solid modelers are not capable of interactively changing a design. The use of different techniques such as parameterization of feature-based models may allow generation of variations of a basic design by simply changing these parameters. This will help users to understand the engineering relationship within their design and those that exist between different parts of an assembly. Perhaps the most important advantage of a feature-based design system is that all essential geometric and manufacturing information can reside within each feature. These features could then be used to perform design analysis and to recognize the kinds of machines and processes that are needed to produce the part.
1.2 **Literature Review**

Since one of the objects of this research is to develop a new technique to create and store geometric models, it is necessary to review work done by different people in addressing various issues of geometric modeling and improvement techniques in CAD/CAM. The papers discussed in this review are grouped according to the issues they address.

1.2.1 **Survey Papers in Modeling**

The following papers review the growing interest in feature-based modeling. Brody (1987) provides a good introduction to feature-based modeling, plus a preliminary study of research by individuals and corporations in that area. Woo (1977) discusses the problems and techniques of shape description for mechanical components. Allen (1984) describes some schemes used for representing geometry on solid modeling systems. He also describes some construction and editing techniques, which may be used to create geometry. Pratt (1988) has done a survey of different approaches to feature-based modeling techniques and has tried to find an optimal approach to design with features and interfacing to automated applications.
1.2.2 Improvement Techniques in CAD

Work on improving techniques in CAD has been done by several researchers. Hatvany (1984) identified knowledge-based techniques as the most important tool for CAD systems of the future. Davidson (1985) has drawn up a list of requirements for a CAD decision-making tool. Serrano and Gossard (1986) have developed interactive software which integrates mathematical models with solid models to aid in the preliminary design of mechanical components.

In order to combine the use of expert systems with geometric modeling, Sankar and Myklebust (1989) have developed a prototype expert system for automatic generation of interfaces between design application programs and CAD/CAM systems. New computer languages are being developed by researchers to improve the current techniques in CAD. Lai (1987) and Wilson (1987) have developed a "Function-Description" language which describes the function of relationships between elements in a mechanical design. Takase and Nakajima (1984) developed a "Feature-Description" language for describing and modeling assembled machines.

1.2.3 Feature Based Geometric Modeling

The area of feature-based geometric modeling is relatively new and the literature is somewhat sparse, compared to information available regarding improvement of techniques used in CAD/CAM. Significant work in feature-based modeling has been done by Luby, Dixon and Simmons (1986). They have created and used a features database for evaluation of manufacturability of castings. This
database, however, appears to consist of a relatively small number of features. They also do not mention whether their program performs addition and subtraction of the features being added to the component. Dixon and Simmons (1986) have worked in the area of creating a features database and talk about the use of such databases. Dixon and Dym (1986) have done some work on geometric representation and reasoning in design. Libardi et al. (1986) have tried to apply feature-based modeling techniques to extrusions. Nnaji and Vishnu (1987) have investigated methods for representing fundamental geometric data and for developing an algorithm for automatic conversion of wire-frame data to solid geometry. Light and Gossard (1982) have developed flexible procedures for definition and modification of geometric models. Gossard, Zuffante and Sakurai (1988) have treated dimensions, tolerances, and features in solid models. Dimensions are represented by a relative position operator and changes in dimensional values automatically translate into changes in geometry and topology. More recently, Cunningham and Dixon (1988) have investigated the origin of features and their role in design and manufacturing activities. However, rules and guidelines for such process-activities generate a very limited set of features. Tikerpuu and Ullman (1988) have established representations for a general description of a design object’s form and functions in terms of the state of design. Shah and Rogers (1988) have grouped product information into sets based on the meaning of the information. This helps in the automation of engineering tasks. Shah, Bhatnagar and Hsiao (1988) have developed a feature mapping and application shell by implementing specific applications such as Group Technology coding and Manufacturing Process selections.

It is apparent that although there is a lot of interest in developing feature-based modeling techniques, there is little consensus on how to approach the problem. It is
also apparent that researchers have selected their own way of formulating such techniques and that no comprehensive solutions have surfaced.

1.2.4 Surface Modeling Techniques

Geometric modeling and its techniques have evolved over the last 15 years. This documentation covers the formulation of different kinds of curves and surfaces, their approximations, subdivisions, placement of knots, smoothing and parameterization. A variety of information can be found on this subject. Of particular interest is the work on B-spline curves and surfaces by Riesenfeld (1973) and Versprille (1975). Both have used B-spline surfaces for computer-aided design applications. Tiller (1983) and Piegl (1987) have used rational B-splines for representing quadric primitives and free-form surfaces. Formulations for different modeling schemes and use of different surfaces are well documented by Mortenson (1985), Faux and Pratt (1979), Barsky (1988), Bartels et al. (1988), Barnhill and Boehm (1983), Greville (1969), Schoenberg (1969), Brodlie (1980), Gardan (1985) and Bezier (1986).

1.3 Problem Definition

Most research into the issues of feature representations has been in the area of feature extraction from existing CSG or boundary representation models, geometric tolerances and manufacturing evaluation of designs. Little thought has
been given to designing with features from the outset, and the manner in which they are defined or created. Current methods for feature definitions are application specific and do not provide designers with generalized procedures for feature creation. In addition, these procedures can be applied to only a limited number of features and hence restrict the ability of a designer to produce detailed models. Hence, development of generalized schemes to create shapes and features is necessary.

A designer’s understanding of a design feature originates from its English and technical definitions. It forms the basis by which he describes a mechanical assembly. He uses simple English words to convey information about the part’s size, shape and functions. Since designers find it convenient to communicate in this manner, the use of English words for feature creation and representation is a logical move towards easing the task of feature creation.

Every mechanical part consists of a unique set of features. Such sets may be similar for some parts, but very rarely the same. In many cases this set must be modified for appropriate use. Such modifications or shape alterations are further described by additional English words. Thus the vocabulary involved in the complete description of features is very large. The primary intention of this work is to lay a foundation for definition and creation of a geometric model using features and to identify an approach that makes best use of the vast vocabulary among designers. Some objectives of this research were:

1. To create an exhaustive library of design features based on English, technical and non-technical terms for shape and form together with the vocabulary for their modification and use.
2. To classify the features by form and topology.
3. To define the features parametrically.
4. To produce suitable feature surface models based on incomplete specification of parameters.

This results in several areas which must be addressed individually and in detail.

1.3.1 A Feature Dictionary and Library

Although many people talk about designing with shapes and features, there has been no exhaustive research on a feature vocabulary which relate the English definition of a feature to its mathematical characterization. The first aim of this research is to create an exhaustive dictionary of geometrical terms that relate to features. This dictionary will contain English definitions of the geometrical terms and will contain information on their synonyms. For ease in feature selection and modeling, features will be classified into groups according to topology and form. This vocabulary will provide a base for further research.

1.3.2 Feature Parameterization

A feature model that can be used with different objects and perform different functions must have versatility and must be able to adapt to changes in design. Any change should be accomplished quickly and with ease. More importantly, this change must be reflected in all related dimensions of the feature and other
components that are related to it. Parameterization of models meets these requirements and allows the feature to have different dimensions, orientations and variable relative dimensions. Each topology group will be parameterized and will be assigned a set of parameters that can be used to create and modify a feature model.

1.3.3 Feature Definition through Incomplete Specification

Depending on its function or application, a feature may have different dimensions. However, these dimensions may be governed by different mathematical rules resulting from existing geometric constraints or those specified by a designer. Shape altering words can be used to change mathematical rules that apply to a feature. A feature must satisfy all these rules but may have any dimensions within the constraints of its definition and mathematical rules. One can take advantage of such rules and use them effectively in creation of feature models based on incomplete specifications of parameters. A methodology to solve for unspecified parameters based on mathematical inequalities will be developed and presented.

1.3.4 Geometric Modeling of Features

Another objective of this research is to mathematically map the verbal and mathematical description of each feature or a class of features to a surface description. Geometric models would be based on parametric definition of the feature class. A method will be developed that will use the points computed from the
parametric definition of a feature and interpolate them to produce uniform rational
B-spline surface models that pass through these points.

1.3.5 Designer-Oriented Modeling Language

Conventional geometric modelers are menu driven, require large amounts of
input data and do not allow data transfer between two different programming
environments. A designer-oriented modeling language, based on simple English
syntax, will be devised to specify procedures to be executed by the modeler in order
to create features using minimal user input. One major advantage of this will be to
allow the modeler to operate in other programming environments.

1.3.6 Other Support Work

A high-level environment will be created to impart some degree of intelligence
to the modeler. Rules will be used to create an inheritance network (to determine
parameter values inherited from a parent feature) and cognition rules (to see if a
feature-parameter is used in a valid manner). A modeling environment for creating,
modifying and deleting features, will be created. Object-oriented programming
techniques will be used to support user definition of form features that can be
positioned and manipulated in a logical manner. The IBM version of PHIGS,
graPHIGS, will be used to provide device independent graphics support.
1.4 Definition of a Feature

Feature-based modeling has applications in both design and manufacturing cycles of product development. In design it can be used for object modeling, mass property analysis, interference checks and failure analysis. In manufacturing it has applications in process planning, manufacturability evaluation, cost analysis, etc. The most general definition of a "feature" is "any entity, geometric or otherwise, that influences any step of the product development cycle". However, for practical purposes, a "feature" definition varies with the application. Most applications do not consider both aspects of feature design. Thus most definitions of a "feature" are limited to the particular application for which it is being used. Cunningham (1988) defines a feature as "a higher order abstract geometric form or entity that is used in reasoning about topology and geometry". Shah (1988) defines features as "recurring patterns of information related to a part's description". The scope of this research is limited to use of features in geometric modeling. It does not take into consideration any manufacturing features. For this work, a feature will be defined as a geometric entity that defines the attributes of a part's nominal size, geometry, shape and topology. The emphasis will be on geometric subsets of a part that can be recognized and readily specified by a designer.
2.1 Feature Vocabulary

To ease modeling, a designer should be allowed to create parts the same way he perceives them - as features. The geometric reasoning process that a designer uses in visualizing a conceptual design should also be reflected in the modeling process. For most engineering applications, the conceptual design consists of complex shapes and features. The vocabulary used to describe such complex features is large and varied. If a designer is to efficiently describe a geometric part in terms of features, a rich set of feature descriptors is necessary. This means that the set of features available to a designer must be a complete and natural set. If the list is complete, a designer must be able to completely describe a part by a set of features that form the list. In addition, any word that a designer uses to describe shape or form should be a feature or a synonym of a feature. Hence, an exhaustive study of English words describing shape and form of engineering features was done to establish a wide vocabulary for feature-based description. This 373 word set
should account for almost all feature shapes and form required to model an engineering part. To help a designer decide which feature is best suited for a specific application, each feature has been described by means of its English, technical or non-technical definitions. These will help him better understand the feature's functions and characteristics. For example, a hole is defined as a cavity or aperture in a solid mass or body and performs the function of creating a hollow space in the part. A wedge is a V-shaped piece of solid material and is used in holding different components of a part in place. Many features possess similar characteristics and can be used to perform similar functions. Such features are synonyms of each other and can be used interchangeably without any change in the model geometry. Such closely related features have been placed in synonym lists and give a designer more choice of features for specific applications. For example, the features packing, pad, padding and wadding are synonyms of each other. A designer may choose to use one to perform the function of cushioning. However, if he is unaware of the function of a packing, there is a possibility that he knows the meaning of one of its synonyms. A designer should now be able to describe any part with a set of features or shapes from this vocabulary. Tables 1 and 2 contain the complete list of all the features that have been considered for this work. This list and the definitions were compiled using various sources. Some notable references were Oberg et al. (1988), Parker (1989), French and Svensen (1966), Yaslow (1969), Horner and Abbey (1952), Del Vecchio (1962), Nayler and Nayler (1967) and Giachino and Beukema (1964).
| 1. abutment | 49. bushing |
| 2. aiguille | 50. button |
| 3. angle | 51. cable |
| 4. annular-hole | 52. cage |
| 5. aperture | 53. cam |
| 6. apex | 54. can |
| 7. apron | 55. canal |
| 8. arbor | 56. cantilever |
| 9. arbor-hole | 57. cap |
| 10. arc | 58. capsule |
| 11. arch | 59. cartridge |
| 12. arm | 60. case |
| 13. arris | 61. casing |
| 14. astragal | 62. cave |
| 15. axle | 63. cavity |
| 16. baffle-plate | 64. cell |
| 17. ball | 65. chamber |
| 18. band | 66. chamfer |
| 19. bar | 67. channel |
| 20. barrel | 68. chase |
| 21. base | 69. chased-groove |
| 22. basket | 70. chassis |
| 23. bead | 71. chimney |
| 24. beam | 72. chord |
| 25. bed | 73. chute |
| 26. bedding | 74. cleat |
| 27. belt | 75. cleavage |
| 28. bench | 76. cleft |
| 29. bend | 77. clevis |
| 30. bevel | 78. cone |
| 31. billet | 79. clough |
| 32. block | 80. coil |
| 33. bolt | 81. coin |
| 34. bore | 82. collar |
| 35. boss | 83. collet |
| 36. bow | 84. column |
| 37. box | 85. compartment |
| 38. brace | 86. cone |
| 39. bracket | 87. conic |
| 40. break | 88. conical |
| 41. brick | 89. constriction |
| 42. broach | 90. convex |
| 43. bubble | 91. cord |
| 44. bucket | 92. core |
| 45. bulb | 93. corner |
| 46. bulge | 94. cotter |
| 47. bullet | 95. counter-bore |
| 48. bump | 96. counter-sink |

A Feature Dictionary and Library
<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature</th>
<th>Feature</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>192. key</td>
<td>238. parallelopiped</td>
<td>284. rim</td>
<td>329. stick</td>
</tr>
<tr>
<td>193. keyhole</td>
<td>239. parting</td>
<td>285. ring</td>
<td>330. stiffener</td>
</tr>
<tr>
<td>194. keyseat</td>
<td>240. partition</td>
<td>286. rivet</td>
<td>331. stopper</td>
</tr>
<tr>
<td>195. keyway</td>
<td>241. patch</td>
<td>287. rod</td>
<td>332. string</td>
</tr>
<tr>
<td>196. knob</td>
<td>242. pattern</td>
<td>288. roller</td>
<td>333. strip</td>
</tr>
<tr>
<td>197. knurl</td>
<td>243. peak</td>
<td>289. rope</td>
<td>334. stud</td>
</tr>
<tr>
<td>198. ladle</td>
<td>244. peen</td>
<td>290. round</td>
<td>335. stuffing</td>
</tr>
<tr>
<td>199. land</td>
<td>245. peg</td>
<td>291. rounding</td>
<td>336. swell</td>
</tr>
<tr>
<td>200. leaf</td>
<td>246. pellet</td>
<td>292. rung</td>
<td>337. swelling</td>
</tr>
<tr>
<td>201. ledge</td>
<td>247. perforation</td>
<td>293. rupture</td>
<td>338. t-bolt</td>
</tr>
<tr>
<td>202. leg</td>
<td>248. perimeter</td>
<td>294. rut</td>
<td>339. t-slot</td>
</tr>
<tr>
<td>203. line</td>
<td>249. pillar</td>
<td>295. scar</td>
<td>340. table</td>
</tr>
<tr>
<td>204. lip</td>
<td>250. pin</td>
<td>296. scoop</td>
<td>341. tack</td>
</tr>
<tr>
<td>205. lobe</td>
<td>251. pipe</td>
<td>297. screen</td>
<td>342. tang</td>
</tr>
<tr>
<td>206. loop</td>
<td>252. pit</td>
<td>298. screw</td>
<td>343. tank</td>
</tr>
<tr>
<td>207. lug</td>
<td>253. plane</td>
<td>299. seal</td>
<td>344. taper</td>
</tr>
<tr>
<td>208. mandrel</td>
<td>254. plate</td>
<td>300. seam</td>
<td>345. tapped-hole</td>
</tr>
<tr>
<td>209. manifold</td>
<td>255. plug</td>
<td>301. seat</td>
<td>346. tear</td>
</tr>
<tr>
<td>210. mask</td>
<td>256. plunger</td>
<td>302. separation</td>
<td>347. telescope</td>
</tr>
<tr>
<td>211. mass</td>
<td>257. pocket</td>
<td>303. shaft</td>
<td>348. tetrahedron</td>
</tr>
<tr>
<td>212. mat</td>
<td>258. pod</td>
<td>304. sharp</td>
<td>349. thickening</td>
</tr>
<tr>
<td>213. membrane</td>
<td>259. point</td>
<td>305. sheave</td>
<td>350. thread</td>
</tr>
<tr>
<td>214. mesh</td>
<td>260. pot</td>
<td>306. sheet</td>
<td>351. tile</td>
</tr>
<tr>
<td>215. messenger</td>
<td>261. pouch</td>
<td>307. shell</td>
<td>352. torus</td>
</tr>
<tr>
<td>216. mosaic</td>
<td>262. prism</td>
<td>308. shield</td>
<td>353. track</td>
</tr>
<tr>
<td>217. nail</td>
<td>263. probe</td>
<td>309. shim</td>
<td>354. trepan</td>
</tr>
<tr>
<td>218. neck</td>
<td>264. projection</td>
<td>310. shoulder</td>
<td>355. tube</td>
</tr>
<tr>
<td>219. needle</td>
<td>265. prominence</td>
<td>311. sieve</td>
<td>356. tunnel</td>
</tr>
<tr>
<td>220. nest</td>
<td>266. protrusion</td>
<td>312. slab</td>
<td>357. twine</td>
</tr>
<tr>
<td>221. net</td>
<td>267. protuberance</td>
<td>313. sleeve</td>
<td>358. twist</td>
</tr>
<tr>
<td>222. nile</td>
<td>268. pulley</td>
<td>314. sleevig</td>
<td>359. undercut</td>
</tr>
<tr>
<td>223. nipple</td>
<td>269. puncture</td>
<td>315. slit</td>
<td>360. valley</td>
</tr>
<tr>
<td>224. node</td>
<td>270. pyramid</td>
<td>316. slot</td>
<td>361. vane</td>
</tr>
<tr>
<td>225. notch</td>
<td>271. rabbet</td>
<td>317. socket</td>
<td>362. vein</td>
</tr>
<tr>
<td>226. nozzle</td>
<td>272. race</td>
<td>318. sphere</td>
<td>363. vent</td>
</tr>
<tr>
<td>227. nut</td>
<td>273. rail</td>
<td>319. spike</td>
<td>364. vertex</td>
</tr>
<tr>
<td>228. o-ring</td>
<td>274. raise</td>
<td>320. spindle</td>
<td>365. void</td>
</tr>
<tr>
<td>229. opening</td>
<td>275. ravine</td>
<td>321. spiral</td>
<td>366. wadding</td>
</tr>
<tr>
<td>230. orifice</td>
<td>276. seam</td>
<td>322. spline</td>
<td>367. washer</td>
</tr>
<tr>
<td>231. oval</td>
<td>277. receptacle</td>
<td>323. split</td>
<td>368. web</td>
</tr>
<tr>
<td>232. packing</td>
<td>278. recess</td>
<td>324. split-ring</td>
<td>369. wedge</td>
</tr>
<tr>
<td>233. pad</td>
<td>279. reel</td>
<td>325. spoke</td>
<td>370. weld</td>
</tr>
<tr>
<td>234. padding</td>
<td>280. rib</td>
<td>326. spool</td>
<td>371. well</td>
</tr>
<tr>
<td>235. nail</td>
<td>281. ribbon</td>
<td>327. spot-face</td>
<td>372. wheel</td>
</tr>
<tr>
<td>236. pallet</td>
<td>282. ridge</td>
<td>328. spring</td>
<td>373. wire</td>
</tr>
</tbody>
</table>

A Feature Dictionary and Library
2.2 Feature Parameterization

There is a need to develop a methodology that will allow easy definition and creation of features. Since designers find it easier to visualize a feature in terms of its parameters or dimensions, a method that allows definition of features in terms of their parameters would be desirable. Such parametric definitions of feature models have distinct advantages since many features have regular and symmetrical shapes which are dependent on these sets of parameters. Often these parameters represent direct geometrical or manufacturing information with only a small change in interpretation. For example, a 10-mm hole in design is translated into a manufacturing process (drilling) that requires a 10-mm drill bit. Defining a feature using such parameters will ultimately help in process planning procedures. Parametrics relate feature parameters directly to parametric equations defining the feature and indirectly to its surface characterization. This link not only allows quick creation of models but also easy and efficient modification of previously created features. Parameterization helps capture the intent of a designer and allows creation of geometric relations between different features in a part. In many cases changes in even one parameter affect values of related parameters, and each change is quickly transmitted through the entire model. For example, when a designer creates a 'through hole' in a plate, he creates a geometrical relation that requires that the length of the hole be equal to the thickness of the plate. If the designer later modifies the plate thickness, the hole dimensions will have to automatically adjust to reflect that change. Many designs vary only in a few parameters. They are basically the same part, with some size variations. With parametric modeling in effect, instead of producing separate designs for each part, only one design is necessary to design the
entire group. This significantly increases productivity by making the design cycle faster and less expensive. Use of parametrics also increases the ability of translating files between one CAD system and another. This is hence, a logical development to promote standards for such movement interfaces.

Parameters are prime factors in deciding the final shape or topology of a feature or class of features. Any group of features that possess a similar topology can be defined by the same or similar set of parameters. Once groups have been formed, it is straightforward to deduce and assign a set of parameters that can be used to represent each feature group. Such grouping and classification techniques are discussed in the next section. However, since one classification scheme uses topology as the basis of grouping, it makes the job of assigning feature parameters a little easier. Each topology group has been assigned parameters that can be used to define and create any feature that has been classified under that group. Figures 1 to 5 show each topology group, parameters that define the group and some example features from each group. Positive Boolean features are represented by solid lines whereas negative Boolean features are represented by dotted lines. Some features have specialized shapes and cannot be grouped into a general category. Such features can still be parameterized. Figures 6 and 7 show parameters defining some such features. Although most features can be grouped into one parameter or topology group, some can have various profiles and cross-sections that are described by different parameters. For example, a bar can have a circular cross-section (defined by radius) or a rectangular cross-section (defined by length and breadth). It is therefore necessary to parameterize different profiles. Some common profiles and their defining parameters are shown in Figure 8.
<table>
<thead>
<tr>
<th>TOPOLOGY GROUP</th>
<th>EXAMPLE FEATURES</th>
<th>PARAMETERS TO DEFINE GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPOLOGY - A</td>
<td>AXLE, BAR, CYLINDER, DISK, PLATE</td>
<td><img src="image" alt="Image of TOPOLOGY - A" /></td>
</tr>
<tr>
<td>TOPOLOGY - B</td>
<td>APERTURE, BORE, HOLE, ORIFICE, REAM</td>
<td><img src="image" alt="Image of TOPOLOGY - B" /></td>
</tr>
<tr>
<td>TOPOLOGY - C</td>
<td>BUSHING, CLIP, GLAND, RING, WASHER</td>
<td><img src="image" alt="Image of TOPOLOGY - C" /></td>
</tr>
<tr>
<td>TOPOLOGY - D</td>
<td>ARC, ARCH, CAVE, RACE, TUNNEL</td>
<td><img src="image" alt="Image of TOPOLOGY - D" /></td>
</tr>
<tr>
<td>TOPOLOGY - E</td>
<td>BAR, BEAM, BILLET, BOSS, CUBE</td>
<td><img src="image" alt="Image of TOPOLOGY - E" /></td>
</tr>
</tbody>
</table>

Figure 1. Parameters defining Major Topology Groups - I.
<table>
<thead>
<tr>
<th>TOPOLOGY GROUP</th>
<th>EXAMPLE FEATURES</th>
<th>PARAMETERS TO DEFINE GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPOLOGY - F</td>
<td>CHANNEL, GROOVE,</td>
<td>![Diagram of a channel, groove, keyway, pocket, and slot]</td>
</tr>
<tr>
<td></td>
<td>KEYWAY, POCKET, SLOT</td>
<td></td>
</tr>
<tr>
<td>TOPOLOGY - G</td>
<td>CASE, CELL, CHAMBER, FRAME, SHELL</td>
<td>![Diagram of a case, cell, chamber, frame, and shell]</td>
</tr>
<tr>
<td>TOPOLOGY - H</td>
<td>ANGLE, BEND, BRACKET, CORNER, ELBOW</td>
<td>![Diagram of an angle, bend, bracket, corner, and elbow]</td>
</tr>
<tr>
<td>TOPOLOGY - I</td>
<td>BALL, BULB, CROWN, GLOBULE, SPHERE</td>
<td>![Diagram of a ball, bulb, crown, globule, and sphere]</td>
</tr>
<tr>
<td>TOPOLOGY - J</td>
<td>BUMP, CUSP, HUMP, PROTRUSION, SWELLING</td>
<td>![Diagram of a bump, cusp, hump, protrusion, and swelling]</td>
</tr>
</tbody>
</table>

Figure 2. Parameters defining Major Topology Groups - II.

A Feature Dictionary and Library 22
<table>
<thead>
<tr>
<th>TOPOLOGY GROUP</th>
<th>EXAMPLE FEATURES</th>
<th>PARAMETERS TO DEFINE GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPOLOGY - K</td>
<td>APEX, PEAK,</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>SHARP, SPIKE,</td>
<td>A.R.</td>
</tr>
<tr>
<td></td>
<td>VERTEX</td>
<td></td>
</tr>
<tr>
<td>TOPOLOGY - L</td>
<td>CRATER, DEFLATION, DENT, DEPRESSION</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>TOPOLOGY - M</td>
<td>DIP, NOTCH</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>TOPOLOGY - N</td>
<td>FLUTE, RACE</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>TOPOLOGY - O</td>
<td>CAP, CURVE, DOME, HEMISPHERE</td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 3. Parameters defining Major Topology Groups - III.

A Feature Dictionary and Library 23
<table>
<thead>
<tr>
<th>TOPOLOGY GROUP</th>
<th>EXAMPLE FEATURES</th>
<th>PARAMETERS TO DEFINE GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPOLOGY - P</td>
<td>GRID, MESH, NEST, NET</td>
<td>![Image of grid topology]</td>
</tr>
<tr>
<td>TOPOLOGY - Q</td>
<td>PULLEY, REEL, SPOOL</td>
<td>![Image of pulley topology]</td>
</tr>
<tr>
<td>TOPOLOGY - R</td>
<td>CURL, HELIX, SPIRAL, SPRING</td>
<td>![Image of spiral topology]</td>
</tr>
<tr>
<td>TOPOLOGY - S</td>
<td>DONUT, RING, TORUS</td>
<td>![Image of donut topology]</td>
</tr>
</tbody>
</table>

Figure 4. Parameters defining Major Topology Groups - IV.
<table>
<thead>
<tr>
<th>TOPOLOGY GROUP</th>
<th>EXAMPLE FEATURES</th>
<th>PARAMETERS TO DEFINE GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPOLOGY - T</td>
<td>RIFLE, TAPPED-HOLE</td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>TOPOLOGY - U</td>
<td>SCREW</td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>TOPOLOGY - V</td>
<td>SEVEL, CONE, COTTER, TAPER</td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td>TOPOLOGY - W</td>
<td>ELLIPSE, ELLIPSOID, OVAL</td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Figure 5. Parameters defining Major Topology Groups - \( V \).
<table>
<thead>
<tr>
<th>OTHER FEATURES</th>
<th>PARAMETERS TO DEFINE FEATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTER-BORE</td>
<td>![COUNTER-BORE Diagram]</td>
</tr>
<tr>
<td></td>
<td>L₁, L₂, R₁, R₂</td>
</tr>
<tr>
<td>COUNTER-SINK</td>
<td>![COUNTER-SINK Diagram]</td>
</tr>
<tr>
<td></td>
<td>L₁, L₂, R₁, R₂</td>
</tr>
<tr>
<td>FLANGE</td>
<td>![FLANGE Diagram]</td>
</tr>
<tr>
<td></td>
<td>L₁, L₂, B₁, B₂, T</td>
</tr>
<tr>
<td>SPLINE</td>
<td>![SPLINE Diagram]</td>
</tr>
<tr>
<td></td>
<td>L₁, L₂, R₁, R₂, N</td>
</tr>
<tr>
<td>NECK</td>
<td>![NECK Diagram]</td>
</tr>
<tr>
<td></td>
<td>L₁, L₂, R₁, R₂</td>
</tr>
</tbody>
</table>

Figure 6. Parameters defining Other Features - I.
<table>
<thead>
<tr>
<th>OTHER FEATURES</th>
<th>PARAMETERS TO DEFINE FEATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILLET</td>
<td>![Fillet Diagram]</td>
</tr>
<tr>
<td>PRISM</td>
<td>![Prism Diagram]</td>
</tr>
<tr>
<td>WEDGE</td>
<td>![Wedge Diagram]</td>
</tr>
<tr>
<td>HEX-NUT</td>
<td>![Hex-Nut Diagram]</td>
</tr>
<tr>
<td>HEX-BOLT</td>
<td>![Hex-Bolt Diagram]</td>
</tr>
</tbody>
</table>

Figure 7. Parameters defining Other Features - II.
Figure 8. Common Profiles and Parameters Defining Them.
2.3 Feature Classification

Although each feature has its own function, many can be described or modeled using the same or very similar set of parameters. For example, a circular bar and a circular plate have different functions, but can be described using the same parameters, for example, radius and length/thickness. On the other hand many features can be used to model a part with specific functional requirements but are described by different parameters. For example, rotundity form can be created by a cylinder or a sphere both of which are described by different parameters. Since the number of features available is large, it is necessary to classify them into groups that will allow the designer to access them intelligently and efficiently. Hence, classification schemes must be devised which will make feature selection and accessing a simple task. The features were classified according to two criteria: topology and shape form. Features under the same topology group are governed by the same set of parameters that provide an easy way of defining them. Classification by shape form will help a designer select a feature that will impart the correct shape to the part. These two schemes will be combined together to allow feature selection based on form and creation based on topology.

2.3.1 Classification by Topology

For the purpose of this work all physical or natural features (Independent of size) which can be expressed as a set of points possessing similar properties of geometric configurations and have similar structural relationships will be termed as
Features that can be described by a similar set of parameters are classified under different groups which in turn are defined or described by parametric equations. Due to the nature of these groups, all features within a particular group would possess a similar topology. This makes implementation of modeling schemes straightforward, since a large number of features can be described using a single set of parametric equations. A detailed discussion about parameterizing each group has been presented earlier. The 23 major topological or parametric groups represent both positive and negative Boolean features. For example, groups A and B are both defined by parameters for radius (R) and length/thickness (L). However, both topologies represent different characteristics that will be imparted to the object that is being modeled. ‘Boss’ which is a part of group A is a positive feature and will act as a projection or a protrusion whereas a blind ‘hole’ classified under group B is a negative feature and will create a depression in the object. As discussed earlier, some features have different profiles and must be classified under more than one topology group. In such cases, correct interpretation of its topology must be arrived at by means of the shape profiles or shape descriptors. Such descriptors are discussed later in the chapter. Tables 3 through 7 contain the complete classification of the feature list by topology.

2.3.2 Classification by Form

Frequently a part that is being modeled has a specific geometric function. This function may put constraints on the shape-form which features that form the part may have. Although, many features have the capability of imparting the required form to the part, a designer should be given a wide choice to choose the most suitable
## Table 3. Feature Classification by Topology - I.

### Topology A

<table>
<thead>
<tr>
<th>Feature Dictionary</th>
<th>Feature Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>aiguille 1</td>
<td>core 17</td>
</tr>
<tr>
<td>arbor 2</td>
<td>cover 18</td>
</tr>
<tr>
<td>axle 3</td>
<td>cylinder 19</td>
</tr>
<tr>
<td>bar 4</td>
<td>dial 20</td>
</tr>
<tr>
<td>beam 5</td>
<td>diaphragm 21</td>
</tr>
<tr>
<td>boss 6</td>
<td>disk 22</td>
</tr>
<tr>
<td>broach 7</td>
<td>dish 23</td>
</tr>
<tr>
<td>bullet 8</td>
<td>drill 26</td>
</tr>
<tr>
<td>button 9</td>
<td>filling 27</td>
</tr>
<tr>
<td>cable 10</td>
<td>filter 28</td>
</tr>
<tr>
<td>cap 11</td>
<td>fin 29</td>
</tr>
<tr>
<td>chord 12</td>
<td>hatch 30</td>
</tr>
<tr>
<td>cleat 13</td>
<td>key 31</td>
</tr>
<tr>
<td>coin 14</td>
<td>leg 32</td>
</tr>
<tr>
<td>column 15</td>
<td>mandrel 33</td>
</tr>
<tr>
<td>cord 16</td>
<td>messenger 34</td>
</tr>
<tr>
<td>aperture 6</td>
<td>dilation 6</td>
</tr>
<tr>
<td>arbor-hole 7</td>
<td>hole 7</td>
</tr>
<tr>
<td>bore 8</td>
<td>inlet 9</td>
</tr>
<tr>
<td>broach 9</td>
<td>keyhole 10</td>
</tr>
<tr>
<td>constriction 10</td>
<td>opening 11</td>
</tr>
<tr>
<td>annular-hole 12</td>
<td>collar 13</td>
</tr>
<tr>
<td>baffle-plate 13</td>
<td>drum 14</td>
</tr>
<tr>
<td>barrel 14</td>
<td>duct 15</td>
</tr>
<tr>
<td>basket 15</td>
<td>filling 16</td>
</tr>
<tr>
<td>bucket 16</td>
<td>flue 17</td>
</tr>
<tr>
<td>bushing 17</td>
<td>gasket 18</td>
</tr>
<tr>
<td>can 18</td>
<td>gland 19</td>
</tr>
<tr>
<td>capsule 19</td>
<td>hub 20</td>
</tr>
<tr>
<td>cartridge 20</td>
<td>journal 21</td>
</tr>
<tr>
<td>chimney 21</td>
<td>lip 22</td>
</tr>
<tr>
<td>clip 22</td>
<td>loop 23</td>
</tr>
<tr>
<td>manifold 23</td>
<td>nipple 24</td>
</tr>
<tr>
<td>nozzle 25</td>
<td>o-ring 26</td>
</tr>
<tr>
<td>packing 27</td>
<td>pad 28</td>
</tr>
<tr>
<td>padding 29</td>
<td>pail 30</td>
</tr>
<tr>
<td>perimeter 31</td>
<td>pipe 32</td>
</tr>
<tr>
<td>probe 33</td>
<td>rim 34</td>
</tr>
<tr>
<td>ring 35</td>
<td>sleeving 36</td>
</tr>
<tr>
<td>split-ring 37</td>
<td>stopper 38</td>
</tr>
<tr>
<td>tube 39</td>
<td>vein 40</td>
</tr>
<tr>
<td>wadding 41</td>
<td>washer 42</td>
</tr>
</tbody>
</table>

### Topology B

### Topology C

A Feature Dictionary and Library 31
Table 4. Feature Classification by Topology - II.

**Topology D**

<table>
<thead>
<tr>
<th>1. arc</th>
<th>3. bow</th>
<th>5. crescent</th>
<th>7. race</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. arch</td>
<td>4. cave</td>
<td>6. curve</td>
<td>8. tunnel</td>
</tr>
</tbody>
</table>

**Topology E**

<table>
<thead>
<tr>
<th>1. abutment</th>
<th>18. column</th>
<th>35. hood</th>
<th>52. ribbon</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. apron</td>
<td>19. core</td>
<td>36. keel</td>
<td>53. rod</td>
</tr>
<tr>
<td>3. baffle-plate</td>
<td>20. cover</td>
<td>37. key</td>
<td>54. seal</td>
</tr>
<tr>
<td>4. band</td>
<td>21. cube</td>
<td>38. leaf</td>
<td>55. seat</td>
</tr>
<tr>
<td>5. bar</td>
<td>22. diaphragm</td>
<td>39. leg</td>
<td>56. sheet</td>
</tr>
<tr>
<td>6. base</td>
<td>23. face</td>
<td>40. mask</td>
<td>57. shield</td>
</tr>
<tr>
<td>7. beam</td>
<td>24. filling</td>
<td>41. mat</td>
<td>58. slab</td>
</tr>
<tr>
<td>8. bed</td>
<td>25. film</td>
<td>42. membrane</td>
<td>59. spoke</td>
</tr>
<tr>
<td>9. bedding</td>
<td>26. filter</td>
<td>43. packing</td>
<td>60. strip</td>
</tr>
<tr>
<td>10. belt</td>
<td>27. fin</td>
<td>44. pad</td>
<td>61. stuffing</td>
</tr>
<tr>
<td>11. bench</td>
<td>28. flake</td>
<td>45. padding</td>
<td>62. table</td>
</tr>
<tr>
<td>12. billet</td>
<td>29. foil</td>
<td>46. pallet</td>
<td>63. thickening</td>
</tr>
<tr>
<td>13. block</td>
<td>30. fold</td>
<td>47. partition</td>
<td>64. tile</td>
</tr>
<tr>
<td>14. boss</td>
<td>31. gib</td>
<td>48. patch</td>
<td>65. track</td>
</tr>
<tr>
<td>15. cantilever</td>
<td>32. guide</td>
<td>49. plane</td>
<td>66. vane</td>
</tr>
<tr>
<td>16. cap</td>
<td>33. gusset</td>
<td>50. plate</td>
<td>67. wadding</td>
</tr>
<tr>
<td>17. cleat</td>
<td>34. hatch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Topology F**

<table>
<thead>
<tr>
<th>1. canal</th>
<th>14. depth</th>
<th>27. kerf</th>
<th>39. recess</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. cavity</td>
<td>15. ditch</td>
<td>28. keyhole</td>
<td>40. ridge</td>
</tr>
<tr>
<td>3. channel</td>
<td>16. fissure</td>
<td>29. keyseat</td>
<td>41. rupture</td>
</tr>
<tr>
<td>4. channel</td>
<td>17. flume</td>
<td>30. keyway</td>
<td>42. rut</td>
</tr>
<tr>
<td>5. chute</td>
<td>18. fracture</td>
<td>31. notch</td>
<td>43. scar</td>
</tr>
<tr>
<td>6. cleavage</td>
<td>19. furrow</td>
<td>32. parting</td>
<td>44. separation</td>
</tr>
<tr>
<td>7. cleft</td>
<td>20. gap</td>
<td>33. pit</td>
<td>45. slit</td>
</tr>
<tr>
<td>8. clough</td>
<td>21. gorge</td>
<td>34. pocket</td>
<td>46. slot</td>
</tr>
<tr>
<td>9. collet</td>
<td>22. groove</td>
<td>35. pouch</td>
<td>47. socket</td>
</tr>
<tr>
<td>10. crack</td>
<td>23. guide</td>
<td>36. rabbet</td>
<td>48. split</td>
</tr>
<tr>
<td>11. crevice</td>
<td>24. gutter</td>
<td>37. ravine</td>
<td>49. valley</td>
</tr>
<tr>
<td>12. cut</td>
<td>25. hole</td>
<td>38. receptacle</td>
<td>50. vein</td>
</tr>
<tr>
<td>13. deep</td>
<td>26. hollow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5. Feature Classification by Topology - III.

**Topology G**

1. box  
2. cage  
3. case  
4. casing  
5. cell  
6. chamber  
7. chassis  
8. compartment  
9. frame  
10. shell  
11. tank

**Topology H**

1. angle  
2. arm  
3. arris  
4. bend  
5. brace  
6. bracket  
7. chamfer  
8. corner  
9. edge  
10. elbow  
11. line  
12. rib  
13. shoulder

**Topology I**

1. ball  
2. bead  
3. bubble  
4. bulb  
5. crown  
6. drop  
7. droplet  
8. globe  
9. globule  
10. mass  
11. pellet  
12. pot  
13. round  
14. rounding  
15. sphere  
16. tear

**Topology J**

1. astragal  
2. bulge  
3. bump  
4. chase  
5. convex  
6. cusp  
7. dune  
8. emboss  
9. entasis  
10. grain  
11. hump  
12. knob  
13. ledge  
14. lobe  
15. nile  
16. node  
17. pod  
18. projection  
19. prominence  
20. protrusion  
21. protuberance  
22. raise  
23. swell  
24. swelling

**Topology K**

1. apex  
2. peak  
3. point  
4. sharp  
5. spike  
6. taper  
7. vertex
Table 6. Feature Classification by Topology - IV.

Topology L

1. crater  
2. deflation

Topology M

1. dip  
2. notch

Topology N

1. flute  
2. race

Topology O

1. cap  
2. cup
3. dome  
4. hemisphere
5. ladle  
6. scoop

Topology P

1. fence  
2. gauze
3. grid
4. grill  
5. mesh
6. mosaic
7. nest
8. net
9. pattern
10. screen
11. sieve
12. web

Topology Q

1. pulley  
2. reel
3. spool

Topology R

1. coil  
2. curl
3. helix  
4. spiral
5. spring
6. thread
7. twist
Table 7. Feature Classification by Topology - V.

<table>
<thead>
<tr>
<th>Topology S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. donut</td>
</tr>
<tr>
<td>2. doughnut</td>
</tr>
<tr>
<td>3. ring</td>
</tr>
<tr>
<td>4. torus</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topology T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rifle</td>
</tr>
<tr>
<td>2. tapped-hole</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topology U</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. screw</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topology V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. bevel</td>
</tr>
<tr>
<td>2. cone</td>
</tr>
<tr>
<td>3. conic</td>
</tr>
<tr>
<td>4. conical</td>
</tr>
<tr>
<td>5. cotter</td>
</tr>
<tr>
<td>6. hub</td>
</tr>
<tr>
<td>7. taper</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topology W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ellipse</td>
</tr>
<tr>
<td>2. ellipsoid</td>
</tr>
<tr>
<td>3. oval</td>
</tr>
</tbody>
</table>
feature for that application. Hence, it is necessary that the available features be
classified according to the form that they have. Also, a designer may wish to browse
a group of features classified by form, based on the recollection of one feature in that
group. Figure 9 shows the general classification scheme adopted for grouping
features according to their form. Shape forms can generally be classified into three
major groups. The primary or general form group gives a designer a general idea
of the presence of form or regularity. In this case all features are classified under
presence of form and regularity. The secondary or special form group represents the
nature of the whole feature. The tertiary or superficial form group represents the
superficial form that will be imparted to the part. Each of these groups is further
classified. The special form group is divided into five sub-groups that represent the
shape of the feature. One sub-group, circularity, is further subdivided into simple and
complex types. The superficial form group is divided into 14 categories, based on the
superficial characteristics that a feature may have. Each feature in the list was
examined and tested for all groups and subgroups. Some were classified under one
group while others were found to be eligible for numerous groups. For example, the
feature 'sphere' can be used not only to give rotundity to the part but also simple
circularity and convexity. Another example is a hemisphere which can additionally
perform the function of imparting concavity to the part. Figures 10 to 14 show some
examples in each category. A user may find it easier to understand a shape form
based on use of some adjectives or verbs. Thus adjectives or verbs that may
represent a shape form are also classified under that shape form. Tables 8 through
13 contain the complete classification of the feature list by shape form.
Figure 9. Feature Classification by Shape Form.
Figure 10. Examples for General Form Features.
Figure 11. Examples for Special Form Features.
Figure 12. Examples for Superficial Form Features - I.
Figure 13. Examples for Superficial Form Features - II.
Figure 14. Examples for Superficial Form Features - III.
### Table 8. Feature Classification by Form - I.

#### Form Group Rotundity

<table>
<thead>
<tr>
<th>Feature Class</th>
<th>Feature Dictionary and Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. aiguille</td>
<td>45. cord</td>
</tr>
<tr>
<td>2. annular-hole</td>
<td>46. core</td>
</tr>
<tr>
<td>3. aperture</td>
<td>47. cotter</td>
</tr>
<tr>
<td>4. apex</td>
<td>48. counter-bore</td>
</tr>
<tr>
<td>5. arbor</td>
<td>49. cover</td>
</tr>
<tr>
<td>6. arbor-hole</td>
<td>50. crown</td>
</tr>
<tr>
<td>7. astragal</td>
<td>51. cusp</td>
</tr>
<tr>
<td>8. axle</td>
<td>52. cylinder</td>
</tr>
<tr>
<td>9. baffle-plate</td>
<td>53. deflation</td>
</tr>
<tr>
<td>10. ball</td>
<td>54. dent</td>
</tr>
<tr>
<td>11. bar</td>
<td>55. depression</td>
</tr>
<tr>
<td>12. barrel</td>
<td>56. dial</td>
</tr>
<tr>
<td>13. base</td>
<td>57. diaphragm</td>
</tr>
<tr>
<td>14. basket</td>
<td>58. dish</td>
</tr>
<tr>
<td>15. bead</td>
<td>59. disk</td>
</tr>
<tr>
<td>16. beam</td>
<td>60. donut</td>
</tr>
<tr>
<td>17. bevel</td>
<td>61. doughnut</td>
</tr>
<tr>
<td>18. bore</td>
<td>62. dowel</td>
</tr>
<tr>
<td>19. boss</td>
<td>63. drift-pin</td>
</tr>
<tr>
<td>20. broach</td>
<td>64. drill</td>
</tr>
<tr>
<td>21. bubble</td>
<td>65. drop</td>
</tr>
<tr>
<td>22. bucket</td>
<td>66. droplet</td>
</tr>
<tr>
<td>23. bulb</td>
<td>67. drum</td>
</tr>
<tr>
<td>24. bulge</td>
<td>68. dune</td>
</tr>
<tr>
<td>25. bullet</td>
<td>69. ellipse</td>
</tr>
<tr>
<td>26. bump</td>
<td>70. ellipsoid</td>
</tr>
<tr>
<td>27. bushing</td>
<td>71. filling</td>
</tr>
<tr>
<td>28. button</td>
<td>72. filter</td>
</tr>
<tr>
<td>29. cable</td>
<td>73. flake</td>
</tr>
<tr>
<td>30. can</td>
<td>74. flange</td>
</tr>
<tr>
<td>31. cap</td>
<td>75. flue</td>
</tr>
<tr>
<td>32. capsule</td>
<td>76. gasket</td>
</tr>
<tr>
<td>33. cartridge</td>
<td>77. gland</td>
</tr>
<tr>
<td>34. cave</td>
<td>78. globe</td>
</tr>
<tr>
<td>35. chimney</td>
<td>79. globule</td>
</tr>
<tr>
<td>36. cleat</td>
<td>80. grain</td>
</tr>
<tr>
<td>37. clip</td>
<td>81. hatch</td>
</tr>
<tr>
<td>38. coil</td>
<td>82. helix</td>
</tr>
<tr>
<td>39. coin</td>
<td>83. hole</td>
</tr>
<tr>
<td>40. collar</td>
<td>84. hump</td>
</tr>
<tr>
<td>41. column</td>
<td>85. journal</td>
</tr>
<tr>
<td>42. cone</td>
<td>86. knob</td>
</tr>
<tr>
<td>43. conic</td>
<td>87. ledge</td>
</tr>
<tr>
<td>44. conical</td>
<td>88. leg</td>
</tr>
<tr>
<td>101. nipple</td>
<td>99. node</td>
</tr>
<tr>
<td>102. o-ring</td>
<td>101. node</td>
</tr>
<tr>
<td>103. orifice</td>
<td>102. o-ring</td>
</tr>
<tr>
<td>104. oval</td>
<td>103. orifice</td>
</tr>
<tr>
<td>105. packing</td>
<td>104. oval</td>
</tr>
<tr>
<td>106. pad</td>
<td>105. packing</td>
</tr>
<tr>
<td>107. padding</td>
<td>106. pad</td>
</tr>
<tr>
<td>108. pail</td>
<td>107. padding</td>
</tr>
<tr>
<td>109. pallet</td>
<td>108. pail</td>
</tr>
<tr>
<td>110. pan</td>
<td>109. pallet</td>
</tr>
<tr>
<td>111. peak</td>
<td>110. pan</td>
</tr>
<tr>
<td>112. peen</td>
<td>111. peak</td>
</tr>
<tr>
<td>113. peg</td>
<td>112. peen</td>
</tr>
<tr>
<td>114. pellet</td>
<td>113. peg</td>
</tr>
<tr>
<td>115. perimeter</td>
<td>114. pellet</td>
</tr>
<tr>
<td>116. pillar</td>
<td>115. perimeter</td>
</tr>
<tr>
<td>117. pin</td>
<td>116. pillar</td>
</tr>
<tr>
<td>118. pipe</td>
<td>117. pin</td>
</tr>
<tr>
<td>119. plate</td>
<td>118. pipe</td>
</tr>
<tr>
<td>120. plug</td>
<td>119. plate</td>
</tr>
<tr>
<td>121. plunger</td>
<td>120. plug</td>
</tr>
<tr>
<td>122. pocket</td>
<td>121. plunger</td>
</tr>
<tr>
<td>123. pod</td>
<td>122. pocket</td>
</tr>
<tr>
<td>124. point</td>
<td>123. pod</td>
</tr>
<tr>
<td>125. pot</td>
<td>124. point</td>
</tr>
<tr>
<td>126. probe</td>
<td>125. pot</td>
</tr>
<tr>
<td>127. projection</td>
<td>126. probe</td>
</tr>
<tr>
<td>128. prominence</td>
<td>127. projection</td>
</tr>
<tr>
<td>129. protrusion</td>
<td>128. prominence</td>
</tr>
<tr>
<td>130. protuberance</td>
<td>129. protrusion</td>
</tr>
<tr>
<td>131. race</td>
<td>130. protuberance</td>
</tr>
<tr>
<td>132. rail</td>
<td>131. race</td>
</tr>
<tr>
<td>133. ream</td>
<td>132. rail</td>
</tr>
<tr>
<td>134. receptacle</td>
<td>133. ream</td>
</tr>
<tr>
<td>135. rifle</td>
<td>134. receptacle</td>
</tr>
<tr>
<td>136. rim</td>
<td>135. rifle</td>
</tr>
<tr>
<td>137. ring</td>
<td>136. rim</td>
</tr>
<tr>
<td>138. rod</td>
<td>137. ring</td>
</tr>
<tr>
<td>139. roller</td>
<td>138. rod</td>
</tr>
<tr>
<td>140. rope</td>
<td>139. roller</td>
</tr>
<tr>
<td>141. round</td>
<td>140. rope</td>
</tr>
<tr>
<td>142. rounding</td>
<td>141. round</td>
</tr>
<tr>
<td>143. scoop</td>
<td>142. rounding</td>
</tr>
<tr>
<td>144. screen</td>
<td>143. scoop</td>
</tr>
<tr>
<td>145. screw</td>
<td>144. screen</td>
</tr>
<tr>
<td>146. seal</td>
<td>145. screw</td>
</tr>
<tr>
<td>147. shaft</td>
<td>146. seal</td>
</tr>
<tr>
<td>148. shell</td>
<td>147. shaft</td>
</tr>
<tr>
<td>149. shield</td>
<td>148. shell</td>
</tr>
<tr>
<td>150. shim</td>
<td>149. shield</td>
</tr>
<tr>
<td>151. sleevings</td>
<td>150. shim</td>
</tr>
<tr>
<td>152. sphere</td>
<td>151. sleevings</td>
</tr>
<tr>
<td>153. spindle</td>
<td>152. sphere</td>
</tr>
<tr>
<td>154. stick</td>
<td>153. spindle</td>
</tr>
<tr>
<td>155. stopper</td>
<td>154. stick</td>
</tr>
<tr>
<td>156. string</td>
<td>155. stopper</td>
</tr>
<tr>
<td>157. stuffing</td>
<td>156. string</td>
</tr>
<tr>
<td>158. swell</td>
<td>157. stuffing</td>
</tr>
<tr>
<td>159. swelling</td>
<td>158. swell</td>
</tr>
<tr>
<td>160. tack</td>
<td>159. swelling</td>
</tr>
<tr>
<td>161. taper</td>
<td>160. tack</td>
</tr>
<tr>
<td>162. tear</td>
<td>161. taper</td>
</tr>
<tr>
<td>163. torus</td>
<td>162. tear</td>
</tr>
<tr>
<td>164. tube</td>
<td>163. torus</td>
</tr>
<tr>
<td>165. tunnel</td>
<td>164. tube</td>
</tr>
<tr>
<td>166. twine</td>
<td>165. tunnel</td>
</tr>
<tr>
<td>167. vein</td>
<td>166. twine</td>
</tr>
<tr>
<td>168. vertex</td>
<td>167. vein</td>
</tr>
<tr>
<td>169. wadding</td>
<td>168. vertex</td>
</tr>
<tr>
<td>170. washer</td>
<td>169. wadding</td>
</tr>
<tr>
<td>171. well</td>
<td>170. washer</td>
</tr>
<tr>
<td>172. wheel</td>
<td>171. well</td>
</tr>
<tr>
<td>173. wire</td>
<td>172. wheel</td>
</tr>
</tbody>
</table>
Table 9. Feature Classification by Form Group - II.

**Form Group Angularity (Solid)**

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>abutment</td>
<td>22.</td>
<td>column</td>
<td>43.</td>
<td>leaf</td>
</tr>
<tr>
<td>2.</td>
<td>angle</td>
<td>23.</td>
<td>core</td>
<td>44.</td>
<td>leg</td>
</tr>
<tr>
<td>3.</td>
<td>apron</td>
<td>24.</td>
<td>corner</td>
<td>45.</td>
<td>mask</td>
</tr>
<tr>
<td>4.</td>
<td>baffle-plate</td>
<td>25.</td>
<td>cover</td>
<td>46.</td>
<td>membrane</td>
</tr>
<tr>
<td>5.</td>
<td>bar</td>
<td>26.</td>
<td>cube</td>
<td>47.</td>
<td>packing</td>
</tr>
<tr>
<td>6.</td>
<td>base</td>
<td>27.</td>
<td>diaphragm</td>
<td>48.</td>
<td>pad</td>
</tr>
<tr>
<td>7.</td>
<td>beam</td>
<td>28.</td>
<td>elbow</td>
<td>49.</td>
<td>padding</td>
</tr>
<tr>
<td>8.</td>
<td>bed</td>
<td>29.</td>
<td>filling</td>
<td>50.</td>
<td>pallet</td>
</tr>
<tr>
<td>9.</td>
<td>bedding</td>
<td>30.</td>
<td>film</td>
<td>51.</td>
<td>pan</td>
</tr>
<tr>
<td>10.</td>
<td>bench</td>
<td>31.</td>
<td>filter</td>
<td>52.</td>
<td>partition</td>
</tr>
<tr>
<td>11.</td>
<td>bend</td>
<td>32.</td>
<td>fin</td>
<td>53.</td>
<td>patch</td>
</tr>
<tr>
<td>12.</td>
<td>billet</td>
<td>33.</td>
<td>foil</td>
<td>54.</td>
<td>pellet</td>
</tr>
<tr>
<td>13.</td>
<td>block</td>
<td>34.</td>
<td>gusset</td>
<td>55.</td>
<td>plate</td>
</tr>
<tr>
<td>14.</td>
<td>boss</td>
<td>35.</td>
<td>key</td>
<td>56.</td>
<td>pocket</td>
</tr>
<tr>
<td>15.</td>
<td>canal</td>
<td>36.</td>
<td>keel</td>
<td>57.</td>
<td>rabbet</td>
</tr>
<tr>
<td>16.</td>
<td>cantilever</td>
<td>37.</td>
<td>keyseat</td>
<td>58.</td>
<td>rail</td>
</tr>
<tr>
<td>17.</td>
<td>cap</td>
<td>38.</td>
<td>rod</td>
<td>59.</td>
<td>receptacle</td>
</tr>
<tr>
<td>18.</td>
<td>cavity</td>
<td>39.</td>
<td>screen</td>
<td>60.</td>
<td>rib</td>
</tr>
<tr>
<td>19.</td>
<td>chamfer</td>
<td>40.</td>
<td>shank</td>
<td>61.</td>
<td>washer</td>
</tr>
<tr>
<td>20.</td>
<td>channel</td>
<td>41.</td>
<td>shield</td>
<td>62.</td>
<td>sheet</td>
</tr>
<tr>
<td>21.</td>
<td>cleat</td>
<td>42.</td>
<td>sheet</td>
<td>63.</td>
<td>seat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64.</td>
<td>slat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65.</td>
<td>sheet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66.</td>
<td>shield</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67.</td>
<td>shim</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68.</td>
<td>shoulder</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69.</td>
<td>siab</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70.</td>
<td>slot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71.</td>
<td>socket</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72.</td>
<td>strip</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73.</td>
<td>stud</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>74.</td>
<td>stuffing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75.</td>
<td>table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76.</td>
<td>thickening</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77.</td>
<td>tile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.</td>
<td>track</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79.</td>
<td>vane</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.</td>
<td>vein</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>81.</td>
<td>wadding</td>
</tr>
</tbody>
</table>

**Form Group Angularity (Hollow)**

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>cage</td>
<td>4.</td>
<td>cell</td>
<td>7.</td>
<td>compartment</td>
</tr>
<tr>
<td>2.</td>
<td>case</td>
<td>5.</td>
<td>chamber</td>
<td>8.</td>
<td>duct</td>
</tr>
<tr>
<td>3.</td>
<td>casing</td>
<td>6.</td>
<td>chassis</td>
<td>9.</td>
<td>frame</td>
</tr>
</tbody>
</table>

**Form Group Curvature**

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>arc</td>
<td>6.</td>
<td>bow</td>
<td>11.</td>
<td>crescent</td>
</tr>
<tr>
<td>2.</td>
<td>arch</td>
<td>7.</td>
<td>brace</td>
<td>12.</td>
<td>cup</td>
</tr>
<tr>
<td>3.</td>
<td>arm</td>
<td>8.</td>
<td>bracket</td>
<td>13.</td>
<td>curve</td>
</tr>
<tr>
<td>4.</td>
<td>arris</td>
<td>9.</td>
<td>corner</td>
<td>14.</td>
<td>dome</td>
</tr>
<tr>
<td>5.</td>
<td>bend</td>
<td>10.</td>
<td>crater</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Form Group Straightness**

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
<th>No.</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>chord</td>
<td>3.</td>
<td>guide</td>
<td>5.</td>
<td>rectilinear</td>
</tr>
<tr>
<td>2.</td>
<td>edge</td>
<td>4.</td>
<td>line</td>
<td>6.</td>
<td>seam</td>
</tr>
</tbody>
</table>
Table 10. Feature Classification by Form Group - III.

**Form Group Circularity (Simple)**

<table>
<thead>
<tr>
<th>Form Group</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>arbor</td>
</tr>
<tr>
<td>2.</td>
<td>axle</td>
</tr>
<tr>
<td>3.</td>
<td>ball</td>
</tr>
<tr>
<td>4.</td>
<td>band</td>
</tr>
<tr>
<td>5.</td>
<td>bar</td>
</tr>
<tr>
<td>6.</td>
<td>barrel</td>
</tr>
<tr>
<td>7.</td>
<td>basket</td>
</tr>
<tr>
<td>8.</td>
<td>bead</td>
</tr>
<tr>
<td>9.</td>
<td>beam</td>
</tr>
<tr>
<td>10.</td>
<td>belt</td>
</tr>
<tr>
<td>11.</td>
<td>bevel</td>
</tr>
<tr>
<td>12.</td>
<td>bore</td>
</tr>
<tr>
<td>13.</td>
<td>boss</td>
</tr>
<tr>
<td>14.</td>
<td>bubble</td>
</tr>
<tr>
<td>15.</td>
<td>bulb</td>
</tr>
<tr>
<td>16.</td>
<td>bullet</td>
</tr>
<tr>
<td>17.</td>
<td>bushing</td>
</tr>
<tr>
<td>18.</td>
<td>button</td>
</tr>
<tr>
<td>19.</td>
<td>cable</td>
</tr>
<tr>
<td>20.</td>
<td>can</td>
</tr>
<tr>
<td>21.</td>
<td>cap</td>
</tr>
<tr>
<td>22.</td>
<td>capsule</td>
</tr>
<tr>
<td>23.</td>
<td>cartridge</td>
</tr>
<tr>
<td>24.</td>
<td>cleat</td>
</tr>
<tr>
<td>25.</td>
<td>clip</td>
</tr>
<tr>
<td>26.</td>
<td>coin</td>
</tr>
<tr>
<td>27.</td>
<td>collar</td>
</tr>
<tr>
<td>28.</td>
<td>column</td>
</tr>
<tr>
<td>29.</td>
<td>cone</td>
</tr>
<tr>
<td>30.</td>
<td>conic</td>
</tr>
<tr>
<td>31.</td>
<td>conical</td>
</tr>
<tr>
<td>32.</td>
<td>cord</td>
</tr>
<tr>
<td>33.</td>
<td>core</td>
</tr>
<tr>
<td>34.</td>
<td>cover</td>
</tr>
<tr>
<td>35.</td>
<td>crown</td>
</tr>
<tr>
<td>36.</td>
<td>cylinder</td>
</tr>
<tr>
<td>37.</td>
<td>dial</td>
</tr>
<tr>
<td>38.</td>
<td>diaphragm</td>
</tr>
<tr>
<td>39.</td>
<td>dish</td>
</tr>
<tr>
<td>40.</td>
<td>disk</td>
</tr>
<tr>
<td>41.</td>
<td>donut</td>
</tr>
<tr>
<td>42.</td>
<td>doughnut</td>
</tr>
<tr>
<td>43.</td>
<td>dowel</td>
</tr>
<tr>
<td>44.</td>
<td>drill</td>
</tr>
<tr>
<td>45.</td>
<td>drop</td>
</tr>
<tr>
<td>46.</td>
<td>droplet</td>
</tr>
<tr>
<td>47.</td>
<td>drum</td>
</tr>
<tr>
<td>48.</td>
<td>dune</td>
</tr>
<tr>
<td>49.</td>
<td>ellipse</td>
</tr>
<tr>
<td>50.</td>
<td>ellipsoid</td>
</tr>
<tr>
<td>51.</td>
<td>fillet</td>
</tr>
<tr>
<td>52.</td>
<td>filling</td>
</tr>
<tr>
<td>53.</td>
<td>filter</td>
</tr>
<tr>
<td>54.</td>
<td>flange</td>
</tr>
<tr>
<td>55.</td>
<td>flue</td>
</tr>
<tr>
<td>56.</td>
<td>gasket</td>
</tr>
<tr>
<td>57.</td>
<td>gland</td>
</tr>
<tr>
<td>58.</td>
<td>globe</td>
</tr>
<tr>
<td>59.</td>
<td>globule</td>
</tr>
<tr>
<td>60.</td>
<td>grain</td>
</tr>
<tr>
<td>61.</td>
<td>hole</td>
</tr>
<tr>
<td>62.</td>
<td>hub</td>
</tr>
<tr>
<td>63.</td>
<td>journal</td>
</tr>
<tr>
<td>64.</td>
<td>leg</td>
</tr>
<tr>
<td>65.</td>
<td>lip</td>
</tr>
<tr>
<td>66.</td>
<td>loop</td>
</tr>
<tr>
<td>67.</td>
<td>mandrel</td>
</tr>
<tr>
<td>68.</td>
<td>manifold</td>
</tr>
<tr>
<td>69.</td>
<td>mass</td>
</tr>
<tr>
<td>70.</td>
<td>membrane</td>
</tr>
<tr>
<td>71.</td>
<td>messenger</td>
</tr>
<tr>
<td>72.</td>
<td>nail</td>
</tr>
<tr>
<td>73.</td>
<td>needle</td>
</tr>
<tr>
<td>74.</td>
<td>o-ring</td>
</tr>
<tr>
<td>75.</td>
<td>oval</td>
</tr>
<tr>
<td>76.</td>
<td>packing</td>
</tr>
<tr>
<td>77.</td>
<td>pad</td>
</tr>
<tr>
<td>78.</td>
<td>padding</td>
</tr>
<tr>
<td>79.</td>
<td>pallet</td>
</tr>
<tr>
<td>80.</td>
<td>pan</td>
</tr>
<tr>
<td>81.</td>
<td>pel</td>
</tr>
<tr>
<td>82.</td>
<td>peg</td>
</tr>
<tr>
<td>83.</td>
<td>pellet</td>
</tr>
<tr>
<td>84.</td>
<td>perimeter</td>
</tr>
<tr>
<td>85.</td>
<td>pillar</td>
</tr>
<tr>
<td>86.</td>
<td>pin</td>
</tr>
<tr>
<td>87.</td>
<td>pipe</td>
</tr>
<tr>
<td>88.</td>
<td>plug</td>
</tr>
<tr>
<td>89.</td>
<td>plunger</td>
</tr>
<tr>
<td>90.</td>
<td>pocket</td>
</tr>
<tr>
<td>91.</td>
<td>pod</td>
</tr>
<tr>
<td>92.</td>
<td>pot</td>
</tr>
<tr>
<td>93.</td>
<td>probe</td>
</tr>
<tr>
<td>94.</td>
<td>rail</td>
</tr>
<tr>
<td>95.</td>
<td>receptacle</td>
</tr>
<tr>
<td>96.</td>
<td>ribbon</td>
</tr>
<tr>
<td>97.</td>
<td>rim</td>
</tr>
<tr>
<td>98.</td>
<td>ring</td>
</tr>
<tr>
<td>99.</td>
<td>rod</td>
</tr>
<tr>
<td>100.</td>
<td>roller</td>
</tr>
<tr>
<td>101.</td>
<td>rope</td>
</tr>
<tr>
<td>102.</td>
<td>round</td>
</tr>
<tr>
<td>103.</td>
<td>rounding</td>
</tr>
<tr>
<td>104.</td>
<td>scoop</td>
</tr>
<tr>
<td>105.</td>
<td>screen</td>
</tr>
<tr>
<td>106.</td>
<td>seal</td>
</tr>
<tr>
<td>107.</td>
<td>shaft</td>
</tr>
<tr>
<td>108.</td>
<td>shell</td>
</tr>
<tr>
<td>109.</td>
<td>shield</td>
</tr>
<tr>
<td>110.</td>
<td>sleeving</td>
</tr>
<tr>
<td>111.</td>
<td>sphere</td>
</tr>
<tr>
<td>112.</td>
<td>spindle</td>
</tr>
<tr>
<td>113.</td>
<td>stick</td>
</tr>
<tr>
<td>114.</td>
<td>stopper</td>
</tr>
<tr>
<td>115.</td>
<td>string</td>
</tr>
<tr>
<td>116.</td>
<td>swell</td>
</tr>
<tr>
<td>117.</td>
<td>swelling</td>
</tr>
<tr>
<td>118.</td>
<td>tack</td>
</tr>
<tr>
<td>119.</td>
<td>swelling</td>
</tr>
<tr>
<td>120.</td>
<td>taper</td>
</tr>
<tr>
<td>121.</td>
<td>tear</td>
</tr>
<tr>
<td>122.</td>
<td>tube</td>
</tr>
<tr>
<td>123.</td>
<td>tunnel</td>
</tr>
<tr>
<td>124.</td>
<td>twine</td>
</tr>
<tr>
<td>125.</td>
<td>vein</td>
</tr>
<tr>
<td>126.</td>
<td>wadding</td>
</tr>
<tr>
<td>127.</td>
<td>washer</td>
</tr>
<tr>
<td>128.</td>
<td>well</td>
</tr>
<tr>
<td>129.</td>
<td>wheel</td>
</tr>
<tr>
<td>130.</td>
<td>wire</td>
</tr>
</tbody>
</table>

**Form Group Circularity (Complex)**

<table>
<thead>
<tr>
<th>Form Group</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>aiguille</td>
</tr>
<tr>
<td>2.</td>
<td>chase</td>
</tr>
<tr>
<td>3.</td>
<td>chased-groove</td>
</tr>
<tr>
<td>4.</td>
<td>chute</td>
</tr>
<tr>
<td>5.</td>
<td>coil</td>
</tr>
<tr>
<td>6.</td>
<td>curl</td>
</tr>
<tr>
<td>7.</td>
<td>helix</td>
</tr>
<tr>
<td>8.</td>
<td>rifle</td>
</tr>
<tr>
<td>9.</td>
<td>screw</td>
</tr>
<tr>
<td>10.</td>
<td>spiral</td>
</tr>
<tr>
<td>11.</td>
<td>spring</td>
</tr>
<tr>
<td>12.</td>
<td>tapped-hole</td>
</tr>
<tr>
<td>13.</td>
<td>thread</td>
</tr>
<tr>
<td>14.</td>
<td>twist</td>
</tr>
</tbody>
</table>

A Feature Dictionary and Library 45
Table 11. Feature Classification by Form Group - IV.

Form Group Convexity

<table>
<thead>
<tr>
<th>Form Group</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convexity</td>
<td>arbor</td>
</tr>
<tr>
<td></td>
<td>arch</td>
</tr>
<tr>
<td></td>
<td>astragal</td>
</tr>
<tr>
<td></td>
<td>bead</td>
</tr>
<tr>
<td></td>
<td>boss</td>
</tr>
<tr>
<td></td>
<td>bulb</td>
</tr>
<tr>
<td></td>
<td>bulge</td>
</tr>
<tr>
<td></td>
<td>bump</td>
</tr>
<tr>
<td></td>
<td>button</td>
</tr>
<tr>
<td></td>
<td>cap</td>
</tr>
<tr>
<td></td>
<td>chimney</td>
</tr>
<tr>
<td></td>
<td>convex</td>
</tr>
<tr>
<td></td>
<td>crown</td>
</tr>
<tr>
<td></td>
<td>cup</td>
</tr>
<tr>
<td></td>
<td>cup</td>
</tr>
<tr>
<td></td>
<td>cylinder</td>
</tr>
<tr>
<td></td>
<td>dilation</td>
</tr>
<tr>
<td></td>
<td>dome</td>
</tr>
<tr>
<td></td>
<td>drop</td>
</tr>
<tr>
<td></td>
<td>droplet</td>
</tr>
<tr>
<td></td>
<td>emboss</td>
</tr>
<tr>
<td></td>
<td>entasis</td>
</tr>
<tr>
<td></td>
<td>globe</td>
</tr>
<tr>
<td></td>
<td>globe</td>
</tr>
<tr>
<td></td>
<td>hemisphere</td>
</tr>
<tr>
<td></td>
<td>hump</td>
</tr>
<tr>
<td></td>
<td>knob</td>
</tr>
<tr>
<td></td>
<td>lobe</td>
</tr>
<tr>
<td></td>
<td>mass</td>
</tr>
<tr>
<td></td>
<td>prominence</td>
</tr>
<tr>
<td></td>
<td>projection</td>
</tr>
<tr>
<td></td>
<td>prominence</td>
</tr>
<tr>
<td></td>
<td>protrusion</td>
</tr>
<tr>
<td></td>
<td>protuberance</td>
</tr>
<tr>
<td></td>
<td>raise</td>
</tr>
<tr>
<td></td>
<td>round</td>
</tr>
<tr>
<td></td>
<td>rounding</td>
</tr>
<tr>
<td></td>
<td>swell</td>
</tr>
<tr>
<td></td>
<td>swelling</td>
</tr>
<tr>
<td></td>
<td>tear</td>
</tr>
</tbody>
</table>

Form Group Concavity

<table>
<thead>
<tr>
<th>Form Group</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concavity</td>
<td>cap</td>
</tr>
<tr>
<td></td>
<td>cave</td>
</tr>
<tr>
<td></td>
<td>cavity</td>
</tr>
<tr>
<td></td>
<td>channel</td>
</tr>
<tr>
<td></td>
<td>constriction</td>
</tr>
<tr>
<td></td>
<td>crater</td>
</tr>
<tr>
<td></td>
<td>cup</td>
</tr>
<tr>
<td></td>
<td>cup</td>
</tr>
<tr>
<td></td>
<td>deflation</td>
</tr>
<tr>
<td></td>
<td>dent</td>
</tr>
<tr>
<td></td>
<td>depression</td>
</tr>
<tr>
<td></td>
<td>dip</td>
</tr>
<tr>
<td></td>
<td>dome</td>
</tr>
<tr>
<td></td>
<td>gutter</td>
</tr>
<tr>
<td></td>
<td>hemisphere</td>
</tr>
<tr>
<td></td>
<td>hollow</td>
</tr>
<tr>
<td></td>
<td>impression</td>
</tr>
<tr>
<td></td>
<td>ladle</td>
</tr>
<tr>
<td></td>
<td>pit</td>
</tr>
<tr>
<td></td>
<td>pulley</td>
</tr>
<tr>
<td></td>
<td>race</td>
</tr>
<tr>
<td></td>
<td>reel</td>
</tr>
<tr>
<td></td>
<td>scoop</td>
</tr>
<tr>
<td></td>
<td>spool</td>
</tr>
<tr>
<td></td>
<td>tunnel</td>
</tr>
</tbody>
</table>

Form Group Flatness

<table>
<thead>
<tr>
<th>Form Group</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flatness</td>
<td>abutment</td>
</tr>
<tr>
<td></td>
<td>apron</td>
</tr>
<tr>
<td></td>
<td>baffle-plate</td>
</tr>
<tr>
<td></td>
<td>band</td>
</tr>
<tr>
<td></td>
<td>bar</td>
</tr>
<tr>
<td></td>
<td>base</td>
</tr>
<tr>
<td></td>
<td>beam</td>
</tr>
<tr>
<td></td>
<td>bed</td>
</tr>
<tr>
<td></td>
<td>bedding</td>
</tr>
<tr>
<td></td>
<td>belt</td>
</tr>
<tr>
<td></td>
<td>bench</td>
</tr>
<tr>
<td></td>
<td>billet</td>
</tr>
<tr>
<td></td>
<td>block</td>
</tr>
<tr>
<td></td>
<td>box</td>
</tr>
<tr>
<td></td>
<td>cantilever</td>
</tr>
<tr>
<td></td>
<td>case</td>
</tr>
<tr>
<td></td>
<td>casing</td>
</tr>
<tr>
<td></td>
<td>coin</td>
</tr>
<tr>
<td></td>
<td>column</td>
</tr>
<tr>
<td></td>
<td>column</td>
</tr>
<tr>
<td></td>
<td>cover</td>
</tr>
<tr>
<td></td>
<td>cube</td>
</tr>
<tr>
<td></td>
<td>face</td>
</tr>
<tr>
<td></td>
<td>fence</td>
</tr>
<tr>
<td></td>
<td>filling</td>
</tr>
<tr>
<td></td>
<td>film</td>
</tr>
<tr>
<td></td>
<td>filter</td>
</tr>
<tr>
<td></td>
<td>fin</td>
</tr>
<tr>
<td></td>
<td>flume</td>
</tr>
<tr>
<td></td>
<td>foil</td>
</tr>
<tr>
<td></td>
<td>gauze</td>
</tr>
<tr>
<td></td>
<td>gib</td>
</tr>
<tr>
<td></td>
<td>grid</td>
</tr>
<tr>
<td></td>
<td>grill</td>
</tr>
<tr>
<td></td>
<td>gusset</td>
</tr>
<tr>
<td></td>
<td>keel</td>
</tr>
<tr>
<td></td>
<td>key</td>
</tr>
<tr>
<td></td>
<td>leaf</td>
</tr>
<tr>
<td></td>
<td>leg</td>
</tr>
<tr>
<td></td>
<td>mask</td>
</tr>
<tr>
<td></td>
<td>mat</td>
</tr>
<tr>
<td></td>
<td>membrane</td>
</tr>
<tr>
<td></td>
<td>mesh</td>
</tr>
<tr>
<td></td>
<td>mosaic</td>
</tr>
<tr>
<td></td>
<td>nest</td>
</tr>
<tr>
<td></td>
<td>net</td>
</tr>
<tr>
<td></td>
<td>packing</td>
</tr>
<tr>
<td></td>
<td>pad</td>
</tr>
<tr>
<td></td>
<td>padding</td>
</tr>
<tr>
<td></td>
<td>pallet</td>
</tr>
<tr>
<td></td>
<td>pan</td>
</tr>
<tr>
<td></td>
<td>partition</td>
</tr>
<tr>
<td></td>
<td>patch</td>
</tr>
<tr>
<td></td>
<td>pattern</td>
</tr>
<tr>
<td></td>
<td>pellet</td>
</tr>
<tr>
<td></td>
<td>plane</td>
</tr>
<tr>
<td></td>
<td>plate</td>
</tr>
<tr>
<td></td>
<td>rail</td>
</tr>
<tr>
<td></td>
<td>ribbon</td>
</tr>
<tr>
<td></td>
<td>screen</td>
</tr>
<tr>
<td></td>
<td>seat</td>
</tr>
<tr>
<td></td>
<td>sheet</td>
</tr>
<tr>
<td></td>
<td>shield</td>
</tr>
<tr>
<td></td>
<td>shim</td>
</tr>
<tr>
<td></td>
<td>sieve</td>
</tr>
<tr>
<td></td>
<td>slab</td>
</tr>
<tr>
<td></td>
<td>strip</td>
</tr>
<tr>
<td></td>
<td>stud</td>
</tr>
<tr>
<td></td>
<td>stuffing</td>
</tr>
<tr>
<td></td>
<td>table</td>
</tr>
<tr>
<td></td>
<td>tank</td>
</tr>
<tr>
<td></td>
<td>thickening</td>
</tr>
<tr>
<td></td>
<td>tile</td>
</tr>
<tr>
<td></td>
<td>track</td>
</tr>
<tr>
<td></td>
<td>wadding</td>
</tr>
<tr>
<td></td>
<td>washer</td>
</tr>
<tr>
<td></td>
<td>web</td>
</tr>
</tbody>
</table>
Table 12. Feature Classification by Form Group - V.

**Form Group Sharpness**

1. apex  
2. cone  
3. conic  
4. conical  
5. corner  
6. cotter  
7. dowel  
8. drift-pin  
9. nail  
10. needle  
11. nile  
12. peak  
13. peg  
14. pin  
15. point  
16. screw  
17. sharp  
18. spike  
19. tack  
20. taper  
21. vertex

**Form Group Bluntness**

1. blunt  
2. edge  
3. round

**Form Group Smoothness**

1. evenness  
2. plane  
3. polished  
4. roller

**Form Group Roughness**

1. bump  
2. dilation  
3. grain  
4. knurl  
5. pod

**Form Group Notch**

1. deflation  
2. dent  
3. depression  
4. dip  
5. notch  
6. valley

**Form Group Fold**

1. fold
Table 13. Feature Classification by Form Group - VI.

**Form Group Furrow**

1. break  
2. canal  
3. cavity  
4. channel  
5. cleavage  
6. cleft  
7. clough  
8. collet  
9. crack  
10. cut  
11. deep  
12. depth  
13. ditch  
14. fissure  
15. flute  
16. fracture  
17. furrow  
18. gap  
19. gorge  
20. groove  
21. gutter  
22. hole  
23. kerf  
24. keyhole  
25. keyseat  
26. keyway  
27. parting  
28. pocket  
29. pouch  
30. rabbet  
31. race  
32. ravine  
33. receptacle  
34. recess  
35. ridge  
36. rupture  
37. rut  
38. scar  
39. separation  
40. slit  
41. slot  
42. socket  
43. split  
44. valley

**Form Group Opening**

1. annular-hole  
2. aperture  
3. arbor-hole  
4. bore  
5. counter-bore  
6. crevice  
7. flange  
8. hole  
9. inlet  
10. nozzle  
11. opening  
12. orifice  
13. perforation  
14. puncture  
15. tapped-hole

**Form Group Closure**

1. baffle-plate  
2. cap  
3. cover  
4. hatch  
5. hem  
6. plug  
7. seal

**Form Group Perforator**

1. bore  
2. broach  
3. counter-bore  
4. ream

**Form Group Stopper**

1. bushing  
2. clip  
3. gasket  
4. stopper
2.4 Rules and Guidelines for Feature Definition

Although features under the same topology group are modeled using the same set of parameters, their physical nature and purpose govern the values that the parameters may take. In general, different cognition rules govern the size, placement and use of each feature. This puts constraints in the way that features can be defined. Many times, design standards place rules or guidelines on a feature. The guidelines state what conditions should or must be imposed on the feature's topology and geometry in order for it to perform its required function. The rules often place quantitative or qualitative restrictions on the parameters that describe the feature. Any program that has to impart intelligence to the design process must have a representation that includes knowledge of feature defining parameters.

2.4.1 Mathematical Relations Between Parameters

Often, similar features impart different characteristics to the object being modeled and perform different functions. Each topology class will effectively represent only one feature unless a scheme to differentiate feature members is evolved. A conceptual study of features demonstrates that such differentiation can be established by creation of mathematical rules which govern the range of parameter values, relationships between various parameters and consist of equality and inequality relations between the parameters. These relations should be formulated taking design and other practical considerations into account. For example load conditions will constrain a 'column' to have a minimum
diameter-to-length ratio, or a ‘washer’ is expected to have a certain
thickness-to-diameter ratio. Since dimensions influence many aspects of design, it
is important that these sets of rules reflect the designer’s perceptions of the feature’s
proportions. Once this differentiation is in place, a topology group can represent
multiple features and can advise a designer about the numerical values of input
parameters required to create a particular feature. Application of these rules will
constrain feature definition within the specified ranges and ensure that it meets the
design rules, guidelines and standards. A feature can be created only if it meets the
inequality constraints. Thus a designer who is unaware of a feature’s proportion is
guided by these rules in creating a reasonable feature.

Prototype mathematical inequality relations for example features from different
groups were created. These relations differentiate features within the same topology
form group. Obviously, synonym features will be governed by the same set of rules.
The relations put constraints on the relative values of parameters and may also
control the values of individual parameters. For example, Topology group A is
defined by two parameters: radius (R) and length (L). The relations put constraints
on the values of the ratio R/L. In the case of Topology E the defining parameters are
length (L), breadths \((B_1, B_2)\), and height (H). The relations govern the ratios \(B_2/B_1\),
\(B_1/L\) and \(H/L\). Tables 14 through 16 show some suggested mathematical relations for
example features. Relations for other topology and form groups can be formed in a
similar manner. Figures 15 and 16 show some extreme cases of features that were
created using these relations.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Shape Form</th>
<th>Topology</th>
<th>Suggested Mathematical Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar</td>
<td>Rotundity</td>
<td>A</td>
<td>$0.1 \leq \frac{R}{L} \leq 0.5$</td>
</tr>
<tr>
<td>Boss</td>
<td>Rotundity</td>
<td>A</td>
<td>$0.5 \leq \frac{R}{L} \leq 5.0$</td>
</tr>
<tr>
<td>Rim</td>
<td>Rotundity</td>
<td>C</td>
<td>$0.25 \leq \frac{R_1}{R_2} &lt; 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1.0 \leq \frac{R_2}{L} \leq 2.0$</td>
</tr>
<tr>
<td>Hub</td>
<td>Rotundity</td>
<td>V</td>
<td>$0.5 \leq \frac{R_2}{R_1} \leq 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1.0 \leq \frac{R_1}{L} \leq 2.0$</td>
</tr>
<tr>
<td>Bevel</td>
<td>Rotundity</td>
<td>V</td>
<td>$0.5 \leq \frac{R_2}{R_1} \leq 0.75$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$2.0 \leq \frac{R_1}{L} \leq 5.0$</td>
</tr>
<tr>
<td>Bar</td>
<td>Angularity</td>
<td>E</td>
<td>$B_2/B_1 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.2 \leq \frac{B_2}{B_1} &lt; 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.2 \leq \frac{H}{L} &lt; 1.0$</td>
</tr>
<tr>
<td>Boss</td>
<td>Angularity</td>
<td>E</td>
<td>$B_2/B_1 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1.0 &lt; \frac{B_2}{L} \leq 10.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$1.0 &lt; \frac{H}{L} \leq 10.0$</td>
</tr>
<tr>
<td>Case</td>
<td>Angularity</td>
<td>G</td>
<td>$0.5 \leq \frac{B}{L} \leq 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.5 \leq \frac{H}{L} \leq 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.1 \leq \frac{T}{L} \leq 0.25$</td>
</tr>
<tr>
<td>Keyway</td>
<td>Angularity</td>
<td>F</td>
<td>$B_2/B_1 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.1 \leq \frac{B_2}{B_1} \leq 0.5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.1 \leq \frac{H}{L} \leq 0.5$</td>
</tr>
<tr>
<td>Packing</td>
<td>Angularity</td>
<td>E</td>
<td>$B_2/B_1 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$20.0 \leq \frac{B_2}{L} \leq 40.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$20.0 \leq \frac{H}{L} \leq 10.0$</td>
</tr>
<tr>
<td>Plate</td>
<td>Angularity</td>
<td>E</td>
<td>$B_2/B_1 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$10.0 \leq \frac{B_2}{L} \leq 20.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$10.0 &lt; \frac{H}{L} \leq 20.0$</td>
</tr>
<tr>
<td>Shell</td>
<td>Angularity</td>
<td>G</td>
<td>$0.5 \leq \frac{B}{L} \leq 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.5 \leq \frac{H}{L} \leq 1.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$0.01 \leq \frac{T}{L} \leq 0.1$</td>
</tr>
</tbody>
</table>
Table 15. Suggested Mathematical Relations For Example Features - II.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Shape Form</th>
<th>Topology</th>
<th>Suggested Mathematical Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>Curvature</td>
<td>H</td>
<td>$10^\circ \leq A \leq 170^\circ$ \n$1.0 \leq R/T \leq 2.0$</td>
</tr>
<tr>
<td>Arch</td>
<td>Curvature</td>
<td>D</td>
<td>$0.75 \leq R_1/R_2 &lt; 1.0$ \n$1.0 \leq R_2/L \leq 2.0$</td>
</tr>
<tr>
<td>Dome</td>
<td>Curvature</td>
<td>O</td>
<td>$0.75 \leq R_1/R_2 &lt; 1.0$</td>
</tr>
<tr>
<td>Tunnel</td>
<td>Curvature</td>
<td>D</td>
<td>$0.75 \leq R_1/R_2 &lt; 1.0$ \n$0.01 \leq R_2/L \leq 1.0$</td>
</tr>
<tr>
<td>Coin</td>
<td>Circularity</td>
<td>A</td>
<td>$10.0 \leq R/L \leq 20.0$</td>
</tr>
<tr>
<td>Dowel</td>
<td>Circularity</td>
<td>A</td>
<td>$0.01 \leq R/L \leq 0.1$</td>
</tr>
<tr>
<td>Sleeving</td>
<td>Circularity</td>
<td>C</td>
<td>$0.75 \leq R_1/R_2 &lt; 1.0$ \n$0.1 \leq R_2/L \leq 1.0$</td>
</tr>
<tr>
<td>Wire</td>
<td>Circularity</td>
<td>A</td>
<td>$0.001 \leq R/L \leq 0.01$</td>
</tr>
<tr>
<td>Hemisphere</td>
<td>Convexity</td>
<td>O</td>
<td>$0.0 \leq R_1/R_2 &lt; 1.0$</td>
</tr>
<tr>
<td>Membrane</td>
<td>Flatness</td>
<td>E</td>
<td>$B_2/B_1 = 1.0$ \n$40.0 \leq B_1/L \leq 100.0$ \n$40.0 \leq H/L \leq 100.0$</td>
</tr>
<tr>
<td>Pad</td>
<td>Flatness</td>
<td>A</td>
<td>$10.0 \leq R/L \leq 20.0$</td>
</tr>
</tbody>
</table>
### Table 16. Suggested Mathematical Relations For Example Features - III.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Shape Form</th>
<th>Topology</th>
<th>Suggested Mathematical Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spike</td>
<td>Sharpness</td>
<td>K</td>
<td>( A.R. = 0.0 ) ( 0.1 \leq R/H \leq 0.2 )</td>
</tr>
<tr>
<td>Taper</td>
<td>Sharpness</td>
<td>K</td>
<td>( 0.0 &lt; A.R. &lt; 1.0 ) ( 0.5 \leq R/H \leq 1.0 )</td>
</tr>
<tr>
<td>Crack</td>
<td>Furrow</td>
<td>F</td>
<td>( 0.0 \leq B_2/B_1 \leq 0.01 ) ( 0.0 \leq B_1/L \leq 0.01 ) ( 0.0 \leq H/L \leq 0.01 )</td>
</tr>
<tr>
<td>Dip</td>
<td>Furrow</td>
<td>M</td>
<td>( 0.0 &lt; A.R. &lt; 1.0 ) ( 0.5 \leq R/H \leq 1.0 )</td>
</tr>
<tr>
<td>Flute</td>
<td>Furrow</td>
<td>N</td>
<td>( 0.05 \leq R/L \leq 0.25 )</td>
</tr>
<tr>
<td>Notch</td>
<td>Furrow</td>
<td>M</td>
<td>( A.R. = 0.0 ) ( 0.1 \leq R/H \leq 0.2 )</td>
</tr>
<tr>
<td>Pocket</td>
<td>Furrow</td>
<td>F</td>
<td>( 0.5 \leq B_2/B_1 \leq 1.0 ) ( 0.5 \leq B_1/L \leq 2.0 ) ( 0.1 \leq H/L \leq 0.5 )</td>
</tr>
<tr>
<td>Slot</td>
<td>Furrow</td>
<td>F</td>
<td>( B_2/B_1 = 1 ) ( 0.05 \leq B_1/L \leq 0.25 ) ( 0.025 \leq H/L \leq 0.25 )</td>
</tr>
<tr>
<td>Collar</td>
<td>Stopper</td>
<td>C</td>
<td>( 0.25 \leq R_1/R_2 \leq 0.75 ) ( 1.0 \leq R_2/L \leq 2.0 )</td>
</tr>
<tr>
<td>Washer</td>
<td>Stopper</td>
<td>C</td>
<td>( 0.25 \leq R_1/R_2 \leq 0.75 ) ( 10.0 \leq R_2/L \leq 20.0 )</td>
</tr>
</tbody>
</table>
Figure 15. Example Features Based On Suggested Guidelines - I.
Collar \( R_1 = 2.5, R_2 = 10.0, L = 10.0 \)

Collar \( R_1 = 7.5, R_2 = 10.0, L = 20.0 \)

Bar \( R = 10.0, L = 100.0 \)

Bar \( R = 10.0, L = 20.0 \)

Bar \( L = 100.0, B_1 = B_2 = 20.0, H = 20.0 \)

Keyway \( L = 60.0, B_1 = B_2 = 6.0, H = 3.0 \)

Pocket \( L = 60.0, B_1 = B_2 = 30.0, H = 6.0 \)

Figure 16. Example Features Based On Suggested Guidelines - II.
2.4.2 Shape-Altering Words or Adjectives

Use of shape-altering words or descriptions puts additional constraints on the defining relations. The above relations are used as guidelines to define a default feature. Many times designers need variations of these features and describe them by using additional words like 'long', 'thick' or 'large'. The use of such words put additional constraints on a feature's definition. Hence, the original relations are no longer valid and must be modified appropriately to match the new description of the feature being defined. Most words have a numerical effect on the constraints. A few words (e.g., smooth, rough) have a qualitative effect on the feature and do not affect the mathematical relations in any way. Still such words are part of a designer's vocabulary and must be accounted for. A set of shape-altering words and parameters has been defined to make such changes in the mathematical relations if required. These parameters, called 'adjectives', act as multiplicative factors to all or specified groups of parameters that define a feature. Because a descriptive word may induce different changes in the dimensions of different features, one must be careful when interpreting the meaning of such factors. This is best explained by means of an example. Consider the descriptive word 'thin' being applied to two different features, a circular 'rod' and a rectangular 'bar'. The radius-to-length ratio (R/L) will be affected in case of the rod but in case of the bar two ratios, B/L and H / L will be changed. Table 17 provides a listing of common shape-altering words and their suggested effect on the mathematical relations. Their effect on mathematical relations that govern feature-parameter values can be better understood by means of an example. In case of a rectangular 'bar', the word 'long' will affect all relations, but 'wide' will affect only that which concerns its breadth. The use of such
shape-altering words gives a designer additional control over the overall shape of the feature and also allows him to apply his normal vocabulary to feature definitions. It also allows him to create gradual changes in a feature’s dimensions by repeated use of different or the same shape-altering words. The designer can hence change the description of the feature till it meets his requirements. For example, if a designer is not satisfied with the dimension range for a ‘long rod’, he can use the word ‘long’ again to further modify the mathematical relations to obtain a ‘long long rod’. Normal vocabulary uses adverbs to further describe nouns. Thus provision should be made for the use of adverbs such as ‘very’ for additional descriptions. Thus a ‘long long rod’ can be described as a ‘very long rod’.

2.5 Feature Definition By Incomplete Specification of Parameters

In the early stages of conceptual design, a designer may only have a rough idea about the dimensions of a part or about the features that make up that part. To require him to enter all parameter values for defining a feature defeats the purpose of an intelligent system. However, the presence of mathematical rules and generic parameter values helps a designer to stay within reason and allow him to create a feature using the default values or by partial specification of parameters. The minimum number of parameters to be specified is governed by the number of independent rules. A designer may be required to specify some parameters and may be given an option of choosing the other parameters. Once has specified the
Table 17. Shape-Altering Words And Their Effect on Mathematical Rules.

<table>
<thead>
<tr>
<th>Shape Altering Word</th>
<th>Suggested Scale Factors</th>
<th>Relations Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>blunt</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>broad</td>
<td>1.25</td>
<td>B/L</td>
</tr>
<tr>
<td>bumpy</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>compressed</td>
<td>1.25</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>deep</td>
<td>1.50</td>
<td>H/L</td>
</tr>
<tr>
<td>default</td>
<td>1.00</td>
<td>N.A.</td>
</tr>
<tr>
<td>extended</td>
<td>0.80</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>fat</td>
<td>1.50</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>fine</td>
<td>0.67</td>
<td>B/L, R/L</td>
</tr>
<tr>
<td>glossy</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>high</td>
<td>1.50</td>
<td>H/L</td>
</tr>
<tr>
<td>jagged</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>long</td>
<td>0.67</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>low</td>
<td>0.67</td>
<td>H/L</td>
</tr>
<tr>
<td>narrow</td>
<td>0.80</td>
<td>R/L</td>
</tr>
<tr>
<td>normal</td>
<td>1.00</td>
<td>N.A.</td>
</tr>
<tr>
<td>plump</td>
<td>1.25</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>pointed</td>
<td>0.00</td>
<td>$R_j/R_{1j}B_{ij}/B_1$</td>
</tr>
<tr>
<td>polished</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>rough</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>shallow</td>
<td>0.67</td>
<td>H/L</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>R/L</td>
</tr>
<tr>
<td>sharp</td>
<td>0.00</td>
<td>$R_j/R_{1j}B_{ij}/B_1$</td>
</tr>
<tr>
<td>short</td>
<td>0.67</td>
<td>H/L</td>
</tr>
<tr>
<td></td>
<td>1.50</td>
<td>R/L</td>
</tr>
<tr>
<td>sleek</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>slender</td>
<td>0.67</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>slim</td>
<td>0.80</td>
<td>B/L, H/L, R/L</td>
</tr>
<tr>
<td>spiked</td>
<td>0.00</td>
<td>$R_j/R_{1j}B_{ij}/B_1$</td>
</tr>
<tr>
<td>tall</td>
<td>1.50</td>
<td>H/L</td>
</tr>
<tr>
<td></td>
<td>0.67</td>
<td>R/L</td>
</tr>
<tr>
<td>tapered</td>
<td>0.80</td>
<td>$R_j/R_{1j}B_{ij}/B_1$</td>
</tr>
<tr>
<td>thick</td>
<td>1.50</td>
<td>B/L, H/L, R/L, $R_j/R_1$</td>
</tr>
<tr>
<td>thin</td>
<td>0.67</td>
<td>B/L, H/L, R/L, $R_j/R_1$</td>
</tr>
<tr>
<td>wide</td>
<td>1.50</td>
<td>B/L</td>
</tr>
</tbody>
</table>
parameters of his choice, the remaining parameters in the set can be computed automatically by applying the rules. This procedure will help speed up the decision process for a designer.

In order to compute the complete set of parameters for a general feature, it is necessary to formulate a general method for parameter computations. A mathematical model that can be used for this purpose is presented below. This model can be used in conjunction with linear inequalities.

Consider a feature (F) that can be represented by the following function

$$ F : f \left( P_1, P_2, \ldots, P_n \right), \quad (2.1) $$

where $f$ is a function that depends on $n$ parameters. If the feature is governed by $m$ linear constraints ($l$) involving $k$ parameters, they can be represented as

$$ l_i : \sum_{j=1}^{n} C_{ij} P_j \quad i = 1, 2, 3, \ldots, m \quad (2.2) $$

where, $C_{ij}$ are coefficients generated from the mathematical rules. $C_{ij} \neq 0$ for $k$ parameters and $C_{ij} = 0$ for the other $n - k$. This results in a matrix equation that can be solved using a variety of methods. However, since the constraints use only $k$ parameters, the other $n - k$ parameters must be specified by the user. Let us first address the equality constraints only. Different cases are possible and must be addressed separately.

If $m = 0$, then there are no constraints and the user must specify all the parameters.
If $m \neq 0$, then two cases exist.
1. If \( k \leq m \), then \( m - k \) are dependent constraints and are redundant. The \( k \)
independent relations can be used to simultaneously compute unique solutions
for the \( k \) parameters.

2. If \( k > m \), then an additional \( k - m \) parameters must be specified. After they are
specified, the \( m \) independent relations can be used to simultaneously compute
unique solutions for the \( m \) parameters.

In general, the constraint set consists of equality and inequality relations. In
such cases the rules must be classified into two groups. If a constraint has a \( \geq \) or a
\( \leq \) relation, then it must be classified into both groups. The first group must consist
of equality relations and must be solved first using the procedure mentioned above.
The solutions obtained from this group must be replaced in the second group
consisting of inequality relations. Now, the second group may be considered.
Change all inequalities to \( > \) relations, by multiplying by \((-1)\) if necessary. Now this
set of relations can be solved by using the process of elimination to get unique values
or a range for the parameters. In the latter case, the designer may be given the
option of selecting a value, or a conservative estimate can be made automatically
using the mean value of the range.

In either case, once all parameters have been computed, they must be used to
check that all mathematical constraints are satisfied. If not, numerical methods may
be used to solve for the constraints using these values as an initial guess as is
illustrated by the two examples given below.

Consider a feature defined by three parameters \( x \), \( y \) and \( z \) with three constraints
\[
\begin{align*}
x - 2y + z &= 3 \\
3x + 4y - z &= 26
\end{align*}
\]
\[8x + 2y - 3z > 30\]

In this case the first two equations are put in one group. The equality group contains two equations in three unknowns; that is, \( m = 2 \) and \( k = 3 \). Hence, one parameter must be specified. Let \( x = 4 \) (chosen arbitrarily). Now the two constraints can be solved simultaneously to get \( y = 6.5 \) and \( z = 12 \). The computed values must now be used in the third constraint to check the validity of the solution. It turns out that \( 8x + 2y - 3z = 33 > 30 \) and hence meets the entire constraint set. Hence \( x = 4.0, y = 6.5, z = 12.0 \) is one of the many acceptable solutions for the feature.

As another example, consider the mathematical relations that govern the definition of a rectangular bar. The relations are:

\[ \frac{B_2}{B_1} = 1.0 \]

\[ 0.2 \leq \frac{B_1}{L} < 1.0 \]

\[ 0.2 \leq \frac{H}{L} < 1.0 \]

There are effectively three parameters that describe the feature but there are only two relations. Hence one dimension must be specified by the user. If the designer chooses the parameter \( L \) to be 100.0, then using the above method we get a range for \( B_1 \) and \( H \). Using the mean value of the ranges we get one acceptable solution to be \( B_1 = B_2 = 40.0 \) and \( H = 40.0 \). Some other acceptable solutions would be \( B_1 = B_2 = 25.0 \) and \( H = 50.0 \) or \( B_1 = B_2 = 60.0 \) and \( H = 30.0 \). The features that would be created using these dimensions are shown in Figure 17.
Figure 17. Some Acceptable Dimensions for a Rectangular Bar.
2.6 *A Feature Library*

The establishment of a feature library based on definition, classification, parameterization and mathematical constraints is now feasible. The library will contain the following information for each feature:

1. Feature name.
2. Feature definition.
3. Feature classification (Topology and Shape Form).
4. Feature synonyms.
5. Mathematical Inequalities or rules governing the feature.
6. A generic model listing default parameters defining the feature.

This library can be used as a tool by the designer to define and create feature-based models. Since this library is not dependent on a particular design environment, it can be used as an interface for any kind of design, modeling or programming environment. It can be used not only to check whether a feature definition meets the constraints, design rules and procedures but also to help a user understand the meaning of the feature, to give the user the choice of using other features which are its synonyms, to create generic models of features and to allow feature creation by incomplete specifications of parameters. Appendix A contains the complete feature dictionary that can be used in conjunction with an application program. Only those features that have been considered in the previous sections contain a set of mathematical relations as examples of feature modeling. As mentioned earlier, the
remaining mathematical relations can be established at a later stage, in the process of establishing a production feature modeler.
Chapter 3: Designer-Oriented Modeling Language

Although CAD/CAM systems automate and speed up the design process, they require a high level of human interaction to get input data. The ease with which such data can be input will control the speed of such interaction. Most systems use graphics input/output as means of interacting with the designer. Such menu-driven systems require the user to have an in-depth knowledge of the menu structure and hierarchy of the system. For efficient use of such systems, a user must learn how to use them. This has tremendous cost disadvantages since considerable time must be spent in training users. In addition, menu interfaces are less flexible and work only in their own database domains. They are more time consuming to customize and require applications programmers to anticipate the kind of questions that will be asked. A flexible designing system must be able to interact freely with other programming environments and any data transfer should be simple. Future systems will integrate novel methods of input like voice recognition and natural language so that input will become simpler and faster. In most cases however, such complex and costly interfaces are not required. A simple language that would allow a user to have linguistic coverage, deductive abilities or intelligent prompting would suffice. One
of the aims of this research is to develop a designer-oriented modeling language called Universal Modeling Language (UML), that will be fast and robust enough to serve users with no knowledge of the underlying domain. This language should be completely portable (i.e. can run on different hardware under different operating systems). The language is a preliminary step to implementing a voice input interface.

Using a natural language approach towards feature selection and modeling will make the task of the designer simpler and faster. In order to be able to devise a method that allows such an approach, one must take a close look at the manner in which a designer thinks and formulates his modeling procedure. In addition, one must consider the grammar of the language structure involved in this process. The elements that make up this structure depend on the kind of processing to be done and the action to be taken. One planned application of UML is its use in a modeling environment coupled to a graphics display. Hence the structure elements will have to handle modeling, system and graphics commands issued by the designer.

3.1 Language Structure Elements

For modeling an object using feature instances, a general language structure will have the form:

\[ \text{[verb]} \ [\text{adjective}] \ [\text{component}] \ [\text{name}] \ [\text{parameters}] \ [\text{location}] \ [\text{orientation}] \]

where the elements of the structure are as described below:
A command can be of different types. It could indicate a modeling, transformation, display, inquiry, archive or control verb. A modeling verb allows creation, modification, deletion of different components and the relative placement of components. These words will operate on a component specified by the user. A transformation verb executes viewing transformations (zoom, pan, rotations, etc.). An inquiry verb helps a user make numerous inquires about a model, component or a feature. Archive verbs allow execution of archiving commands like 'recall', 'start' or 'exit'. Thus, as in all languages a verb is the principal element in the entire structure. It helps decide the level of interpretation required and the flow of the program. The verb list and the classification scheme employed is shown in Table 18.

Use of shape altering would be restricted to features. It indicates what coefficients will be applicable in modifying mathematical constraints for defining that feature.

It could be 'feature', 'object' or 'assembly' and indicates the position of the component in the object hierarchy. In case of a modeling verb, it plays a major role in interpretation of the other elements of the structure.

Any component must be named in order to identify it. This allows the program to keep track of all components existing in the model. This simplifies commands for any further operations on the named component.

These are numerical values (to be specified in a fixed order) that allow definition of features. This list must be completely specified by the user. In case it is not, then mathematical relations defined for that feature are applied to compute the missing parameters. In some cases they may represent graphics parameters.

This represents the location of the component in question. It uses object hierarchy to compute the actual location. Assemblies are located in global coordinates, objects in assembly coordinates and features in object coordinates.
Table 18. Command Verbs - List and Classification

**Modeling Verbs**
- add
- addto
- alter
- append
- assemble
- attach
- cancel
- change
- create
- define
- delete
- discard
- edit
- eliminate
- enlarge
- erase
- exclude
- extrude
- fasten
- fetch
- fix
- include
- insert
- invert
- join
- label
- link
- loft
- merge
- mix
- model
- modify
- reduce
- remove
- rename
- replace
- reshape
- scale
- shape
- stretch
- subtract
- sweep
- truncate
- use

**Transformation Verbs**
- locate
- move
- orient
- place
- position
- put
- rotate
- translate
- turn

**Archiving, Display and Other Verbs**
- abort
- bye
- compare
- display
- draw
- end
- execute
- exit
- file
- find
- frame
- hardcopy
- help
- hide
- inquire
- list
- open
- pan
- pause
- plot
- query
- quit
- recall
- render
- run
- save
- shade
- stop
- store
- view
- zoom
Positioning with respect to another component is done by using ‘prepositions’ and the ‘name’ of a pre-defined component. Table 19 gives a list of common prepositions that are used commonly in describing relative positions.

**[orientation]**: This represents the orientation of the component in question. It uses the method mentioned above to compute the actual orientation of the component.

The set of elements must be complete for every command. In certain cases when some elements are not required, their values default to ‘not applicable’ so that no ambiguity will exist while interpreting the input. This structuring scheme gives flexibility to the type of logical input that can be interpreted by software.

### 3.2 Overview of UML

With this in mind, an English language interface was developed and can be used to provide a wide variety of functions like modeling (creation and modification of models), viewing (transformations) and interrogation (inquiries for definition, parameters, etc.). The description given below gives an idea about UML construction and how individual pieces contribute to its working. Figure 18 shows the elementary steps in processing string inputs by UML.
Table 19. Commonly Used Prepositions.

<table>
<thead>
<tr>
<th>Preposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>above</td>
</tr>
<tr>
<td>across</td>
</tr>
<tr>
<td>after</td>
</tr>
<tr>
<td>among</td>
</tr>
<tr>
<td>angled</td>
</tr>
<tr>
<td>around</td>
</tr>
<tr>
<td>axial</td>
</tr>
<tr>
<td>back</td>
</tr>
<tr>
<td>before</td>
</tr>
<tr>
<td>below</td>
</tr>
<tr>
<td>beneath</td>
</tr>
<tr>
<td>between</td>
</tr>
<tr>
<td>border</td>
</tr>
<tr>
<td>bottom</td>
</tr>
<tr>
<td>center</td>
</tr>
<tr>
<td>concentric</td>
</tr>
<tr>
<td>eccentric</td>
</tr>
<tr>
<td>edge</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>external</td>
</tr>
<tr>
<td>front</td>
</tr>
<tr>
<td>horizontal</td>
</tr>
<tr>
<td>in</td>
</tr>
<tr>
<td>inside</td>
</tr>
<tr>
<td>internal</td>
</tr>
<tr>
<td>into</td>
</tr>
<tr>
<td>left</td>
</tr>
<tr>
<td>lower</td>
</tr>
<tr>
<td>middle</td>
</tr>
<tr>
<td>normal</td>
</tr>
<tr>
<td>offset</td>
</tr>
<tr>
<td>outside</td>
</tr>
<tr>
<td>over</td>
</tr>
<tr>
<td>perpendicular</td>
</tr>
<tr>
<td>radial</td>
</tr>
<tr>
<td>right</td>
</tr>
<tr>
<td>start</td>
</tr>
<tr>
<td>through</td>
</tr>
<tr>
<td>top</td>
</tr>
<tr>
<td>under</td>
</tr>
<tr>
<td>upper</td>
</tr>
<tr>
<td>vertical</td>
</tr>
</tbody>
</table>
Figure 18. Input Processing by UML.
3.2.1 Word Parser

The word parser is used to break the sentence into its individual elements, for example, text or numbers. Any 'separators' are recognized and are discarded while parsing. The elements are then passed to the next part of the program for further treatment. Table 20 gives the list of separators that can be recognized by UML.

3.2.2 Structure Generator

This portion of UML restructures the parsed input into a form that is more understandable by the interpreter. Technical descriptions of a product need not contain words or phrases that are obvious from the context. Such 'filler words' do not play a constructive role in sentence structures and can be removed in order to simplify the sentence. The next step is to convert the sentence into a understandable grammatical structure or 'representation language'. All words of the sentence are checked and assigned as values to the appropriate element of the structure. This allows the user to enter identical commands in a number of ways rather than just one fixed format. However, it is necessary to remove all ambiguities from the sentence. Any missing information must be compiled by either prompting the user, by using rules and definitions in the data structure, or from previously established references. The complete structured sentence is now passed to the interpreter for decision making. Table 20 lists filler words that are removed from the sentence.
Table 20. List of Filler Words and Separators

### Filler Words

- a
- an
- and
- as
- at
- by
- called
- for
- from
- in
- into
- it
- location
- named
- of
- on
- orientation
- out
- parameters
- the
- this
- to
- up
- upto
- with

### Separators

- (  
- )  
- :  
- #  
- ,  
- @  
- {  
- }  
- ?  
- !  
- "
3.2.3 Interpreter

This is the most important module of UML. A complete analysis of the command is done in this section and the direction of flow of the program is decided. Each element of the statement structure affects the direction and hence each must be analyzed. The 'verb', which has primary influence on the command, is first analyzed and then classified as a modeling, graphics, inquiry or system function. The control is then passed to the appropriate function module for complete implementation of the command.

3.3 Execution of Input Command

Once structuring has been completed, and each element has been assigned a value, they are individually processed and interpreted. If an element has missing information that is required for successful execution of the command, or the information does not meet the guidelines for the specific command, then the user is prompted for additional or alternate information. In some cases, it may be difficult for a user to decide the value to be assigned. Then, the program has the capability of using rules and references to decide the value to be assigned to the element.

Now, the command is ready to be fully executed. The control is passed to each element utility which in turn governs the further transfer of control. The first element to be analyzed is the 'verb'. It is classified into 'modeling', 'graphics' or 'inquiry' functions. This specifies the set of elements required, and the order in which they
must be processed. In case of a ‘modeling’ function, the elements that must be processed include the ‘component’ type and ‘name’. If the value of ‘component’ is ‘feature’ then all other elements are also needed. A ‘graphics’ function generally does not affect the model. Hence, only the ‘parameters’ element is needed. Some commands, like ‘rendering’, also require the ‘name’ of a component to which the function must be applied. An ‘inquiry’ function is normally applied to components and requires the component ‘name’. Other elements do not affect this function and hence are not needed.

Each element that is processed allows the program to make partial decisions. Each partial decision is translated into a set of environment dependent calls. A series of such decisions leads to complete implementation of a command.

### 3.3.1 Sample Conversion of Input by UML

As described earlier, UML consists of three major steps. Application of UML to a feature-based modeling system can be described by means of an example. Consider the following input:

**Define a long rod with parameters (20.0, 150.0) at location (0.0 0.0 0.0) and orientation (30.0, 30.0, 30.0)**

This input first passes through the parser which removes the separators and returns a string array:

**Define a long rod with parameters 20.0 150.0 at location 0.0 0.0 0.0 and orientation 30.0 30.0 30.0**

This array is now processed and all filler words that do not form part of the structure are removed:
Define long rod 20.0 150.0 0.0 0.0 0.0 30.0 30.0 30.0

Now the structure elements are given the appropriate values so that:

[verb] = DEFINE
[adjective] = LONG
[component] = FEATURE
[name] = ROD
[parameters] = 20.0 150.0
[location] = 0.0 0.0 0.0
[orientation] = 30.0 30.0 30.0

3.4 Application of UML in Modeling Systems.

UML has been developed to act as a tool that allows modeling in an environment of a user’s choice. English ASCII characters can be processed by all computer systems and any hardware. Hence any software, such as UML, which uses English-like language to interact with its users can operate independently of the environment. Since all commands (except graphics display functions) are independent of the modeling environment, it is easy to integrate applications programs that have been customized for specific graphics packages.

An important application of UML in feature-based modeling systems is the ability of the user to interactively create a geometric model representing a feature or to make a series of inquiries about it. UML uses the feature library to answer user queries. This allows the user to extract information of mathematical rules, default parameters, etc., and helps decide which feature is best suited for a specific application. It allows the user to pass feature parameters to a modeling utility that can return feature definition in terms of parametric equations, surface points or the...
control polyhedron for uniform rational B-spline surfaces. The set of surface points allows integration of other surfaces to the modeler. This allows the user to create models using different kinds of surfaces. UML can also be used with application-specific menu structures. If the hierarchy of the menu structure is known, the user can, by a simple command, get the elements that form a specific menu page. The elements are available as an array of character strings and can be used by the programmer to write an interactive program that uses a graphics support of his choice.

Thus, UML acts as a factor in unification of different modeling environments. It can be used as a stepping stone to develop natural language procedures that allow quick, efficient and complete creation or interrogation of modeling databases. UML presents the best approach for future integration of software with voice recognition systems that translate voice into a string of ASCII characters. This will allow the user to interact with software in a manner that he is best suited and trained for - conversing in simple English.
Chapter 4: Geometric Modeling of Features

In keeping with today’s advanced styling requirements, a designer should be able to quickly create a variety of sophisticated shapes. Most shapes can be modeled using straightforward modeling schemes. Some complex shapes require blending of a series of surface profiles. Such complex sculptured surfaces cannot be modeled using normal techniques. This may require ‘lofting’ and ‘sweeping’ procedures which would allow a designer to specify complex planar shapes at different positions and orientations that can be lofted or cross-sections that may be swept about complex three dimensional curves.

Of all the methods used in the creation of solid models, B-Rep (Boundary Representation) appears to be the most suitable general method to define a solid model using features. The B-Rep method makes use of the exterior surfaces to create a solid model. Such surfaces should be versatile in nature and easily modeled. Several surfaces are popular in geometric modeling. However, B-spline surfaces have distinct advantages over others. They have local control over the shape of the curves that define the surface and hence complex shapes can be
modeled with ease. Use of rationality further enhances this characteristic. Uniform rational B-spline surfaces will be used as a base for feature representation.

4.1 Modeling of B-spline Surfaces

In order to successfully implement a surface modeler, it is necessary define the basic mathematics for the surface. This has been developed over the years and is explained in detail by Pratt (1979), Mortenson (1985), Riesenfeld (1973), Verspriille (1975) and many others. A B-spline surface is defined in terms of its characteristic polyhedron. The shape of the surface approximates the polyhedron. Any quadrilateral patch element on the surface defined by \((m + 1) \times (n + 1)\) points, can be expressed by:

\[
p(u,w) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} h_{ij} P_{ij} N_{i,k}(u) N_{j,l}(w)}{\sum_{i=0}^{m} \sum_{j=0}^{n} h_{ij} N_{i,k}(u) N_{j,l}(w)}
\]  
(4.1)

where \(P_{ij}\) are the vertices of the defining polyhedron, \(h_{ij}\) are the corresponding homogeneous coordinates and \(N_{i,k}(u)\) and \(N_{j,l}(w)\) are the blending functions (given below) whose degrees are governed by parameters \(k\) and \(l\). The blending functions for \(N_{i,k}(u)\) are defined recursively by the following expressions:

\[
N_{i,1}(u) = \begin{cases} 
1 & \text{if } t_i \leq u < t_{i+1} \\
0 & \text{otherwise} 
\end{cases}
\]  
(4.2)
\[ N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \] (4.3)

where \( t_i \) are the 'knot values' that relate the parametric variables to the control points. The blending functions for \( N_{i,j}(u) \) also have the same form.

This is the general representation for a B-spline surface. If the knot values are uniformly spaced in the range \( k \leq i \leq m \) and \( l \leq j \leq n \) then the resulting surface is a uniform B-spline surface. If they are not, then the resulting surface is a non-uniform B-spline surface. The homogenous coordinates \( h_{ij} \) act as weights on the corresponding control point and have the same effect as having multiple-coincident points. They help in controlling the shape of the surface by either pulling it closer or pushing it further from the control point. If \( h_{ij} = 1.0 \) for all points, the denominator in equation (4.1) goes to 1.0 and the resulting surface is a non-rational surface.

For this work, we will consider uniform rational B-spline surfaces. Hence the knot values must be equally spaced. Let the values of \( t_i \) be given by:

\[ t_i = \begin{cases} 1 & \text{if } i < k \\ i - k + 1 & \text{if } k \leq i \leq m \\ m - k + 2 & \text{if } i > m \end{cases} \]

and let \( t_j \) be given by similar expressions. The resulting blending functions act like switches, turning on and off the terms that they control. The symmetries and congruences of their shapes allow us to develop a more convenient and more familiar matrix notation. Choose an interval in \( i \) and \( j \) so that \( k \leq i \leq m \) and \( l \leq j \leq n \).
Let \( t_i \) and \( t_j \) be computed from equation (4.4) so that \( t_i = i - 3 \) (\( k = 4 \), for cubic splines) and \( t_j = j - 3 \) (\( l = 4 \), for cubic splines). Using the recursive expressions in (4.2) and (4.3) and reparameterizing the intervals we get a general matrix form for the surface as:

\[
p_{st}(u,w) = \frac{U_k}{U_k} M_k (hP)_k W_i^{T} \quad s \in [1:m + 2 - k] \]
\[
M = \begin{bmatrix}
-1 & 3 & -3 & 1 \\
3 & 6 & 3 & 0 \\
-3 & 0 & 3 & 0 \\
1 & 4 & 1 & 0
\end{bmatrix}
\]

\[
W_i^{T} \quad t \in [1:n + 2 - l]
\]

\[
u, w \in [0, 1]
\]

For cubic surfaces, \( k = l = 4 \) so that:

\[
U = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}
\]

\[
W = \begin{bmatrix} w^3 & w^2 & w & 1 \end{bmatrix}
\]

\((hP)_k\) is the matrix containing the control polyhedron and the weights, and \( h_k \) is the matrix containing the weights. Thus if the points on the control polyhedron are known then the surface can be computed using the above expression. The position of the control points govern the shape of the surface and hence must be chosen judiciously. The process of specifying the control polyhedron to create a surface will be called the forward modeling process. Figure 19 shows a simple example of modeling a B-spline surface using the forward modeling technique. It shows the control polyhedron and the resulting B-spline surface.
Figure 19. Example of Forward Modeling.
4.2 Inverse Modeling of B-spline Surfaces

The ability to model free-form and sculpted surfaces is dependent on the ease with which the set of control points is predicted. The capabilities of the modeler will be enhanced if a designer is allowed to simply specify a finite set of points that lie on the surface. Some surface defining techniques interpolate a given set of points, which means that the surface produced passes exactly through the control points. These techniques have a disadvantage when incorporated into an interactive CAD program. Specifically, we do not get a strong intuitive feel for how to change or control the shape of the surface. For example, if we try to change the shape of a spline-interpolated surface by moving one or more control points, we may produce unexpected and undesirable perturbations and inflections. Alternatively, another approach defines a surface that only approximates or approaches the control points. In such cases it is easier to predict the changes in the shape of the surface and provide better shape control. However, since they do not pass through the control points, it is difficult to control their exact shape and size. Periodic B-splines fall in the latter group: that is, they do not interpolate the end-points on the characteristic polyhedron and seldom pass through other control points. The use of parametric equations to define the geometry of a part requires that the surface pass through the set of computed points. Thus a method must be devised to accurately compute and construct a polyhedron that will ensure that the surface will pass through the specified set of points. Also, in case modifications are required, it is better that the surface shape be controlled in a predictable way by changing only a few simple parameters.
This method will be called the inverse modeling method and can be derived by applying matrix operations on the basic representation for B-spline surfaces. This leads to the following inverse relation:

\[ P_{ki} = M_k^{-1} V_k^{-1} p_{2d(u, w)} X_i^{-1} (M_i^T)^{-1} \]  

(4.6)

For, \( k = l = 4 \) \( V \) and \( X \) are:

\[
V = \begin{bmatrix}
  u_1^3 & u_1^2 & u_1 & 1 \\
  u_2^3 & u_2^2 & u_2 & 1 \\
  u_3^3 & u_3^2 & u_3 & 1 \\
  u_4^3 & u_4^2 & u_4 & 1 
\end{bmatrix} \quad \quad X = \begin{bmatrix}
  w_1^3 & w_2^3 & w_3^3 & w_4^3 \\
  w_1^2 & w_2^2 & w_3^2 & w_4^2 \\
  w_1 & w_2 & w_3 & w_4 \\
  1 & 1 & 1 & 1 
\end{bmatrix}
\]

The values of the parameters \( u \) and \( w \), corresponding to each point, can be specified by the user or can be computed internally by equal spacing parameterization. Figure 20 shows an example where the specified points through which the surface must pass, the computed polyhedron and the modeled surface are shown. Using this inverse relation one can now specify the points that the surface patch will interpolate and compute the corresponding control points, which will not be interpolated, needed to generate the patch.

A disadvantage of surfaces that do not directly interpolate control points is that any change in the control points may result in unpredictable changes in the surface. Thus a designer may have little knowledge of the local shape of the surface, making it difficult to model complex shapes. A distinct advantage of using this inverse method is that the designer can now reshape a surface by recomputing the surface points without worrying about the changes in the control points or the transmission of this change to the surface.
Figure 20. Example of Inverse Modeling.
This procedure can be used very efficiently for feature-based modeling. Since, a designer defines a part in terms of its parameters, they can be used to compute the points lying on the surface of the feature. The inverse relation can then be applied to create a B-spline surface that interpolates the computed points.

The method allows the modeling of single surface patches. In general, a feature will consist of more than one surface patch. Thus it is necessary to break down a feature in a manner that allows easy implementation of inverse modeling. A large patch requires a greater number of points to ensure the accuracy of the shape. This means a larger number of points must to be computed and interpolated. Hence it is necessary to find the optimum shapes for which the inverse method will work accurately and easily. It was found that this method worked best when (if possible) a feature is broken down into surfaces that are rectangular, cylindrical or elliptical in shape. Rectangular shapes are modeled as one patch, whereas cylindrical and elliptical surfaces are split into patches that cover one quadrant or octant as the case may be. This ensures that the symmetric nature of a feature is preserved, and that the surface is represented accurately. Some examples of the basic patches used are shown in Figure 21.

This method was used to model several features. Figures 22 and 23 show hidden surface images of some features that were modeled. Figure 22 shows the surfaces defining an ellipsoid, a neck, a hollow cone, a solid cone, a cone with a cylindrical hole, a counter-bore and a counter-sink. Figure 23 shows surfaces representing a sphere, hemisphere, hollow hemisphere, cube, truncated prism, wedge, cylinder, hollow cylinder and an arch.
Figure 21. Examples of Patches Using Inverse Solution.
Figure 22. Examples of Features Using Inverse Solution.
Figure 23. Examples of Features Using Inverse Solution.
Chapter 5: FeatureMod - A Prototype Feature Based Modeler

We can now investigate the possibility of implementing a prototype feature-based modeling system. This system should incorporate the ideas discussed in preceding sections and also meet the basic requirements of a feature modeler. A good system can be developed only after taking an in-depth look at such requirements.

5.1 Requirements of a Feature Modeler

A good feature modeler should address several issues of geometric modeling. It should be capable of operating independently in any computer environment, be coupled to a graphics display system, perform basic functions of modeling and have a data structure that contains complete information of a geometric model. Functional
requirements of a feature-based modeling system include system, graphics and modeling functions. In addition, a data structure that contains information of different elements in the model should be developed. In order to meet these requirements, a brief analysis of such functions is necessary. Archiving and graphics functions are similar for any modeling environment. Modeling functions and the data structure are more important and are discussed as they apply to feature-based modeling.

**Archiving Functions:** Any modeler should be independent of its operating system. To operate as a complete system it should have the capability of model storage and retrieval without loss of any information. Such functions act as a link between the designer and the data structure. Extensive queries on the data structure should be possible and external links to peripherals like printers and plotters should be established. Thus functions like store, recall, exit, query, list and plot should be supported by the modeler.

**Graphics Functions:** Today’s development environment requires designing at different locations simultaneously. The ability to combine this work will greatly depend on the compatibility of the modeling systems used. Older systems use graphics software that is device dependent and hence is not portable and has compatibility problems with other systems. The need to provide a portable modeling system is the most important reason to have device independence. This can be achieved by the use of ISO standards for three dimensional graphics. A modeler that utilizes graphics software that meets the specifications and follows guidelines of three dimensional graphics standards will be portable and can be used at different sites without difficulty.
Viewing transformations allow a designer to observe an object in different orientations. Hence functions like rotate, zoom and pan functions that provide single or multiple views should be implemented. Other graphics support needed include rendering, hidden surface removal and shading to create realistic images and provide a designer with a better perspective of the object and the status of design. Such functions will also help remove viewing ambiguities that are present in wireframe and surface models.

**Modeling Functions:** The virtues and drawbacks of a modeler originate in the way that modeling is handled. The ease with which modeling functions are executed will determine the quality of the modeler. Some examples of functions are creation, deletion, modification, positioning, copying, instancing and interrogation. Creation and deletion of features should be handled unambiguously and at the designer’s instigation. Modification of existing features that result in local surface redefinition or even a completely new set of parameters should be possible. Positioning by means of translation and rotation transformations will allow global positioning of the feature leaving its geometry and defining dimensional parameters unchanged. Copying is similar to moving, except that the original feature still remains in position and maintains its separate identity. Establishment of object hierarchy allows a designer to apply modeling functions at different levels, for example assemblies, objects and feature instances. It also allows model creation by means of instancing.

Execution of any modeling function leads to changes at all levels of the model and should be transmitted correctly to the data structure. The capability of a modeler to handle extensive interrogation will be governed by the amount of information available in the database. Any analysis to be done on the model will require a great
deal of information from the database. A good data structure should store information that will make such analysis easy and quick. Information about feature parameters, surface definition, location, orientation, attributes and its level in the object hierarchy are some examples of information that should be stored in the database.

Keeping the requirements in mind, a prototype system called FeatureMod has been developed. Written in FORTRAN 77, it uses graPHIGS, the IBM version of PHIGS, for graphics display. The modeling system is operational on the IBM 5080. However, any device that supports graPHIGS can be used for modeling. As mentioned before, FeatureMod uses the previously discussed feature dictionary to create surface models of features using rational B-splines and allows user interaction using menu picks or UML.

5.2 Overview of FeatureMod

FeatureMod is a prototype feature-based modeling system that allows the creation of a design by using a vocabulary of familiar geometric features and shape descriptors. Figure 24 shows the data flow diagram of FeatureMod. A user can interact with the modeler through a modeling and graphics interface. This interface allows a user to create features by accessing the feature dictionary and modeling library. Graphics utilities for transformations and rendering can also be accessed through this interface.
Figure 24. Data Flow Diagram of FeatureMod.
Interaction with the modeling and graphics interface can be achieved by two different methods. The first method utilizes the previously discussed Universal Modeling Language and allows a designer to take a natural language approach towards modeling. The second method is a conventional menu pick method that requires that a user be familiar with the hierarchy of the system.

The graphics library consists of graPHIGS routines which allow display of 3-D graphics primitives such as lines, markers, text, etc. and allows attachment of different attributes to these primitives. The system library contains procedures that allow execution of system functions like store and recall. Transformation library contains routines that will allow a user to display single or multiple orthographic views of an object. Valuators are used to apply window transformations for rotation, pan and zoom. The feature library contains information of different features available for modeling and is used in deciding whether a feature meets the inequality criteria. The modeling library consists of procedures to create a hierarchical feature-based model. The database contains complete information about the geometric model, its attributes and the object hierarchy.

A graphics display, attached to the interface, is mainly used to display the current status of a design and as means of communication between the user and the modeler. The display screen is broken into several areas used for component display, menu display, prompt display, scroll message area, template area, string input and valuator display area. The reserved areas on the screen are shown in Figure 25.
VALUATOR AREA
DISPLAY AREA
MENU AREA

Figure 25. Reserved Areas on Display Screen.
5.3 Object and Menu Hierarchy in FeatureMod

FeatureMod uses a bottom-up approach for model creation. A model consists of assemblies, objects and feature instances. Assemblies occupy the highest levels in the hierarchy and the lowest levels are occupied by features. Thus features can be created in local coordinates and be placed in the parent object's coordinate system. Similarly an object can be placed relatively in an assembly's coordinate space. A representative object hierarchy for a model is shown in Figure 26. A feature can be part of different objects, but an object can be a part of only one assembly. If an object has to be used in more than one assembly, it should be copied before being used again.

The use of menus in the modeler make the establishment of a menu tree necessary. The tree is based on the command sequence that should be executed in order to reach a particular location in it. Thus each menu item is accessible only by a pre-specified path. This means that a user must be familiar with the modeling procedures in order to use the system efficiently. The menu tree is shown in Figure 27.

5.4 Data Storage in FeatureMod

To support easy evaluation of a model the geometric database should represent the model explicitly in terms of features and other data related to them. This section
Figure 26. Representative Object Hierarchy in FeatureMod.
Figure 27. Menu Hierarchy in FeatureMod.
briefly describes how the symbolic representation of the design geometry in FeatureMod provides the needed information. The database contains information about different assemblies, objects and features. Figure 28 gives an idea about the information stored for each component and how they relate to each other.

**Assembly:** The information consists of assembly number, assembly name, its position and orientation in the global coordinate system and identities of the objects that make up that assembly.

**Object:** Similar to an assembly, object information consists of object number, object name, parent assembly identity, its position and orientation in the assembly coordinate system and identities of the features that make up that object.

**Feature:** This is the lowest component in the object hierarchy and hence contains maximum information. Parameters defining the geometry of the features are stored separately for convenience. In order to simplify computation, it stores the control polyhedrons defining the surfaces that make up the feature. In addition it contains information about the feature number, feature name, parent object identity, its position and orientation in the object coordinate system and attributes to be used for computation and display.

### 5.5 Design Methodology for FeatureMod

The procedure required to create a design is simple. The user first has to initiate a model that can later be stored or retrieved. Once initialization is complete the user can now start modeling. First, the user should build an object by using feature instances. Features are created by specifying a feature name from the
Figure 28. Data Storage in FeatureMod.

<table>
<thead>
<tr>
<th>ASSEMBLY</th>
<th>LOCATION</th>
<th>ORIENTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>NUMBER OF OBJECTS</td>
<td></td>
</tr>
<tr>
<td>OBJECT IDENTITIES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT</th>
<th>ASSEMBLY</th>
<th>LOCATION</th>
<th>ORIENTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>NUMBER OF FEATURES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEATURE IDENTITIES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>OBJECT</th>
<th>LOCATION</th>
<th>ORIENTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>ATTRIBUTES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SURFACE PATCHES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
dictionary or by picking from the displayed menu. Parameters required for feature
definition are obtained by means of templates. A template with default parameters
for the feature is displayed in the message area. The user is allowed to change the
parameter values within a range specified by inequality constraints. If all parameters
are not specified, then the list is completed by using mathematical relations that
govern the feature definition. The feature can be positioned by the user and
attributes for surface creation and display can be attached to it. In some cases
different types of features may be available and can be selected by toggling the menu
item. Figure 29a shows a template used for defining a counter-bore. By selecting
item 'counter' the same template can now be used for defining a counter-sink. Figure
29b shows the template after toggling. After an object is defined it can be edited by
using the 'edit' command or can be added to by the 'addto' command. Once different
objects are defined, they can be assembled into one assembly. Assembling is also
done with the help of templates which allow positioning of entire objects with respect
to the assembly's coordinate axes. As mentioned before, an object can be a part of
only one assembly. If an object is already a part of an assembly or is non existent,
then the user is warned and is advised to take corrective action. Figure 30 shows
example templates used for creating assemblies with warning prompts.

Only one object or assembly is displayed on the screen at a time. This allows
the display on the screen to be kept at a minimal level. However, when necessary
the entire model can be displayed by using a simple command. Viewing
transformations and rendering can be applied to the model at any time for improved
visualization of the object.
Figure 29. Templates for Creating (a) Counter-bore and (b) Counter-sink.
<table>
<thead>
<tr>
<th>NON-EXISTENT OBJECT</th>
<th>OBJECT = OBJECT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION = 25.0 -40.0 30.0</td>
<td></td>
</tr>
<tr>
<td>ORIENTATION = 30.0 30.0 30.0</td>
<td></td>
</tr>
<tr>
<td>ASSEMBLY COMPLETE</td>
<td></td>
</tr>
<tr>
<td>EXECUTE</td>
<td></td>
</tr>
<tr>
<td>ABORT</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT ALREADY ASSEMBLED</th>
<th>OBJECT = OBJECT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION = 25.0 -40.0 30.0</td>
<td></td>
</tr>
<tr>
<td>ORIENTATION = 30.0 30.0 30.0</td>
<td></td>
</tr>
<tr>
<td>ASSEMBLY COMPLETE</td>
<td></td>
</tr>
<tr>
<td>EXECUTE</td>
<td></td>
</tr>
<tr>
<td>ABORT</td>
<td></td>
</tr>
</tbody>
</table>

Figure 30. Sample Templates for Creating Assemblies.
Thus FeatureMod is a working prototype that allows easy creation of complex geometric models and uses high-level representation of the part geometry in the form of features. Used as an interactive CAD environment with color graphics it allows the building of a model by reasonable and logical definitions of features. A dual input system that uses UML and menu picks allow both advanced and entry-level users to use the system efficiently. The combination of advanced surface modeling and parametric programming capabilities allows creation of models that communicate precise geometric information, and, possibly manufacturing information to the production personnel. The ability to simultaneously render image and information makes FeatureMod an elegant tool for enriching and refining the flow of information from engineering to manufacturing. Appendix B gives the entire listing of the routines that make up FeatureMod.
Chapter 6: Case Studies

6.1 Feature Vocabulary

The lists of features, shapes, forms and shape-altering words are combined to form a vocabulary that must match that of a designer. If the list is complete, then any word that a designer uses to describe a shape or form should use a combination of these lists to produce a parametric feature or a synonym of a feature. Thus, it is important that this list be complete.

In other words, a designer who intends to model any part should be able to use these lists, and with proper combinations be able to completely describe the part under consideration. In order to check for the completeness of the feature vocabulary, two parts were used to perform case studies.
6.1.1 Pulley Mount

The first part considered is an assembly used for mounting a shaft and pulley arrangement (Figure 31 and 32). A designer who is given the task of modeling this part might describe it in the following manner:

The part consists of a thin plate (used as a base) with two through holes and two tapped-holes. A vertical column is attached to the plate by means of a weld. A cantilever beam that supports the pulley is attached to the column by means of a deep split ring to be tightened around the column by means of a hex-bolt and hex-nut passing through a hole in it. A rib provides additional support to the beam. The shaft passes through a tapering hub attached to the pulley by means of thick rectangular spokes. The axial movement of the pulley is restricted by two collars, one screwed to the shaft and the other forming a part of the cantilever beam. The top end of the shaft has a tapped-hole for lubrication. The pulley rests on a conical hub which is part of the cantilever beam and collar arrangement. Figures 31 and 32 show different views of the part and the (visible) features used to describe the part.

6.1.2 Flow Controlling Valve

The second part considered is a valve which regulates the flow of fluid through an elbow joint (Figure 33). A designer may describe it as follows:

The part consists of two thick square plates with rounded corners and a valve seal between them. Each plate has four through holes which allow the plates to be clamped together by means of hex-nuts, washers and hex-bolts. The lower plate has flanges with rectangular slots on two sides. All corners are rounded. This is done
Figure 31. Features Describing a Pulley Mount - I.
by holes on either end. The pipe is attached to the lower plate by means of a weld. The upper plate has a conical hub on which sits the hand wheel. The hand wheel has a conical dip on the top and a cylindrical boss at the bottom. The flow control shaft is threaded and goes through a hole in the conical hub. Figure 33 shows the part and the (visible) features used to describe the part.

6.2 FeatureMod Implementation

The pulley mount was also modeled on the prototype feature-based modeling system that has been developed. The set of features that were identified from the designer's description were used to form the part. Figures 34 through 37 show the modeled part. Since no Boolean operations have been implemented, the model consists of a set of features but is not the final product.
Figure 33. Features Describing a Flow Controlling Valve.
Figure 34. A Pulley Mount Arrangement Created by FeatureMod - Front View.
Figure 35. A Pulley Mount Arrangement Created by FeatureMod - Top View.
Figure 36. A Pulley Mount Arrangement Created by FeatureMod - Right Side View.
Figure 37. A Pulley Mount Arrangement Created by FeatureMod - Isometric View.
Chapter 7: Conclusions and Recommendations

7.1 Conclusions

There is a great deal of research activity in the area of feature-based modeling. Prior to this, no in-depth study of commonly used design features has been done. A feature-based modeling system can be useful only if it has the capability of representing a large number of features and makes the task of creation quick and easy for the designer.

There is a need to develop methods that not only make the task of feature creation easy, but also make the selection process an intelligent one. The effectiveness of such methods will depend on the number of features that are available for creation and the ease with which a designer can modify them to create different variations. There is also a need to speed up the feature creation procedure. This is mainly governed by the manner in which feature dimensions and other parameters can be specified. If a designer is allowed to use any word describing
shape or form to produce a feature, newer methods for feature selection and creation need to be developed. This clearly precludes the use of conventional menu screens. A natural language approach seems to be necessary. Thus, a method that uses a natural language approach to allow intelligent selection and creation of features seems to be an appropriate step in speeding the design process.

The main objective of this research was to find a suitable approach that bridges the gaps in the existing feature modeling methods. It aims at formulating a generalized procedure for using features as the basis for creation of geometric models and to best exploit the existence of a large vocabulary among designers or production personnel for this purpose. An exhaustive study of English terms used to describe the shapes and forms of geometric features was performed to improve the ease with which design by features can be accomplished. A study of shape altering words that result in additional description or modification of a feature was done. In order to simplify implementation, a method of systematic grouping of features with similar topologies was developed. Features were also classified by shape forms to permit quick access of similar features based on recollection of similar features. A dictionary that allows a designer to better understand the nature and function of a feature was created.

Parametric features were created from groups to allow easy definition, creation and modification. Prototype inequality relations to differentiate between features in the same topology group were established. These rules establish relations between different parameters that define a feature and assist a designer in selecting reasonable values for the parameters. They also assist in creation of a feature through incomplete specification of parameters. A method to compute a set of

Conclusions and Recommendations
parameters using the mathematical inequalities and partial specification of parameters in the set was developed.

The parametric equations and the parameter values decide the points that lie on the surface. Any surface used for representation must interpolate these points. The resulting feature is modeled with rational B-spline surfaces. These surfaces are versatile and allow local control of a shape, but by nature do not interpolate points on the defining control polyhedron. An inverse relation that allows a B-spline surface to interpolate pre-determined points was developed. This approach gives great flexibility in modeling of complex shapes and their surface representation.

To allow a designer to use similar procedures for visualizing and creating features, a natural language approach seemed necessary. A modeling language that allows a designer to use simple English commands to perform different tasks of modeling was developed. It provides a designer with means of easy communication with a modeling system, and hence eases the manner in which he may define a feature.

Finally, a prototype system for designing with features that partially incorporates the above-mentioned work and other requirements for feature-based modeling systems was developed. It allows creation of geometric models using object hierarchy (assemblies, objects and feature instances) and easy placement and aligning of different components. This provides a designer with a flexible tool for using features to create a part.

To conclude, this work provides a designer with a flexible method for feature selection for creation of a part. It provides him/her with an easy method for
specifying commands, since it uses simple English for input. The use of mathematical relations further enhances the ability of a designer to produce reasonable features and also allows incomplete specification of parameters for feature creation.

7.2 Recommendations

The work described in this dissertation is comprehensive but is definitely not complete. The nature of mechanical parts requires that all possible shapes and forms be accounted for. It is obvious that a larger number of topological groups will provide better flexibility in modeling different shapes. There is a need to identify more specialized shapes that can be grouped into newer topologies and to parameterize them. The classification schemes should be further developed to account for any specialized shapes that cannot be grouped currently.

While mathematical rules for some features were established, those for the other features must be compiled. These rules should be flexible and may be defined by a user. The ability to use mathematical relations for computing missing parameters must be further enhanced by developing schemes that can handle non linear relations. The vocabulary of modeling verbs and shape altering words must be enhanced to account for the most complex commands and modifications.

It is recommended that further research be done in processing of natural language input and be used as a base for achieving the ultimate of goal of processing voice input. While uniform rational B-spline surfaces are suitable, it is recommended
that non-uniform rational B-spline surfaces (NURBS) be implemented for added flexibility.

The ability to use a model for mass properties, interference checks and other utilities depends on its completeness. A model is complete only if it is a solid model. Thus, procedures for converting a surface model into a solid must be implemented. Additionally, Boolean operations must be implemented. Boolean operations require the ability to compute the intersection between two solids. The difficulty with which intersections can be found depends on the level at which the problem is to be handled. The lower the level, the easier it is to compute the intersections. An efficient method to compute surface intersections must be implemented to make Boolean operations easier.
References


FEATURE NAME : ABUTMENT
ENGLISH DEFI : A surface or mass provided to withstand thrust.
GROUP CLASSI : TOPOLOGY GROUP - E  FORM GROUP - B2, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : AIGUILLE
ENGLISH DEFI : A slender form of drill used for boring or drilling a blast hole in a rock.
GROUP CLASSI : TOPOLOGY GROUP - A  FORM GROUP - B1, BSB
SYNONYMS : DRILL
RELATIONS : 
DEFINING PARAMETERS : R, L
END

FEATURE NAME : ANGLE
ENGLISH DEFI : The figure or space made by the meeting of two straight lines or two plane surfaces.
GROUP CLASSI : TOPOLOGY GROUP - H  FORM GROUP - B2
SYNONYMS : BEND, BRACKET, CORNER, ELBOW
RELATIONS : 
DEFINING PARAMETERS : L, B, H, T, R, A
END

FEATURE NAME : ANNULAR-HOLE
ENGLISH DEFI : A space between the casing and wall of a hole or between a drill pipe and casing.
GROUP CLASSI : TOPOLOGY GROUP - C  FORM GROUP - B1, C11
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R1, R2, L
END

FEATURE NAME : APERTURE
ENGLISH DEFI : A small opening or an orifice.
GROUP CLASSI : TOPOLOGY GROUP - B  FORM GROUP - B1, C11
SYNONYMS : INLET, OPENING, ORIFICE, PERFORATION, PUNCTURE
RELATIONS : 
DEFINING PARAMETERS : R, L
END

FEATURE NAME : APEX
ENGLISH DEFI : 
GROUP CLASSI : 
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : 
END
Feature Name: Apron

English Def: A plate serving to cover or protect a machine.

Group Class: Topology Group - E Form Group - B2, C3

Synonyms: 2

Relations: 1

Defining Parameters: 2 L, B1, B2, H

Feature Name: Arbor

English Def: A shaft, spindle or axle in different machines.

Group Class: Topology Group - A Form Group - B1, B5A, C1

Synonyms: Axle, Mandrel, Shaft, Spindle

Relations: 1 R, L

Feature Name: Arbor-Hole

English Def: A hole in a revolving cutter or grinding wheel for mounting it on an arbor.

Group Class: Topology Group - B Form Group - B1, C11

Synonyms: 1

Relations: 1

Defining Parameters: 1 R, L

Feature Name: Arc

English Def: Anything with a shape of an arch, a curve or part of a circle.

Group Class: Topology Group - D Form Group - B3

Synonyms: 1

Relations: 1 R1, R2, L

Feature Name: Arch

English Def: A curved structure spanning an opening.

Group Class: Topology Group - D Form Group - B3, C1

Synonyms: b

Relations: 1 R1, R2, L

Feature Name: Arm

English Def: A side of an angle.

Group Class: Topology Group - H Form Group - B3

Synonyms: 1

Relations: L, B, H, T, R, A

Feature Name: Arris

English Def: A short edge or angle at the junction of two surfaces, especially moldings and raised edges.

Group Class: Topology Group - H Form Group - B3

Synonyms: 1

Relations: L, B, H, T, R, A

Feature Name: Astragal

English Def: A small convex molding.

Group Class: Topology Group - J Form Group - B1, C1

Synonyms: 1

Relations: 1
DEFINING PARAMETERS : A, B
END
FEATURE NAME : AXLE
ENGLISH DEF : A supporting member that carries wheels and rotates with the wheel.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : ARBOR, MANDREL, SHAFT, SPINDLE
RELATIONS :
DEFINING PARAMETERS END
FEATURE NAME : BAFPLATE
ENGLISH DEF : A plate that regulates the flow of a fluid as in a steam boiler, flue or gasoline muffler.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - C, E FORM GROUP - B1, B2, C3, C12
SYNONYMS :
RELATIONS :
DEFINING PARAMETERS R1, R2, L
DEFINING PARAMETERS L, B1, B2, H
END
FEATURE NAME : BALL
ENGLISH DEF : A spherical or nearly spherical object.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS : GLOBE, SPHERE
RELATIONS :
DEFINING PARAMETERS R
END
FEATURE NAME : BAND
ENGLISH DEF : A flat flexible strip of any material often used for binding or securing.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - E FORM GROUP - B5A, C3
SYNONYMS : BELT, RIBBON, STRIP
RELATIONS :
DEFINING PARAMETERS L, B1, B2, H
END
FEATURE NAME : BAR
ENGLISH DEF : An elongated piece of metal of simple uniform cross section dimensions, usually rectangular, circular or hexagonal.
SYNONYMS : CORE, ROD
RELATIONS :
DEFINING PARAMETERS R, L
DEFINING PARAMETERS L, B1, B2, H
END
FEATURE NAME : BARREL
ENGLISH DEF : A container having a circular lateral cross section that is largest in the middle and ends that are flat.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS :
RELATIONS :
DEFINING PARAMETERS R1, R2, L
END
FEATURE NAME : BASE
ENGLISH DEF : The lowest or supporting part of anything.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - E FORM GROUP - B1, B2, C3
SYNONYMS : BED, BEDDING
RELATIONS :
DEFINING PARAMETERS L, B1, B2, H
END
FEATURE NAME : BASKET
ENGLISH DEF : A light weight container with perforations.
ENGLISH DEF : GROUP CLASS : TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS :

Appendix A - Feature Dictionary
RELATIONS
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : BEAD
ENGLISH DEF : A small, usually round object.
GROUP CLASSI : TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : BEAM
ENGLISH DEF : A body whose function is to carry lateral loads and bending moments.
GROUP CLASSI : TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS : R, L
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : BED
ENGLISH DEF : The part of a machine having precisely machined ways or bearing surface which supports or aligns other machine parts.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : BASE, BEDDING
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : BEDDING
ENGLISH DEF : That which forms a bed or foundation.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : BASE, BED
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : BELT
ENGLISH DEF : An endless band of flexible material for transmitting power from one wheel or shaft to another.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B5A, C3
SYNONYMS : BAND, RIBBON, STRIP
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : BENCH
ENGLISH DEF : A table for mechanical work.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : BEND
ENGLISH DEF : To assume the form of a curve; crook; bow.
GROUP CLASSI : TOPOLOGY GROUP - H FORM GROUP - B2, B3
SYNONYMS : ANGLE, BRACKET, CORNER, ELBOW
RELATIONS
DEFINING PARAMETERS : L, B, H, T, R, A
END
FEATURE NAME : BEVEL
ENGLISH DEF : An inclination of two surfaces other than perpendicular.
GROUP CLASSI : TOPOLOGY GROUP - V FORM GROUP - B1, B5A
SYNONYMS
RELATIONS
DEFINING PARAMETERS : R1, R2, L, A.R.
FEATURE NAME : BILLET
ENGLISH DEFI : A semifinished, short, thick, bar in form of a cylinder or rectangular prism.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : BLOCK
ENGLISH DEFI : A solid piece of metal with one or more flat surface.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : BOLSTER-PLATE
ENGLISH DEFI : A plate fixed on the bed of a power press to locate and support the die assembly.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A BUTTON, CAP, COVER, DIAL, DISK, HATCH, PLATE
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
END

FEATURE NAME : BORE
ENGLISH DEFI : To cut or drill a hole in or through a work piece.
GROUP CLASSI : TOPOLOGY GROUP - B FORM GROUP - B1, B5A, C11, FORM GROUP - C13
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B2, B5A, C1
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
END

FEATURE NAME : BOSS
ENGLISH DEFI : Protuberance on a part to add strength, facilitate assembly, provide for fastening, etc.
GROUP CLASSI : TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A, C1
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : BOW
ENGLISH DEFI : A part shaped as an arc or a polygon.
GROUP CLASSI : TOPOLOGY GROUP - D FORM GROUP - B3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R1, R2, L
END

FEATURE NAME : BOX
ENGLISH DEFI : A receptacle or a case usually having a lid.
GROUP CLASSI : TOPOLOGY GROUP - G FORM GROUP - B2B, C3 CASE, CASING, FRAME
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B, H, T
END

FEATURE NAME : BRACE
ENGLISH DEFI : A diagonally placed structural member that withstands tension and compression.
GROUP CLASSI : TOPOLOGY GROUP - H FORM GROUP - B3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B, H, T, R, A
END
<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BRACKET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A brace used to strengthen an angle; a piece of metal used to support an object.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - H FORM GROUP - B3</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td>ANGLE, BEND, CORNER, ELBOW</td>
</tr>
<tr>
<td>RELATIONS</td>
<td>L, B, H, T, R, A</td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>To separate into pieces or fragments; to crack.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - F FORM GROUP - C10</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td>CRACK, CREVICE, CUT, FISSURE, FRACTURE, RUPTURE</td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>The shaping of a part by pushing or pulling a broach across the surface or through an existing hole in the work piece.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - A, B FORM GROUP - B1, C13</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td></td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>R, L</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BUBBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A liquid globule filled with air or other gases.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td>DROP, DROPLET, GLOBULE, TEAR</td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>R</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BUCKET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A deep cylindrical vessel.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - C FORM GROUP - B1</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td></td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>R1, R2, L</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BULB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A rounded protuberance.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td></td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>R</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BULGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A protuberant, rounded part.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - J FORM GROUP - B1, C1, C7</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td>BUMP, ENTASIS, SWELL, SWELLING</td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>A, B</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BULLET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>Any small ball or cylindrical object.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - A FORM GROUP - B1, B5A</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td></td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td>R, L</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>BUMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGLISH DEF!</td>
<td>A protuberance or an uneven place.</td>
</tr>
<tr>
<td>GROUP CLASS!</td>
<td>TOPOLOGY GROUP - J FORM GROUP - B1, C1, C7</td>
</tr>
<tr>
<td>SYNONYMS</td>
<td></td>
</tr>
<tr>
<td>RELATIONS</td>
<td></td>
</tr>
<tr>
<td>DEFINING PARAMETERS</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
<tr>
<td>Feature Name</td>
<td>English Defi</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BUSHING</td>
<td>A metallic lining for a hole designed to insulate or prevent abrasion between moving parts.</td>
</tr>
<tr>
<td>BUTTON</td>
<td>A knob or disk serving as a fastening.</td>
</tr>
<tr>
<td>CABLE</td>
<td>A stranded rope like assembly of wire or fiber.</td>
</tr>
<tr>
<td>CAGE</td>
<td>A boxlike structure made up of wires or bars.</td>
</tr>
<tr>
<td>CAN</td>
<td>A cylindrical metal vessel or container, usually with an open top or a removable cover.</td>
</tr>
<tr>
<td>CANAL</td>
<td>A passage or a duct or a tube.</td>
</tr>
<tr>
<td>CANTILEVER</td>
<td>A long structural member, as a truss, beam or slab lying across the supports with projecting arms in balance.</td>
</tr>
<tr>
<td>CAP</td>
<td>An object that acts as a cover.</td>
</tr>
</tbody>
</table>
DEFINING PARAMETERS: L, B1, B2, H
DEFINING PARAMETERS: R1, R2

END

FEATURE NAME: CAPSULE
ENGLISH DEFINITION: A thin covering, container or seal.
GROUP CLASSIFICATION: TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS: CARTRIDGE
RELATIONS: R1, R2, L

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: CARTRIDGE
ENGLISH DEFINITION: Any small container or casing.
GROUP CLASSIFICATION: TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS: CAPSULE
RELATIONS: R1, R2, L

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: CASE
ENGLISH DEFINITION: A box for containing something.
GROUP CLASSIFICATION: TOPOLOGY GROUP - G FORM GROUP - B2B, C3
SYNONYMS: BOX, CASING, FRAME
RELATIONS: L, B, H, T

DEFINING PARAMETERS: L, B, H, T

END

FEATURE NAME: CASING
ENGLISH DEFINITION: A protective case or covering.
GROUP CLASSIFICATION: TOPOLOGY GROUP - G FORM GROUP - B2B, C3
SYNONYMS: BOX, CASE, FRAME
RELATIONS: L, B, H, T

DEFINING PARAMETERS: L, B, H, T

END

FEATURE NAME: CAVE
ENGLISH DEFINITION: A chamber beneath the earth. in a mountain.
GROUP CLASSIFICATION: TOPOLOGY GROUP - D FORM GROUP - B1, C2
SYNONYMS: 
RELATIONS: R1, R2, L

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: CAVITY
ENGLISH DEFINITION: A hollow or sunken space; hole.
GROUP CLASSIFICATION: TOPOLOGY GROUP - F FORM GROUP - B2, C2
SYNONYMS: CHANNEL, FURROW, GROOVE, RUT, SLOT
RELATIONS: L, B1, B2, H

DEFINING PARAMETERS: L, B1, B2, H

END

FEATURE NAME: CELL
ENGLISH DEFINITION: A small compartment, receptacle or cavity.
GROUP CLASSIFICATION: TOPOLOGY GROUP - G FORM GROUP - B2B
SYNONYMS: CHAMBER, COMPARTMENT
RELATIONS: L, B, H, T

DEFINING PARAMETERS: L, B, H, T

END

FEATURE NAME: CHAMBER
ENGLISH DEFINITION: An enclosed space or cavity; a sleeve or channel.
GROUP CLASSIFICATION: TOPOLOGY GROUP - G FORM GROUP - B2B
SYNONYMS: CELL, COMPARTMENT
RELATIONS: L, B, H, T

DEFINING PARAMETERS: L, B, H, T

END

FEATURE NAME: CHAMFER
ENGLISH DEFINITION: To cut away a corner of; a cut a furrow in.
GROUP CLASSIFICATION: TOPOLOGY GROUP - H FORM GROUP - B2
SYNONYMS: 

END
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHANNEL
ENGLISH DEFI A tubular passage, groove, furrow or a cavity.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - B2, C2, C10
SYNONYMS CAVITY, FURROW, GROOVE, RUT, SLOT
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHASE
ENGLISH DEFI To ornament by embossing or engraving.
GROUP CLASSI TOPOLOGY GROUP - J FORM GROUP - BSB, C1
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHASED-GROOVE
ENGLISH DEFI A groove or a slot.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - BSB
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHASSIS
ENGLISH DEFI A flat, rectangular frame on which a body is mounted.
GROUP CLASSI TOPOLOGY GROUP - G FORM GROUP - B2B
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHIMNEY
ENGLISH DEFI A tube, usually of glass or a funnel.
GROUP CLASSI TOPOLOGY GROUP - C FORM GROUP - B1, C1
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHORD
ENGLISH DEFI A line segment which intersects a curve or surface only at the end points of the segment.
GROUP CLASSI TOPOLOGY GROUP - A FORM GROUP - B4
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CHUTE
ENGLISH DEFI An inclined through or a vertical passage.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - BSB
SYNONYMS FLUME
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CLEAT
ENGLISH DEFI A strip of metal fastened across or projecting from a surface to strengthen, support or provide a grip.
GROUP CLASSI TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A
SYNONYMS FIN, RIB
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME CLEAVAGE
ENGLISH DEFI: A split or a cleft.

GROUP CLASSI: TOPOLOGY GROUP - F FORM GROUP - C10

SYNONYMS: CLEFT

DEFINING PARAMETERS: L, B1, B2, H

END

FEATURE NAME: CLEFT

ENGLISH DEFI: A fissure, divide or rift that separates partially or completely.

GROUP CLASSI: TOPOLOGY GROUP - F FORM GROUP - C10

SYNONYMS: CLEAVAGE

DEFINING PARAMETERS: L, B1, B2, H

END

FEATURE NAME: CLEVIS

ENGLISH DEFI: A U-shaped shackle for connecting a rod and a pin.

GROUP CLASSI: TOPOLOGY GROUP - N.A. FORM GROUP - N.A.

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: CLIP

ENGLISH DEFI: A device that fastens by gripping, clasping or hooking one part to another.

GROUP CLASSI: TOPOLOGY GROUP - C FORM GROUP - B1, B5A, C14

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: COIL

ENGLISH DEFI: A series of concentric rings or spirals.

GROUP CLASSI: TOPOLOGY GROUP - R FORM GROUP - B1, B5B

SYNONYMS: CURL, HELIX, SPIRAL SPRING

DEFINING PARAMETERS: R1, R2, L, N

END

FEATURE NAME: COIN

ENGLISH DEFI: A thin piece of metal, usually circular in shape.

GROUP CLASSI: TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C3

DEFINING PARAMETERS: R, L

END

FEATURE NAME: COLLAR

ENGLISH DEFI: A ring placed around an object to restrict its motion, hold it in place, or cover an opening.

GROUP CLASSI: TOPOLOGY GROUP - C FORM GROUP - B1, B5A

DEFINING PARAMETERS: R1, R2, L

END

FEATURE NAME: COLLET

ENGLISH DEFI: A split, coned sleeve.

GROUP CLASSI: TOPOLOGY GROUP - F FORM GROUP - C10

DEFINING PARAMETERS: L, B1, B2, H

END
FEATURE NAME: COLUMN
ENGLISH DEFI: A vertical shaft designed to bear axial loads.
GROUP CLASSI: TOPOLOGY GROUP - A, E
SYNONYMS: PILLAR
RELATIONS: DEFINING PARAMETERS: R, L
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: COMPARTMENT
ENGLISH DEFI: A part or sub-division of an enclosed space.
GROUP CLASSI: TOPOLOGY GROUP - G
SYNONYMS: CELL, CHAMBER
RELATIONS: DEFINING PARAMETERS: L, B, H, T
END

FEATURE NAME: CONE
ENGLISH DEFI: A solid having a circle as its base and tapering evenly on all its surfaces to a point.
GROUP CLASSI: TOPOLOGY GROUP - V
SYNONYMS: RELATIONS: DEFINING PARAMETERS: R1, R2, L, A.R.
END

FEATURE NAME: CONIC
ENGLISH DEFI: An object with a shape of a cone.
GROUP CLASSI: TOPOLOGY GROUP - V
SYNONYMS: CONICAL
RELATIONS: DEFINING PARAMETERS: R1, R2, L, A.R.
END

FEATURE NAME: CONSTRICTION
ENGLISH DEFI: Narrowing of a channel or cylindrical member.
GROUP CLASSI: TOPOLOGY GROUP - B
SYNONYMS: RELATIONS: DEFINING PARAMETERS: R, L
END

FEATURE NAME: CONVEX
ENGLISH DEFI: Curving outward; as the exterior of a globe.
GROUP CLASSI: TOPOLOGY GROUP - J
SYNONYMS: RELATIONS: DEFINING PARAMETERS: A, B
END

FEATURE NAME: CORD
ENGLISH DEFI: A string or small rope; twine.
GROUP CLASSI: TOPOLOGY GROUP - A
SYNONYMS: ROPE, STRING, TWINE
RELATIONS: DEFINING PARAMETERS: R, L
END

FEATURE NAME: CORE
ENGLISH DEFI: The central or innermost part of a ring, similar to a rod or shaft.
GROUP CLASSI: TOPOLOGY GROUP - A, E
END

Appendix A - Feature Dictionary
SYNONYMS: BAR, ROD

RELATIONS:  

DEFINING PARAMETERS: R, L

DEFINING PARAMETERS: L, B1, B2, H

FEATURE NAME: CORNER

ENGLISH DEF: The point or edge formed due to the meeting of two lines or two surfaces.

GROUP CLASS: TOPOLOGY GROUP - H  FORM GROUP - B2, B3, C4

SYNONYMS: ANGLE, BEND, BRACKET, ELBOW

RELATIONS:  

DEFINING PARAMETERS: L, B, H, T, R, A

FEATURE NAME: COTTER

ENGLISH DEF: A conical key, wedge or pin that is split lengthwise.

GROUP CLASS: TOPOLOGY GROUP - V  FORM GROUP - B1, B5A, C4

SYNONYMS: KEY, WEDGE

RELATIONS:  

DEFINING PARAMETERS: R1, R2, L, A.R.

FEATURE NAME: COUNTER-BORE

ENGLISH DEF: A depression created by cutting the edge of a hole to allow placing of a component.

GROUP CLASS: TOPOLOGY GROUP - N.A.  FORM GROUP - B1, C11, C13

SYNONYMS:  

RELATIONS:  

DEFINING PARAMETERS: R1, R2, L1, L2

FEATURE NAME: COUNTER-SINK

ENGLISH DEF: A depression created by cutting the edge of a hole to allow placing of a component.

GROUP CLASS: TOPOLOGY GROUP - N.A.  FORM GROUP - B1, C11, C13

SYNONYMS:  

RELATIONS:  

DEFINING PARAMETERS: R1, R2, L1, L2

FEATURE NAME: COVER

ENGLISH DEF: A plate whose primary function is to protect or conceal.

GROUP CLASS: TOPOLOGY GROUP - A, E  FORM GROUP - B1, B2, B5A, C3

GROUP CLASS: TOPOLOGY GROUP - FORM GROUP - C12

SYNONYMS: BOLSTER-PLATE, BUTTON, CAP, DIAL, DISK, HATCH, PLATE

RELATIONS:  

DEFINING PARAMETERS: R, L

DEFINING PARAMETERS: L, B1, B2, H

FEATURE NAME: CRACK

ENGLISH DEF: A fissure; to break without separation of parts.

GROUP CLASS: TOPOLOGY GROUP - F  FORM GROUP - C10

SYNONYMS: BREAK, CREVICE, CUT, FISSURE, FRACTURE, RUPTURE

RELATIONS:  

DEFINING PARAMETERS: L, B1, B2, H

FEATURE NAME: CRATER

ENGLISH DEF: A large bowl shaped topographic depression with steep sides.

GROUP CLASS: TOPOLOGY GROUP - L  FORM GROUP - B1, C2, C8

SYNONYMS:  

RELATIONS:  

DEFINING PARAMETERS: A, B

FEATURE NAME: CRESCENT

ENGLISH DEF: Any part that has one convex and one concave edge.

GROUP CLASS: TOPOLOGY GROUP - D  FORM GROUP - B3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CREVICE
ENGLISH DEFI : A fissure or a crack; cleft.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - C11
SYNONYMS : BREAK, CRACK, CUT, FISSURE, FRACTURE, RUPTURE
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CROWN
ENGLISH DEFI : The top dome of a surface.
GROUP CLASSI : TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CUBE
ENGLISH DEFI : A solid bounded by six equal squares and having all its angles right angles.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CUP
ENGLISH DEFI : A small, open vessel often with a handle.
GROUP CLASSI : TOPOLOGY GROUP - O FORM GROUP - B1, C2
SYNONYMS : LADLE, SCOOP
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CURL
ENGLISH DEFI : To twist into ringlets or curves to form a spiral shape.
GROUP CLASSI : TOPOLOGY GROUP - R FORM GROUP - B1, B5B
SYNONYMS : COIL, HELIX, SPIRAL, SPRING
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CURVE
ENGLISH DEFI : A line continuously bent, as an arc of a circle.
GROUP CLASSI : TOPOLOGY GROUP - D FORM GROUP - B3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CUSP
ENGLISH DEFI : A pointed or rounded projection.
GROUP CLASSI : TOPOLOGY GROUP - J FORM GROUP - B1, C1
SYNONYMS
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CUT
ENGLISH DEFI : An opening created due to penetration with a sharp edge; gash; pierce.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS : BREAK, CRACK, CREVICE, FISSURE, FRACTURE, RUPTURE
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : CYLINDER
ENGLISH DEF! 1: A solid bounded by a cylindrical surface and two parallel planes.

GROUP CLASSI 2: TOPOLOGY GROUP - A
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DEEP
ENGLISH DEF! 1: Having a depth or dimension.

GROUP CLASSI 2: TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DEFLATION
ENGLISH DEF! 1: To reduce or restrict.

GROUP CLASSI 2: TOPOLOGY GROUP - L
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DENT
ENGLISH DEF! 1: A small depression made by striking or pressing.

GROUP CLASSI 2: TOPOLOGY GROUP - B, C2, C8
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DEPRESSION
ENGLISH DEF! 1: A low or depressed place or surface.

GROUP CLASSI 2: TOPOLOGY GROUP - L
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DEPTH
ENGLISH DEF! 1: The state or degree of being deep.

GROUP CLASSI 2: TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DIAL
ENGLISH DEF! 1: Any graduated circular plate or face or a rotating disk.

GROUP CLASSI 2: TOPOLOGY GROUP - A, B
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DIAPHRAGM
ENGLISH DEF! 1: A thin sheet placed between parallel parts of a member to increase its rigidity.

GROUP CLASSI 2: TOPOLOGY GROUP - A, E
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: DILATION
ENGLISH DEF! 1: To widen or enlarge.
DEFINING PARAMETERS : R, L
END
FEATURE NAME : DIP
ENGLISH DEFI : A hollow or depression.
GROUP CLASSI : TOPOLOGY GROUP - M FORM GROUP - C2, C8
SYNONYMS : RELATIONS : DEFINING PARAMETERS : END
FEATURE NAME : DISH
ENGLISH DEFI : An open, concave, usually shallow container.
ENGLISH DEFI : A hollow or depression.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : RELATIONS : DEFINING PARAMETERS : END
FEATURE NAME : DISK
ENGLISH DEFI : A fairly flat circular plate.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : BOLSTER-PLATE, BUTTON, CAP, COVER, DIAL, HATCH, PLATE
RELATIONS : DEFINING PARAMETERS : R, L
END
FEATURE NAME : DITCH
ENGLISH DEFI : A long narrow trench dug in the ground.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS : RELATIONS : DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : DOME
ENGLISH DEFI : A roof resembling an inverted cup or hemisphere.
GROUP CLASSI : TOPOLOGY GROUP - O FORM GROUP - B1, C1, C2
SYNONYMS : HEMISPHERE
RELATIONS : DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : DONUT
ENGLISH DEFI : A small cake having a hole in the center.
GROUP CLASSI : TOPOLOGY GROUP - S FORM GROUP - B1, B5A
SYNONYMS : DOUGHNUT
RELATIONS : DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : DOUGHNUT
ENGLISH DEFI : A small cake having a hole in the center.
GROUP CLASSI : TOPOLOGY GROUP - S FORM GROUP - B1, B5A
SYNONYMS : DOUGHNUT
RELATIONS : DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : DOWEL
ENGLISH DEFI : A headless cylindrical pin or peg fitted tightly into adjacent holes of two pieces so as to hold them together.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C4
SYNONYMS : NAIL, NEEDLE, PEG, PIN, STUD, TACK
RELATIONS : DEFINING PARAMETERS : R, L
END
FEATURE NAME : DRIFT-PIN
ENGLISH DEFI : A tapered steel pin used to bring rivet holes fair in assembling steel work.
GROUP CLASS: TOPOLOGY GROUP - A FORM GROUP - B1, C4
SYNONYMS
DEFINING PARAMETERS
END
FEATURE NAME: DRILL
ENGLISH DEF: A tool used for boring holes in hard substances.
GROUP CLASS: TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS: AIGUILLE
DEFINING PARAMETERS: R, L
END
FEATURE NAME: DROP
ENGLISH DEF: Something resembling a drop in shape and size.
GROUP CLASS: TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS: BUBBLE, DROPLET, GLOBULE, TEAR
DEFINING PARAMETERS
END
FEATURE NAME: DROPLET
ENGLISH DEF: Something resembling a drop in shape and size.
GROUP CLASS: TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS: BUBBLE, DROP, GLOBULE, TEAR
DEFINING PARAMETERS: R
END
FEATURE NAME: DRUM
ENGLISH DEF: A hollow cylindrical container.
GROUP CLASS: TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS
DEFINING PARAMETERS: R1, R2, L
END
FEATURE NAME: DUCT
ENGLISH DEF: Any tube, canal or passage through which a fluid is conveyed.
GROUP CLASS: TOPOLOGY GROUP - C FORM GROUP - B2B
SYNONYMS: FLUE, GLAND, PIPE, TUBE
DEFINING PARAMETERS: R1, R2, L
END
FEATURE NAME: DUNE
ENGLISH DEF: A hill of loose sand.
GROUP CLASS: TOPOLOGY GROUP - J FORM GROUP - B1, B5A, C1
SYNONYMS
DEFINING PARAMETERS: A, B
END
FEATURE NAME: EDGE
ENGLISH DEF: A bounding or dividing curve.
GROUP CLASS: TOPOLOGY GROUP - H FORM GROUP - B4
SYNONYMS: LINE
DEFINING PARAMETERS: L, B, H, T, R, A
END
FEATURE NAME: ELBOW
ENGLISH DEF: A joint at the bend of an object.
GROUP CLASS: TOPOLOGY GROUP - H FORM GROUP - B2
SYNONYMS: ANGLE, BEND, BRACKET, CORNER
DEFINING PARAMETERS: L, B, H, T, R, A
END
FEATURE NAME: ELLIPSE
ENGLISH DEF
Appendix A - Feature Dictionary
139
ENGLISH DEF1 : An oval shaped curve; a conic section.
GROUP CLASSI : TOPOLOGY GROUP - W FORM GROUP - B1, B5A, C1
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : ELLIPSOID
ENGLISH DEFI : A solid of which every plane surface is a ellipse or a
ENGLISH DEFI : circle.
GROUP CLASSI : TOPOLOGY GROUP - W FORM GROUP - B1, B5A, C1
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : EMBOSS
ENGLISH DEFI : A raised pattern on a surface produced by means of a die.
GROUP CLASSI : TOPOLOGY GROUP - J FORM GROUP - C1
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : A, B
END
FEATURE NAME : ENTASIS
ENGLISH DEFI : A slight swelling.
GROUP CLASSI : TOPOLOGY GROUP - J FORM GROUP - C1
SYNONYMS : BUMP, BULGE, SWELL, SWELLING
RELATIONS : 
DEFINING PARAMETERS : A, B
END
FEATURE NAME : FACE
ENGLISH DEFI : The front or principal surface of any object.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : PLANE, SHEET
RELATIONS : 
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : FENCE
ENGLISH DEFI : A structure of rails, stakes, strung wire, etc., erected
ENGLISH DEFI : as an enclosure, barrier or boundary.
GROUP CLASSI : TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B, DL, DB
END
FEATURE NAME : FILLERS
ENGLISH DEFI : Either plate or ring fills used to take up space.
GROUP CLASSI : TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
END
DEFINING PARAMETERS : R1, R2, L
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : FILLET
ENGLISH DEFI : A rounded filling of the internal angle between two
ENGLISH DEFI : surfaces.
GROUP CLASSI : TOPOLOGY GROUP - N.A. FORM GROUP - B5A
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
END
FEATURE NAME : FILLING
ENGLISH DEFI : Material used to fill cavities.
GROUP CLASSI : TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
END

Appendix A - Feature Dictionary 140
SYNONYMS
RELATIONS
DEFINING PARAMETERS
R, L
DEFINING PARAMETERS
R1, R2, L
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FILM
ENGLISH DEFINITION
A thin covering or layer.
GROUP CLASSIFICATION
TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS
FOIL, FOLD, LEAF, MEMBRANE
RELATIONS
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FILTER
ENGLISH DEFINITION
A porous article or material for separating matter.
GROUP CLASSIFICATION
TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
R, L
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FIN
ENGLISH DEFINITION
A projecting plate, structure, appendage or attachment.
GROUP CLASSIFICATION
TOPOLOGY GROUP - A, E FORM GROUP - B2, C3
SYNONYMS
CLEAT, RIB
RELATIONS
DEFINING PARAMETERS
R, L
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FISSURE
ENGLISH DEFINITION
A long narrow opening, cleft or furrow.
GROUP CLASSIFICATION
TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS
BREAK, CRACK, CREVICE, CUT, FRACTURE, RUPTURE
RELATIONS
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FLAKE
ENGLISH DEFINITION
A small, thin piece peeled or split off from a surface.
GROUP CLASSIFICATION
TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS
L, B1, B2, H
END
FEATURE NAME
FLANGE
ENGLISH DEFINITION
A projecting rim or a collar of a mechanical part design
to aid attachment or to increase stiffness.
GROUP CLASSIFICATION
TOPOLOGY GROUP - N.A. FORM GROUP - B1, B5A, C11
SYNONYMS
RELATIONS
DEFINING PARAMETERS
L1, L2, B1, B2, R, T
END
FEATURE NAME
FLUE
ENGLISH DEFINITION
A pipe or a tube through which smoke, hot air, steam is
drawn off.
GROUP CLASSIFICATION
TOPOLOGY GROUP - C FORM GROUP - B1, B2, B5A
SYNONYMS
DUCT, GLAND, PIPE, TUBE
RELATIONS
DEFINING PARAMETERS
R1, R2, L
END
FEATURE NAME
FLUME
ENGLISH DEFINITION
A chute or a through.
GROUP CLASSIFICATION
TOPOLOGY GROUP - F FORM GROUP - C3

Appendix A - Feature Dictionary
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FLUTE
ENGLISH DEFI A groove having a curved section, especially when parallel to the main axis.
GROUP CLASSI TOPOLOGY GROUP - N FORM GROUP - C10
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FOIL
ENGLISH DEFI A metal hammered or rolled into thin, pliant sheets.
GROUP CLASSI TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FOLD
ENGLISH DEFI To wrap up or enclose with a thin covering.
GROUP CLASSI TOPOLOGY GROUP - E FORM GROUP - C9
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FRACTURE
ENGLISH DEFI A break, crack or rupture.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FRAME
ENGLISH DEFI A case or border made to enclose something.
GROUP CLASSI TOPOLOGY GROUP - G FORM GROUP - B2B
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME FURROW
ENGLISH DEFI Any long, narrow, deep depression as a groove, rut or deep wrinkle.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME GAP
ENGLISH DEFI An opening, wide crack or deep notch.
GROUP CLASSI TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME GASKET
ENGLISH DEFI A packing, usually in form of a sheet or ring to make a joint leakproof.
GROUP CLASSI TOPOLOGY GROUP - C FORM GROUP - B1, B5A, C14
SYNONYMS RELATIONS
DEFINING PARAMETERS END
FEATURE NAME: GAUZE
ENGLISH DEF: Any tiny, open-mesh material.
GROUP CLASSI: TOPOLOGY GROUP - P
SYNONYMS: L, B, DL, DB
END

FEATURE NAME: GIB
ENGLISH DEF: A plate of metal or other material machined to hold other parts in place.
GROUP CLASSI: TOPOLOGY GROUP - E
SYNONYMS: L, B1, B2, H
END

FEATURE NAME: GLAND
ENGLISH DEF: A long cylindrical body with a hollow center, generally used for conveying something.
GROUP CLASSI: TOPOLOGY GROUP - C
SYNONYMS: DUCT, FLUE, PIPE, TUBE
END

FEATURE NAME: GLOBE
ENGLISH DEF: A sphere.
GROUP CLASSI: TOPOLOGY GROUP - I
SYNONYMS: BALL, SPHERE
END

FEATURE NAME: GLOBULE
ENGLISH DEF: A tiny sphere or drop.
GROUP CLASSI: TOPOLOGY GROUP - I
SYNONYMS: BUBBLE, DROP, DROPLET, TEAR
END

FEATURE NAME: GORGE
ENGLISH DEF: A narrow deep ravine.
GROUP CLASSI: TOPOLOGY GROUP - F
END

FEATURE NAME: GRAIN
ENGLISH DEF: A small rounded prominence.
GROUP CLASSI: TOPOLOGY GROUP - J
END

FEATURE NAME: GRID
ENGLISH DEF: An arrangement of regularly placed parallel or intersecting bars.
GROUP CLASSI: TOPOLOGY GROUP - P
SYNONYMS: L, B, DL, DB
END

FEATURE NAME: GRILL
ENGLISH DEF: A grating used as a screen.
GROUP CLASSI: TOPOLOGY GROUP - P
END
RELATIONS
DEFINING PARAMETERS : L, B, DL, DB
END

FEATURE NAME : GROOVE
ENGLISH DEFI : A long narrow indentation or furrow cut into a surface;
ENGLISH DEFI : Any narrow depression, channel or rut.
GROUP CLASSI : TOPOLOGY GROUP - F  FORM GROUP - C10
SYNONYMS : CAVITY, CHANNEL, FURROW, RUT, SLOT
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : GUIDE
ENGLISH DEFI : Any device that regulates or controls the operations of a
part.
GROUP CLASSI : TOPOLOGY GROUP - E, F  FORM GROUP - B4
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : GUSSET
ENGLISH DEFI : A plate used to connect various members.
GROUP CLASSI : TOPOLOGY GROUP - E  FORM GROUP - B1, B2, C3
SYNONYMS :  
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : GUTTER
ENGLISH DEFI : Any groove or channel.
GROUP CLASSI : TOPOLOGY GROUP - F  FORM GROUP - B2, C2, C10
SYNONYMS :  
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : HAND-WHEEL
ENGLISH DEFI : A wheel that is designed especially to be grasped by
hand.
GROUP CLASSI : TOPOLOGY GROUP - A  FORM GROUP - B1, C12
SYNONYMS :  
DEFINING PARAMETERS : R, L
END

FEATURE NAME : HATCH
ENGLISH DEFI : A cover over an opening in a floor or plane.
GROUP CLASSI : TOPOLOGY GROUP - A, E  FORM GROUP - B1, C12
SYNONYMS : BOLSTER-PLATE, BUTTON, CAP, COVER, DIAL, DISK, PLATE
RELATIONS : R, L
DEFINING PARAMETERS : L, B1, B2, H
END

FEATURE NAME : HELIX
ENGLISH DEFI : A line thread or wire curved as if wound in a single layer
GROUP CLASSI : TOPOLOGY GROUP - R  FORM GROUP - B1, B5B
SYNONYMS : COIL, CURL, SPIRAL, SPRING
RELATIONS : R1, R2, L, N
DEFINING PARAMETERS :  
END

FEATURE NAME : HEM
ENGLISH DEFI : To shut in, enclose or restrict by folding in.
GROUP CLASSI : TOPOLOGY GROUP - N.A.  FORM GROUP - C12
SYNONYMS :  
RELATIONS :  
END
FEATURE NAME: HEMISPHERE
ENGLISH DEFINITION: A half-sphere formed by a plane passing through the center of the sphere.
SYNONYMS: TOPOLOGY GROUP - O FORM GROUP - B1, C1, C2
DEFINING PARAMETERS: R1, R2
RELATIONS:
END

FEATURE NAME: HEX-BOLT
ENGLISH DEFINITION: A pin or rod for holding something in place, usually having a head on one end and threading on the other.
SYNONYMS: TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
DEFINING PARAMETERS: R1, R2, L, T
RELATIONS:
END

FEATURE NAME: HEX-NUT
ENGLISH DEFINITION: A small block of metal having an internal thread so that it can be fitted with a bolt or screw.
SYNONYMS: TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
DEFINING PARAMETERS: R1, R2, T
RELATIONS:
END

FEATURE NAME: HOLE
ENGLISH DEFINITION: A cavity or aperture in a solid mass or body. Can be blind or through.
SYNONYMS: TOPOLOGY GROUP - B, F FORM GROUP - B1, B5A, C11
DEFINING PARAMETERS: R, L
DEFINING PARAMETERS: L, B1, B2, H
RELATIONS:
END

FEATURE NAME: HOLLOW
ENGLISH DEFINITION: A cavity, valley or empty space in anything.
SYNONYMS: TOPOLOGY GROUP - F FORM GROUP - C2
RELATIONS:
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: HOOD
ENGLISH DEFINITION: Anything resembling a hood in form or use.
SYNONYMS: TOPOLOGY GROUP - E FORM GROUP - N.A.
DEFINING PARAMETERS: L, B1, B2, H
RELATIONS:
END

FEATURE NAME: HUB
ENGLISH DEFINITION: The central part of a wheel, pulley or propeller attached to the driving member.
SYNONYMS: TOPOLOGY GROUP - V FORM GROUP - B1, B5A, C4
RELATIONS:
DEFINING PARAMETERS: R1, R2, L, A.R.
END

FEATURE NAME: HUMP
ENGLISH DEFINITION: A rounded protuberance.
SYNONYMS: TOPOLOGY GROUP - J FORM GROUP - B1, C1
RELATIONS: PROMINENCE, PROTUBERANCE, PROTRUSION
DEFINING PARAMETERS: A, B
END

FEATURE NAME: IMPRESSION

appendix A - feature dictionary
ENGLISH DEFI : A mark made by pressure or engraving.
GROUP CLASSI : TOPOLOGY GROUP - L FORM GROUP - B1, C2
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : INLET
ENGLISH DEFI : A relatively narrow entrance or orifice for admission of fluid.
GROUP CLASSI : TOPOLOGY GROUP - B FORM GROUP - C11
SYNONYMS
APERTURE, OPENING, ORIFICE, PERFORATION, PUNTURE
RELATIONS
DEFINING PARAMETERS
END
FEATURE NAME : JOURNAL
ENGLISH DEFI : The part of shaft or crank which is supported by and turns in a bearing.
GROUP CLASSI : TOPOLOGY GROUP · C FORM GROUP - B1, B5A
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEEL
ENGLISH DEFI : The main structural member of a vessel, running fore and aft along the bottom, to which all the crosswise members are solidly fixed.
GROUP CLASSI : TOPOLOGY GROUP · E FORM GROUP - B2, C3
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KERF
ENGLISH DEFI : A narrow deep cut.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - C10
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEY
ENGLISH DEFI : A wedge, cotter pin used to secure various parts.
GROUP CLASSI : TOPOLOGY GROUP - A, E FORM GROUP - B2, C3
SYNONYMS
COTTER, WEDGE
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYHOLE
ENGLISH DEFI : A hole or slot for receiving a key used to fasten.
GROUP CLASSI : TOPOLOGY GROUP - B, F FORM GROUP - B2, C10
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYSEAT
ENGLISH DEFI : A groove or channel for placing a key in any mechanical part.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - B2, C10
SYNONYMS
KEYWAY
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYWAY
ENGLISH DEFI :
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - B2, C10
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEY
ENGLISH DEFI : A wedge, cotter pin used to secure various parts.
GROUP CLASSI : TOPOLOGY GROUP - A, E FORM GROUP - B2, C3
SYNONYMS
COTTER, WEDGE
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYHOLE
ENGLISH DEFI : A hole or slot for receiving a key used to fasten.
GROUP CLASSI : TOPOLOGY GROUP - B, F FORM GROUP - B2, C10
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYSEAT
ENGLISH DEFI : A groove or channel for placing a key in any mechanical part.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - B2, C10
SYNONYMS
KEYWAY
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEYWAY
ENGLISH DEFI :
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - B2, C10
SYNONYMS
RELATIONS
DEFINING PARAMETERS END
FEATURE NAME : KEY
ENGLISH DEFI: A groove or channel for placing a key in any mechanical part.
GROUP CLASSI: TOPOLOGY GROUP - F
SYNONYMS: KEYSEAT
RELATIONS: DEFINING PARAMETERS: L, B1, B2, H
FEATURE NAME: KNOB
ENGLISH DEFI: A rounded protuberance, bunch or hump.
GROUP CLASSI: TOPOLOGY GROUP - J
SYNONYMS: NODE
RELATIONS: DEFINING PARAMETERS: A, B
FEATURE NAME: KNURL
ENGLISH DEFI: A cup-shaped vessel with a long handle.
GROUP CLASSI: TOPOLOGY GROUP - O
SYNONYMS: CUP, SCOOP
RELATIONS: DEFINING PARAMETERS: R1, R2
FEATURE NAME: LADLE
ENGLISH DEFI: Metal in form of a very thin sheet or plate.
GROUP CLASSI: TOPOLOGY GROUP - E
SYNONYMS: FILM, FOIL, FOLD, MEMBRANE
RELATIONS: DEFINING PARAMETERS: L, B1, B2, H
FEATURE NAME: LEAF
ENGLISH DEFI: A raised edge or a narrow shelf-like projection.
GROUP CLASSI: TOPOLOGY GROUP - J
SYNONYMS: RELATIONS: DEFINING PARAMETERS: A, B
FEATURE NAME: LEDGE
ENGLISH DEFI: A support resembling a leg in shape, position or function.
GROUP CLASSI: TOPOLOGY GROUP - A, E
SYNONYMS: RELATIONS: DEFINING PARAMETERS: R, L
DEFINING PARAMETERS: L, B1, B2, H
FEATURE NAME: LEG
ENGLISH DEFI: A slender continuous mark or indentation.
GROUP CLASSI: TOPOLOGY GROUP - H
SYNONYMS: EDGE
RELATIONS: DEFINING PARAMETERS: L, B, H, T, R, A
FEATURE NAME: LINE
ENGLISH DEFI: A marginal part or structure resembling a flared edge.
GROUP CLASSI: TOPOLOGY GROUP - C
SYNONYMS: RELATIONS: DEFINING PARAMETERS: R1, R2, L
FEATURE NAME: LIP
Appendix A · Feature Dictionary
FEATURE NAME: LOBE
ENGLISH DEFINITION: A rounded division, protuberance or part.
GROUP CLASS: TOPOLOGY GROUP - J
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: A, B
END

FEATURE NAME: LOOP
ENGLISH DEFINITION: A ring or bent piece of metal serving as a fastener.
GROUP CLASS: TOPOLOGY GROUP - C
SYNONYMS: RIM, RING
RELATIONS: 
DEFINING PARAMETERS: R1, R2, L
END

FEATURE NAME: LUG
ENGLISH DEFINITION: A projecting "ear", usually rectangular in cross section.
GROUP CLASS: TOPOLOGY GROUP - E
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: MANDREL
ENGLISH DEFINITION: A shaft inserted through a hole in a component to support the work during machining.
GROUP CLASS: TOPOLOGY GROUP - A
SYNONYMS: AXLE, ARBOR, SHAFT, SPINDLE
RELATIONS: 
DEFINING PARAMETERS: R, L
END

FEATURE NAME: MANIFOLD
ENGLISH DEFINITION: A branch pipe arrangement having several or many openings.
GROUP CLASS: TOPOLOGY GROUP - C
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: R1, R2, L
END

FEATURE NAME: MASK
ENGLISH DEFINITION: A covering used to conceal all or part of an object.
GROUP CLASS: TOPOLOGY GROUP - E
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: MASS
ENGLISH DEFINITION: An object having no specific shape but a relatively large size.
GROUP CLASS: TOPOLOGY GROUP - I
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: R
END

FEATURE NAME: MAT
ENGLISH DEFINITION: A small, flat piece of material sometimes used for protection or covering.
GROUP CLASS: TOPOLOGY GROUP - E
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: MEMBRANE
ENGLISH DEFINITION: A thin, pliable layer of tissue serving to cover or line an object or part.
GROUP CLASS: TOPOLOGY GROUP - E
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: L, B1, B2, BSA, C3
END
SYNONYMS: FILM, FOIL, FOLD, LEAF
RELATIONS: L, B1, B2, H
DEFINING PARAMETERS: L, B1, B2, H
END

FEATURE NAME: MESH
ENGLISH DEF: A net, nest or screen
GROUP CLASS: TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS: NET, NEST, SCREEN
RELATIONS: 
DEFINING PARAMETERS: L, B, DL, DB
END

FEATURE NAME: MESSENER
ENGLISH DEF: A small cylindrical metal weight.
GROUP CLASS: TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: R, L
END

FEATURE NAME: MOSAIC
ENGLISH DEF: A design arrangement resembling a network.
GROUP CLASS: TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: L, B, DL, DB
END

FEATURE NAME: NAIL
ENGLISH DEF: A slender, usually pointed, fastener with a head designed for insertion by impact.
ENGLISH DEF: A slender, pointed instrument usually of steel used to transmit vibration.
GROUP CLASS: TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C4
SYNONYMS: DOWEL, NEEDLE, PEG, PIN, STUD, TACK
RELATIONS: 
DEFINING PARAMETERS: R, L
END

FEATURE NAME: NECK
ENGLISH DEF: A groove around a shaft, usually near the end.
GROUP CLASS: TOPOLOGY GROUP - N.A. FORM GROUP - B1, B5A, C1
SYNONYMS: 
RELATIONS: 
DEFINING PARAMETERS: R1, R2, R3, L1, L2, L3
END

FEATURE NAME: NEEDLE
ENGLISH DEF: A slender, pointed instrument usually of steel used to transmit vibration.
GROUP CLASS: TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C4
SYNONYMS: DOWEL, NEEDLE, PEG, PIN, STUD, TACK
RELATIONS: 
DEFINING PARAMETERS: R, L
END

FEATURE NAME: NEST
ENGLISH DEF: Something constructed with meshes.
GROUP CLASS: TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS: MESH, NET, SCREEN
RELATIONS: 
DEFINING PARAMETERS: L, B, DL, DB
END

FEATURE NAME: NET
ENGLISH DEF: Something constructed with meshes.
GROUP CLASS: TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS: MESH, NET, SCREEN
RELATIONS: 
DEFINING PARAMETERS: L, B, DL, DB
END

FEATURE NAME: NILE
ENGLISH DEFI: A small projecting point.
GROUP CLASSI: TOPOLOGY GROUP - J
SYNONYMS:
DEFINING PARAMETERS: A, B
END
FEATURE NAME: NIPPLE
ENGLISH DEFI: A short piece of tubing, usually with internal/external threading used to couple pipes.
GROUP CLASSI: TOPOLOGY GROUP - C
SYNONYMS:
DEFINING PARAMETERS: R1, R2, L
END
FEATURE NAME: NODE
ENGLISH DEFI: A knob, protuberance or swelling.
GROUP CLASSI: TOPOLOGY GROUP - J
SYNONYMS:
DEFINING PARAMETERS: A, B
END
FEATURE NAME: NOTCH
ENGLISH DEFI: A V-shaped cut in a surface or a rectangular depression.
GROUP CLASSI: TOPOLOGY GROUP - F, M
SYNONYMS:
DEFINING PARAMETERS: L, B1, B2, H, R, H, A.R.
END
FEATURE NAME: NOZZLE
ENGLISH DEFI: A tube-like device, usually stream-lined for acceleration and directing fluid flow.
GROUP CLASSI: TOPOLOGY GROUP - C
SYNONYMS:
DEFINING PARAMETERS: R1, R2, L
END
FEATURE NAME: O-RING
ENGLISH DEFI: A circular object with functions similar to that of a gasket.
GROUP CLASSI: TOPOLOGY GROUP - C
SYNONYMS:
DEFINING PARAMETERS: R1, R2, L
END
FEATURE NAME: OPENING
ENGLISH DEFI: A widening of a crevice or an open space.
GROUP CLASSI: TOPOLOGY GROUP - B
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: R, L
END
FEATURE NAME: ORIFICE
ENGLISH DEFI: An opening into a cavity or aperture.
GROUP CLASSI: TOPOLOGY GROUP - B
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: R, L
END
FEATURE NAME: OVAL
ENGLISH DEFI: Having a shape of an egg or resembling an ellipsoid.
GROUP CLASSI: TOPOLOGY GROUP - W
SYNONYMS:
END
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R1, R2 |
| **END** | : |  |
| **FEATURE NAME** | : | PACKING |
| **ENGLISH DEFI** | : | Any material used in packing, closing a joint. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3 |
| **SYNONYMS** | : | PAD, PACKING, PADDING |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R, L |
| **DEFINING PARAMETERS** | : | R1, R2, L |
| **DEFINING PARAMETERS** | : | L, B1, B2, H |
| **END** | : |  |
| **FEATURE NAME** | : | PAD |
| **ENGLISH DEFI** | : | A layer of material used as a cushion or for protection. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3 |
| **SYNONYMS** | : | PACKING, PAD, PADDING |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R, L |
| **DEFINING PARAMETERS** | : | R1, R2, L |
| **DEFINING PARAMETERS** | : | L, B1, B2, H |
| **END** | : |  |
| **FEATURE NAME** | : | PADDING |
| **ENGLISH DEFI** | : | A layer of material used as a cushion or for protection. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3 |
| **SYNONYMS** | : | PACKING, PAD, PADDING |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R, L |
| **DEFINING PARAMETERS** | : | R1, R2, L |
| **DEFINING PARAMETERS** | : | L, B1, B2, H |
| **END** | : |  |
| **FEATURE NAME** | : | PAIL |
| **ENGLISH DEFI** | : | A cylindrical or slightly tapered container. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - B FORM GROUP - B1, B5A |
| **SYNONYMS** | : |  |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R1, R2, L |
| **END** | : |  |
| **FEATURE NAME** | : | PAILLET |
| **ENGLISH DEFI** | : | A slender flat piece of wood. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - E FORM GROUP - B1, B2, B5A, C3 |
| **SYNONYMS** | : |  |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | L, B1, B2, H |
| **END** | : |  |
| **FEATURE NAME** | : | PAN |
| **ENGLISH DEFI** | : | A wide, shallow receptacle. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - B FORM GROUP - B1, B2, B5A, C3 |
| **SYNONYMS** | : |  |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | R, L |
| **END** | : |  |
| **FEATURE NAME** | : | PARTING |
| **ENGLISH DEFI** | : | The act of separating or dividing. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - F FORM GROUP - C10 |
| **SYNONYMS** | : |  |
| **RELATIONS** | : |  |
| **DEFINING PARAMETERS** | : | L, B1, B2, H |
| **END** | : |  |
| **FEATURE NAME** | : | PARTITION |
| **ENGLISH DEFI** | : | Something that divides or separates. |
| **GROUP CLASSI** | : | TOPOLOGY GROUP - E FORM GROUP - B2, C3 |
| **SYNONYMS** | : |  |
| **RELATIONS** | : |  |
DEFINING PARAMETERS : L, B, DL, DB
END
FEATURE NAME : PATTERN
ENGLISH DEFI : A design arrangement resembling a network.
GROUP CLASSI : TOPOLOGY GROUP - P FORM GROUP - C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : L, B, DL, DB
END
FEATURE NAME : PEAK
ENGLISH DEFI : The highest point or apex.
GROUP CLASSI : TOPOLOGY GROUP - K FORM GROUP - B1, C4
SYNONYMS : APEX, VERTEX
RELATIONS : 
DEFINING PARAMETERS : R, H, A.R.
END
FEATURE NAME : PEEN
ENGLISH DEFI : The end of a hammer head, usually shaped like a wedge.
GROUP CLASSI : TOPOLOGY GROUP - N.A. FORM GROUP - B1, C1
SYNONYMS : 
RELATIONS : 
END
FEATURE NAME : PEG
ENGLISH DEFI : A small pointed or tapered piece to fasten parts.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C4
SYNONYMS : DOWEL, NAIL, NEEDLE, PIN, STUD, TACK
RELATIONS : 
DEFINING PARAMETERS : R, L
END
FEATURE NAME : PELLET
ENGLISH DEFI : A small round cylinder or ball.
GROUP CLASSI : TOPOLOGY GROUP - A, I FORM GROUP - B1, B2, B5A, C3
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R, L
DEFINING PARAMETERS : R
END
FEATURE NAME : PERFORATION
ENGLISH DEFI : A tiny hole created by drilling or piercing operation.
GROUP CLASSI : TOPOLOGY GROUP - B FORM GROUP - C11
SYNONYMS : APERTURE, INLET, OPENING, ORIFICE, PUNCTURE
RELATIONS : 
DEFINING PARAMETERS : R, L
END
FEATURE NAME : PERIMETER
ENGLISH DEFI : The boundary of any two dimensional object.
GROUP CLASSI : TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS : 
RELATIONS : 
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : PILLAR
ENGLISH DEFI : A vertical shaft designed to bear axial loads.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : COLUMN

RELATIONS
DEFINING PARAMETERS :  R, L
END

FEATURE NAME :  PIN
ENGLISH DEF :  A cylindrical fastener used to join two members or parts
with freedom of angular movement at the joint.
GROUP CLASSI :  TOPOLOGY GROUP - A
SYNONYMS :  DOWEL, NAIL, NEEDLE, PEG, STUD, TACK
RELATIONS
DEFINING PARAMETERS :  R, L
END

RELATIONS
DEFINING PARAMETERS :  R1, R2, L
END

FEATURE NAME :  PIPE
ENGLISH DEF :  A cylindrical tube or a long conducting passage.
GROUP CLASSI :  TOPOLOGY GROUP - C
SYNONYMS :  DUCT, FLUE, GLAND, TUBE
RELATIONS
DEFINING PARAMETERS :  R1, R2, L
END

FEATURE NAME :  PIT
ENGLISH DEF :  A wide and deep cavity.
GROUP CLASSI :  TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS :  L, B1, B2, H
END

FEATURE NAME :  PLANE
ENGLISH DEF :  A surface such that a straight line joining any of its
two points wholly lies on the surface.
GROUP CLASSI :  TOPOLOGY GROUP - E
SYNONYMS :  FACE, SHEET
RELATIONS
DEFINING PARAMETERS :  L, B1, B2, H
END

FEATURE NAME :  PLATE
ENGLISH DEF :  A flat piece of metal of arbitrary minimum thickness and
width depending on type of metal.
GROUP CLASSI :  TOPOLOGY GROUP - A, E
SYNONYMS :  BOLSTER-PLATE, BUTTON, CAP, COVER, DIAL, DISK, HATCH
RELATIONS
DEFINING PARAMETERS :  R, L
DEFINING PARAMETERS :  L, B1, B2, H
END

FEATURE NAME :  PLUG
ENGLISH DEF :  A piece of cylindrical metal used to stop or close a
hole.
GROUP CLASSI :  TOPOLOGY GROUP - A
SYNONYMS
RELATIONS
DEFINING PARAMETERS :  R, L
END

FEATURE NAME :  PLUNGER
ENGLISH DEF :  A long rod or piston.
GROUP CLASSI :  TOPOLOGY GROUP - A
SYNONYMS
RELATIONS
DEFINING PARAMETERS :  R, L
END

FEATURE NAME :  POCKET
ENGLISH DEF :  Any opening, receptacle or container.
GROUP CLASSI :  TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS :  L, B1, B2, H
END
<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>ENGLISH DEF</th>
<th>GROUP CLASS</th>
<th>TOPOLOGY GROUP</th>
<th>FORM GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>POD</td>
<td>A seed vessel or capsule.</td>
<td>TOPOLOGY GROUP - J</td>
<td>B1, B2, C1, C7</td>
<td></td>
</tr>
<tr>
<td>POINT</td>
<td>Something sharp and tapering such as a point of a needle.</td>
<td>TOPOLOGY GROUP - K</td>
<td>B1, C4</td>
<td></td>
</tr>
<tr>
<td>POT</td>
<td>A round, fairly deep vessel of metal.</td>
<td>TOPOLOGY GROUP - I</td>
<td>B1, B2</td>
<td></td>
</tr>
<tr>
<td>POUCH</td>
<td>A small bag, sack or other container.</td>
<td>TOPOLOGY GROUP - F</td>
<td>C10</td>
<td></td>
</tr>
<tr>
<td>PRISM</td>
<td>A solid whose sides are equal and parallel plane figures and whose lateral faces are parallelograms.</td>
<td>TOPOLOGY GROUP - N.A.</td>
<td>B2, C3</td>
<td></td>
</tr>
<tr>
<td>PROBE</td>
<td>A small tubing containing the sensing element of electronic equipment.</td>
<td>TOPOLOGY GROUP - C</td>
<td>B1, B5A</td>
<td></td>
</tr>
<tr>
<td>PROJECTION</td>
<td>A protrusion.</td>
<td>TOPOLOGY GROUP - J</td>
<td>B1, C1</td>
<td></td>
</tr>
<tr>
<td>PROMINENCE</td>
<td>Jutting out or projecting.</td>
<td>TOPOLOGY GROUP - J</td>
<td>B1, C1</td>
<td></td>
</tr>
<tr>
<td>PROTRUSION</td>
<td>To push or thrust out; to project outward.</td>
<td>TOPOLOGY GROUP - J</td>
<td>B1, C1</td>
<td></td>
</tr>
</tbody>
</table>

Appendix A - Feature Dictionary 154
<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>ENGLISH DEF</th>
<th>GROUP CLASS</th>
<th>SYNONYMS</th>
<th>RELATIONS</th>
<th>DEFINING PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTUBERANCE</td>
<td>Something that protrudes; a knob; prominence.</td>
<td>TOPOLOGY GROUP - J FORM GROUP - B1, C1</td>
<td>HUMP, PROMINENCE, PROTRUSION</td>
<td></td>
<td>A, B</td>
</tr>
<tr>
<td>PULLEY</td>
<td>A wheel grooved to receive a rope to increase mechanical advantage.</td>
<td>TOPOLOGY GROUP - Q FORM GROUP - C2</td>
<td></td>
<td></td>
<td>R1, R2, R3, L1, L2, L3</td>
</tr>
<tr>
<td>PUNCTURE</td>
<td>A small hole made by piercing with a sharp point.</td>
<td>TOPOLOGY GROUP - B FORM GROUP - C11</td>
<td>APERTURE, INLET, OPENING, PERFORATION, PUNCTURE</td>
<td></td>
<td>R, L</td>
</tr>
<tr>
<td>PYRAMID</td>
<td>A solid consisting of a polygonal base and triangular sides with a common vertex.</td>
<td>TOPOLOGY GROUP - N.A. FORM GROUP - B2A, C3</td>
<td>PRISM, TETRAHEDRON</td>
<td></td>
<td>L, B, H, A.R.</td>
</tr>
<tr>
<td>RABBIT</td>
<td>A recess or groove near the edge of one piece of wood, cut so as to receive the edge of another piece.</td>
<td>TOPOLOGY GROUP - F FORM GROUP - B2, C10</td>
<td></td>
<td></td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>RACE</td>
<td>A sluice or channel by which to conduct water.</td>
<td>TOPOLOGY GROUP - D, N FORM GROUP - B1, C2</td>
<td></td>
<td></td>
<td>R1, R2, L</td>
</tr>
<tr>
<td>RAIL</td>
<td>A bar of metal resting on supports.</td>
<td>TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A, C3</td>
<td>TRACK</td>
<td></td>
<td>R, L</td>
</tr>
<tr>
<td>RAISE</td>
<td>Elevated.</td>
<td>TOPOLOGY GROUP - J FORM GROUP - C1</td>
<td></td>
<td></td>
<td>L, B1, B2, H</td>
</tr>
</tbody>
</table>

Appendix A - Feature Dictionary
END
FEATURE NAME : RAVINE
ENGLISH DEFI : A small, narrow valley with steeply sloping sides.
GROUP CLASSI : TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : REAM
ENGLISH DEFI : To enlarge or clear out a hole or to enlarge a taper.
GROUP CLASSI : TOPOLOGY GROUP - B
SYNONYMS
RELATIONS
DEFINING PARAMETERS : R, L
END
FEATURE NAME : RECEPTACLE
ENGLISH DEFI : Anything that serves as to contain or hold anything.
GROUP CLASSI : TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : RECESS
ENGLISH DEFI : A depression or indentation in any surface.
GROUP CLASSI : TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : REEL
ENGLISH DEFI : A rotary frame or device used for winding.
ENGLISH DEFI : A narrow strip of metal with finished edges.
GROUP CLASSI : TOPOLOGY GROUP - Q
SYNONYMS : SPOOL
RELATIONS
DEFINING PARAMETERS : R1, R2, R3, L1, L2, L3
END
FEATURE NAME : RIB
ENGLISH DEFI : A transverse structural member that gives cross section shape and strength to a portion of an object.
GROUP CLASSI : TOPOLOGY GROUP - H
SYNONYMS
RELATIONS
DEFINING PARAMETERS : L, B, H, T, R, A
END
FEATURE NAME : RIBBON
ENGLISH DEFI : A narrow strip of metal with finished edges.
GROUP CLASSI : TOPOLOGY GROUP - E
SYNONYMS : BAND, BELT, STRIP
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : RIDGE
ENGLISH DEFI : An elongated, narrow, steep sided elevation.
GROUP CLASSI : TOPOLOGY GROUP - F
SYNONYMS
RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : RIFLE
ENGLISH DEFI : A bore hole that follows a spiral course.
GROUP CLASSI : TOPOLOGY GROUP - T
SYNONYMS
RELATIONS
DEFINING PARAMETERS : R1, L, T, T2, H, N
END
FEATURE NAME : RIM
ENGLISH DEFI : An outer edge or border, sometimes raised or projecting.
GROUP CLASSI : TOPOLOGY GROUP - C FORM GROUP - B1, BSA
SYNONYMS : LOOP, RING
RELATIONS :
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : RING
ENGLISH DEFI : A tie member of a chain link, usually circular.
GROUP CLASSI : TOPOLOGY GROUP - C, T FORM GROUP - B1, BSA
SYNONYMS : LOOP, RIM
RELATIONS :
DEFINING PARAMETERS : R1, R2, L
DEFINING PARAMETERS : R1, L, T, T2, H, N
END
FEATURE NAME : RIVET
ENGLISH DEFI : A short soft metal bolt, having a head on one end, used to join objects.
GROUP CLASSI : TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
SYNONYMS :
RELATIONS :
DEFINING PARAMETERS :
END
FEATURE NAME : ROD
ENGLISH DEFI : A thin round bar of metal or wood.
ENGLISH DEFI : A straight, slim piece of wood, metal or other material.
GROUP CLASSI : TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, BSA
SYNONYMS : BAR, CORE
RELATIONS :
DEFINING PARAMETERS : R, L
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : ROLLER
ENGLISH DEFI : A cylindrical device for rolling or rotating.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS :
RELATIONS :
DEFINING PARAMETERS : R, L
END
FEATURE NAME : ROPE
ENGLISH DEFI : A construction of twisted fibers, intertwined in strands to form a thick cord.
GROUP CLASSI : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : CORD, STRING, TWINE
RELATIONS :
DEFINING PARAMETERS :
END
FEATURE NAME : ROUND
ENGLISH DEFI : Having a contour that is circular; spherical; ring-shaped cylindrical.
GROUP CLASSI : TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS :
RELATIONS :
DEFINING PARAMETERS :
END
FEATURE NAME : ROUNding
ENGLISH DEFI : Having a contour that is circular; spherical; ring-shaped cylindrical.
GROUP CLASSI : TOPOLOGY GROUP - I FORM GROUP - B1, B5A, C1
SYNONYMS :
RELATIONS :
<table>
<thead>
<tr>
<th>FEATURE NAME</th>
<th>ENGLISH DEF</th>
<th>GROUP CLASS</th>
<th>SYNONYMS</th>
<th>RELATIONS</th>
<th>DEFINING PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUPTURE</td>
<td>To break apart; separate into parts.</td>
<td>TOPOLOGY GROUP - F FORM GROUP - C10</td>
<td>BREAK, CRACK, CREVICE, CUT, FISSURE, FRACTURE</td>
<td></td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>RUT</td>
<td>A sunken track or groove.</td>
<td>TOPOLOGY GROUP - F FORM GROUP - C10</td>
<td>CAVITY, CHANNEL, FURROW, GROOVE, SLOT</td>
<td></td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>SCAR</td>
<td>A mark, damage or lasting effect resulting from an accident.</td>
<td>TOPOLOGY GROUP - B, F FORM GROUP - C10</td>
<td></td>
<td></td>
<td>R, L</td>
</tr>
<tr>
<td>SCOOP</td>
<td>A cup-shaped vessel with a long handle.</td>
<td>TOPOLOGY GROUP - O FORM GROUP - B1, B2, B5A, C2</td>
<td>CUP, LADLE</td>
<td></td>
<td>R1, R2</td>
</tr>
<tr>
<td>SCREEN</td>
<td>A wire mesh or netting forming a partition.</td>
<td>TOPOLOGY GROUP - P FORM GROUP - C3</td>
<td>MESH, NEST, NET</td>
<td></td>
<td>L, B, DL, DB</td>
</tr>
<tr>
<td>SCREW</td>
<td>A cylindrical body with a helical groove cut into its surface.</td>
<td>TOPOLOGY GROUP - U FORM GROUP - B1, B5B, C4</td>
<td></td>
<td></td>
<td>R1, L, T, T2, H, N</td>
</tr>
<tr>
<td>SEAL</td>
<td>A device used to fasten, secure or close.</td>
<td>TOPOLOGY GROUP - A, E FORM GROUP - B1, B2, B5A, C12</td>
<td></td>
<td></td>
<td>R, L</td>
</tr>
<tr>
<td>SEAM</td>
<td>The line of junction between parts; to mark with a crack fissure or cut.</td>
<td>TOPOLOGY GROUP - N.A. FORM GROUP - B4</td>
<td></td>
<td></td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>SEAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ENGLISH DEFI: The place where anything is situated, located or established.
GROUP CLASSI: TOPOLOGY GROUP - E
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SEPARATION
ENGLISH DEFI: The act or process of separating; division.
GROUP CLASSI: TOPOLOGY GROUP - F
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHAFT
ENGLISH DEFI: A cylindrical piece of metal used to carry rotating machine parts.
GROUP CLASSI: TOPOLOGY GROUP - A
SYNONYMS: ARBOR, AXLE, MANDREL, SPINDLE
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHARP
ENGLISH DEFI: Having a keen edge or an acute point.
GROUP CLASSI: TOPOLOGY GROUP - K
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHEET
ENGLISH DEFI: A piece of metal or other substance hammered, rolled or cut very thin.
GROUP CLASSI: TOPOLOGY GROUP - E
SYNONYMS: FACE, PLANE
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHELL
ENGLISH DEFI: A hollow structure or vessel, generally thin and weak.
GROUP CLASSI: TOPOLOGY GROUP - G
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHIELD
ENGLISH DEFI: A cover that conceals or protects.
GROUP CLASSI: TOPOLOGY GROUP - E
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHIM
ENGLISH DEFI: A thin piece of wood or steel placed under a member to bring it to a desired elevation.
GROUP CLASSI: TOPOLOGY GROUP - A, E
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS
END
FEATURE NAME: SHOULDER
ENGLISH DEFI: Anything that supports, bears up or projects like a shoulder.
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>English Definition</th>
<th>Group Class</th>
<th>Topology Group</th>
<th>Relations</th>
<th>Defining Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEVE</td>
<td>A frame of wire mesh.</td>
<td>1</td>
<td>TOPOLOGY GROUP - P</td>
<td>2</td>
<td>L, B, H, T, R, A</td>
</tr>
<tr>
<td>SLAB</td>
<td>A thick flat piece or slice of material.</td>
<td>2</td>
<td>TOPOLOGY GROUP - E</td>
<td>2</td>
<td>L, B, DL, DB</td>
</tr>
<tr>
<td>SLEEVEING</td>
<td>A cylindrical tubing designed to fit over another part.</td>
<td>2</td>
<td>TOPOLOGY GROUP - C</td>
<td>2</td>
<td>R1, R2, L</td>
</tr>
<tr>
<td>SLIT</td>
<td>A long narrow cut or opening; slash.</td>
<td>2</td>
<td>TOPOLOGY GROUP - F</td>
<td>2</td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>SLOT</td>
<td>A long narrow groove or opening; cut.</td>
<td>2</td>
<td>TOPOLOGY GROUP - F</td>
<td>2</td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>SOCKET</td>
<td>A cavity or opening adapted to receive or hold something.</td>
<td>2</td>
<td>TOPOLOGY GROUP - F</td>
<td>2</td>
<td>L, B1, B2, H</td>
</tr>
<tr>
<td>SPHERE</td>
<td>A solid having a surface every point on which is equidistant from the center; ball, globe.</td>
<td>2</td>
<td>TOPOLOGY GROUP - I</td>
<td>2</td>
<td>R</td>
</tr>
<tr>
<td>SPIKE</td>
<td>A large pointed nail.</td>
<td>2</td>
<td>TOPOLOGY GROUP - K</td>
<td>2</td>
<td>R, H, A.R.</td>
</tr>
<tr>
<td>SPINDLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ENGLISH DEFINITION: A rotating rod, axis or shaft, usually slender.

GROUP CLASS: TOPOLOGY GROUP - A
SYNONYMS: ARBOR, AXLE, MANDREL, SHAFT
DEFINING PARAMETERS: R, L

END FEATURE NAME: SPIRAL
ENGLISH DEFINITION: A curve winding like a screw thread or helix.

GROUP CLASS: TOPOLOGY GROUP - R
SYNONYMS: COIL, CURL, HELIX, SPRING
DEFINING PARAMETERS: R1, R2, L, N

END FEATURE NAME: SPINE
ENGLISH DEFINITION: A long keyway. Sometimes also a flat key.

GROUP CLASS: TOPOLOGY GROUP - N.A.
SYNONYMS: N.A.
DEFINING PARAMETERS: R1, R2, L, N

END FEATURE NAME: SPLIT
ENGLISH DEFINITION: To separate into two parts, especially equal.

GROUP CLASS: TOPOLOGY GROUP - F
SYNONYMS: N.A.
DEFINING PARAMETERS: L, B1, B2, H

END FEATURE NAME: SPLIT-RING
ENGLISH DEFINITION: A circular band for holding, connecting or hanging that grips by means of a tightening screw.

GROUP CLASS: TOPOLOGY GROUP - C
SYNONYMS: N.A.
DEFINING PARAMETERS: R1, R2, L

END FEATURE NAME: SPOKE
ENGLISH DEFINITION: One of the small radiating bars inserted in the hub of a wheel or pulley to support the rim.

GROUP CLASS: TOPOLOGY GROUP - E
SYNONYMS: N.A.
DEFINING PARAMETERS: L, B1, B2, H

END FEATURE NAME: SPOOL
ENGLISH DEFINITION: A concave cylinder used for winding.

GROUP CLASS: TOPOLOGY GROUP - Q
SYNONYMS: REEL
DEFINING PARAMETERS: R1, R2, R3, L1, L2, L3

END FEATURE NAME: SPOT-FACE
ENGLISH DEFINITION: A finished round spot on a rough surface, to give a good seat to a screw or bolt-head.

GROUP CLASS: TOPOLOGY GROUP - N.A.
SYNONYMS: N.A.
DEFINING PARAMETERS: R1, R2, L, N

END FEATURE NAME: SPRING
ENGLISH DEFINITION: An elastic body or a contrivance, as a coiled wire.

GROUP CLASS: TOPOLOGY GROUP - R
SYNONYMS: COIL, CURL, HELIX, SPIRAL
DEFINING PARAMETERS: R1, R2, L, N
FEATURE NAME | STICK
---|---
ENGLISH DEFI | A slender piece of wood; like a baton or wand.
GROUP CLASSI | TOPOLOGY GROUP - A FORM GROUP - B1, B5A, C1
SYNONYMS |
RELATIONS |
DEFINING PARAMETERS | R, L
END

FEATURE NAME | STIFFENER
---|---
ENGLISH DEFI | Angle, plate or channel riveted to a member to prevent buckling.
GROUP CLASSI | TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
SYNONYMS |
RELATIONS |
DEFINING PARAMETERS |
END

FEATURE NAME | STOPPER
---|---
ENGLISH DEFI | Something that stops up or closes an opening; one that stops or checks a movement.
GROUP CLASSI | TOPOLOGY GROUP - C FORM GROUP - B1, B5A, C14
SYNONYMS |
RELATIONS |
DEFINING PARAMETERS | R1, R2, L
END

FEATURE NAME | STRING
---|---
ENGLISH DEFI | A slender line or strip, thinner than a cord and thicker than a thread.
GROUP CLASSI | TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS | CORD, ROPE, TWINE
RELATIONS |
DEFINING PARAMETERS | R, L
END

FEATURE NAME | STRIP
---|---
ENGLISH DEFI | A narrow, thin and comparatively long piece of metal.
GROUP CLASSI | TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS | BAND, BELT, RIBBON
RELATIONS |
DEFINING PARAMETERS | L, B1, B2, H
END

FEATURE NAME | STUD
---|---
ENGLISH DEFI | A rivet, boss or nail with a large head; A short rod or bolt threaded at both ends without a head.
GROUP CLASSI | TOPOLOGY GROUP - A FORM GROUP - B2, C3
SYNONYMS | DOWEL, NAIL, NEEDLE, PEG, PIN, TACK
RELATIONS |
DEFINING PARAMETERS | R, L
END

FEATURE NAME | STUFFING
---|---
ENGLISH DEFI | A layer of material used for cushion or for protection.
GROUP CLASSI | TOPOLOGY GROUP - E FORM GROUP - B1, B2, C3
SYNONYMS |
RELATIONS |
DEFINING PARAMETERS | L, B1, B2, H
END

FEATURE NAME | SWELL
---|---
ENGLISH DEFI | A local enlargement.
GROUP CLASSI | TOPOLOGY GROUP - J FORM GROUP - B1, B5A, C1
SYNONYMS | BUMP, BULGE, ENTASIS, SWELLING
RELATIONS |
DEFINING PARAMETERS |
END

FEATURE NAME | SWELLING
---|---
ENGLISH DEFI | A local enlargement.
GROUP CLASSI | TOPOLOGY GROUP - J FORM GROUP - B1, B5A, C1
SYNONYMS |
RELATIONS |
DEFINING PARAMETERS |
END

Appendix A - Feature Dictionary
SYNONYMS: BUMP, BULGE, ENTASIS, SWELL
RELATIONS:
DEFINING PARAMETERS: A, B
END
FEATURE NAME: T-BOLT
ENGLISH DEF: A bolt with a rectangular head used in a T-slot.
GROUP CLASSI: TOPOLOGY GROUP - N.A.  FORM GROUP - N.A.
SYNONYMS:
RELATIONS:
END
FEATURE NAME: T-SLOT
ENGLISH DEF: A slot shaped like an inverted-T, to hold or guide pieces.
GROUP CLASSI: TOPOLOGY GROUP - N.A.  FORM GROUP - N.A.
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS:
END
FEATURE NAME: TABLE
ENGLISH DEF: A platform or a plate used for support.
GROUP CLASSI: TOPOLOGY GROUP - E  FORM GROUP - B2, C3
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: L, B1, B2, H
END
FEATURE NAME: TACK
ENGLISH DEF: A small, sharp-pointed, nail, commonly with tapering sides and a flat head.
GROUP CLASSI: TOPOLOGY GROUP - A  FORM GROUP - B1, B5A, C4
SYNONYMS: DOWEL, NAIL, NEEDLE, PEG, PIN, STUD
RELATIONS:
DEFINING PARAMETERS: R, L
END
FEATURE NAME: TANK
ENGLISH DEF: A large vessel, basin or receptacle for holding a fluid.
GROUP CLASSI: TOPOLOGY GROUP - G  FORM GROUP - B2B, C3
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: L, B, H, T
END
FEATURE NAME: TAPER
ENGLISH DEF: A gradual diminution of size in an elongated object.
GROUP CLASSI: TOPOLOGY GROUP - V  FORM GROUP - B1, B5A C4
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: R1, R2, L, A.R.
END
FEATURE NAME: TAPPED-HOLE
ENGLISH DEF: A hole containing an internal screw thread.
GROUP CLASSI: TOPOLOGY GROUP - T  FORM GROUP - B5B, C11
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: R1, L, T, T2, H, N
END
FEATURE NAME: TEAR
ENGLISH DEF: A drop of liquid; globular.
GROUP CLASSI: TOPOLOGY GROUP - I  FORM GROUP - B1, B5A, C1
SYNONYMS: BUBBLE, DROP, DROPLET, GLOBULE
RELATIONS:
DEFINING PARAMETERS:
END
FEATURE NAME: TETRAHEDRON
ENGLISH DEF: A polyhedron bounded by four plane triangular faces.
GROUP CLASSI: TOPOLOGY GROUP - N.A.  FORM GROUP - B2, C2
SYNONYMS: PRISM, PYRAMID
RELATIONS
DEFINING PARAMETERS : L, B, H, A.R.
END
FEATURE NAME : THICKENING
ENGLISH DEF : A thickened place or a part.
GROUP CLASS : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : THREAD
ENGLISH DEF : The spiral ridge of a screw.
GROUP CLASS : TOPOLOGY GROUP - R FORM GROUP - B5B
SYNONYMS : RELATIONS
DEFINING PARAMETERS : R1, R2, L, N
END
FEATURE NAME : TILE
ENGLISH DEF : A thin plate used for covering or protecting.
GROUP CLASS : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : TORUS
ENGLISH DEF : The surface of a doughnut.
GROUP CLASS : TOPOLOGY GROUP - S FORM GROUP - B1, B5A
SYNONYMS : RELATIONS
DEFINING PARAMETERS : R1, R2
END
FEATURE NAME : TRACK
ENGLISH DEF : A set of rails or a rail.
GROUP CLASS : TOPOLOGY GROUP - E FORM GROUP - B2, C3
SYNONYMS : RAIL RELATIONS
DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : TREPAN
ENGLISH DEF : To cut an outside annular groove around a hole.
GROUP CLASS : TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
SYNONYMS : RELATIONS
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : TUBE
ENGLISH DEF : A long cylindrical body with a hollow center, generally used for conveying something.
GROUP CLASS : TOPOLOGY GROUP - C FORM GROUP - B1, B5A
SYNONYMS : DUCT, FLUE, GLAND, PIPE RELATIONS
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : TUNNEL
ENGLISH DEF : An underground passage-way.
GROUP CLASS : TOPOLOGY GROUP - D FORM GROUP - B1, B5A, C2
SYNONYMS : RELATIONS
DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : TWINE
ENGLISH DEF : A string consisting of two strands twisted together.
GROUP CLASS : TOPOLOGY GROUP - A FORM GROUP - B1, B5A
SYNONYMS : CORD, ROPE, STRING
RELATIONS
DEFINING PARAMETERS : R, L
END
FEATURE NAME : TWIST
ENGLISH DEFI : Thread or cord made of slightly twisted strands.
GROUP CLASSI : TOPOLOGY GROUP - R FORM GROUP - B5B
SYNONYMS : DEFINING PARAMETERS : R1, R2, L, N
END
FEATURE NAME : UNDERCUT
ENGLISH DEFI : A cut that leaves an overhanging edge. A cut with inwardly sloping sides.
GROUP CLASSI : TOPOLOGY GROUP - N.A. FORM GROUP - N.A.
SYNONYMS : RELATIONS : DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : VALLEY
ENGLISH DEFI : Any depression or hollow like a valley.
GROUP CLASSI : TOPOLOGY GROUP - F FORM GROUP - C8, C10
SYNONYMS : RELATIONS : DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : VANE
ENGLISH DEFI : An arm or blade extending from the rotating shaft; a movable thin plate of metal.
GROUP CLASSI : TOPOLOGY GROUP - E FORM GROUP - B2
SYNONYMS : RELATIONS : DEFINING PARAMETERS : L, B1, B2, H
END
FEATURE NAME : VEIN
ENGLISH DEFI : A thin, tiny tubular vessel.
GROUP CLASSI : TOPOLOGY GROUP - C, F FORM GROUP - B1, B2, B5A
SYNONYMS : RELATIONS : DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME : VERTEX
ENGLISH DEFI : The highest point of anything; apex.
GROUP CLASSI : TOPOLOGY GROUP - K FORM GROUP - B1, C4
SYNONYMS : APEX, PEAK
RELATIONS : DEFINING PARAMETERS : R, H, A.R.
END
FEATURE NAME : WADDING
ENGLISH DEFI : A small compact mass of any shaft or flexible substance.
GROUP CLASSI : TOPOLOGY GROUP - A, C, E FORM GROUP - B1, B2, B5A, C3
SYNONYMS : PACKING, PAD, PADDING
RELATIONS : DEFINING PARAMETERS : R, L
END
FEATURE NAME : WASHER
ENGLISH DEFI : A flattened, ring-shaped device used to improve tightness of a screw fastener.
GROUP CLASSI : TOPOLOGY GROUP - C FORM GROUP - B1, B2, B5A, C3
SYNONYMS : RELATIONS : DEFINING PARAMETERS : R1, R2, L
END
FEATURE NAME: WEB
ENGLISH DEF: Any structure woven of or as interlaced strands.
GROUP CLASS: TOPOLOGY GROUP - P
FORM GROUP - C3
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: L, B, DL, DB
END

FEATURE NAME: WEDGE
ENGLISH DEF: A V-shaped piece of solid metal.
GROUP CLASS: TOPOLOGY GROUP - N.A.
FORM GROUP - N.A.
SYNONYMS: KEY, COTTER
RELATIONS:
DEFINING PARAMETERS: L, B, H
END

FEATURE NAME: WELD
ENGLISH DEF: A rounded filling used to unite two pieces of metal.
GROUP CLASS: TOPOLOGY GROUP - N.A.
FORM GROUP - N.A.
SYNONYMS:
RELATIONS:
END

FEATURE NAME: WELL
ENGLISH DEF: A vertical hole or shaft.
GROUP CLASS: TOPOLOGY GROUP - B
FORM GROUP - B1, B5A
SYNONYMS: HOLE
RELATIONS:
DEFINING PARAMETERS: R, L
END

FEATURE NAME: WHEEL
ENGLISH DEF: A circular frame with a hub at the center and capable of rotating on a central axes for attachment to an axle.
GROUP CLASS: TOPOLOGY GROUP - A
FORM GROUP - B1, B5A
SYNONYMS:
RELATIONS:
DEFINING PARAMETERS: R, L
END

FEATURE NAME: WIRE
ENGLISH DEF: A strand of ductile metal.
GROUP CLASS: TOPOLOGY GROUP - A
FORM GROUP - B1, B5A
SYNONYMS: CABLE
RELATIONS:
DEFINING PARAMETERS: R, L
END
Appendix B - FeatureMod Program Listing

Index Of Subroutines.

1. FADTO - Allows additions to existing objects and assemblies
2. FASSM - Allows the addition of objects to different assemblies
3. FBCKV - Displays the back view of an object
4. FBMAT - Computes the P-matrix for a periodic cubic B-spline surface and displays the point mesh
5. FBMSH - Computes the mesh of points lying on a periodic cubic B-spline surface
6. FBOTV - Displays the bottom view of an object
7. FBPCH - Creates a B-spline surface patch for specified control points
8. FCAPS - Converts a mixed case string input to all upper case
9. FCOMM - Sets up counters for shading
10. FCSN1 - Computes w-tangents at endpoints of patch for n eq 0.0
11. FCSN2 - Computes w-tangents at endpoints of patch for 0.0 lt n lt 2.0
12. FCSN3 - Computes w-tangents at endpoints of patch for n = 2.0
13. FCSN4 - Computes w-tangents at endpoints of patch for n gt 2.0
14. FCTCP - Maps points on a B-spline curve into its control points
15. FDASM - Defines New Assemblies
16. FDEL - Deletes a specified object or assembly
17. FDINS - Defines new instances and makes them part of object
18. FDISP - Displays a specified object or assembly
19. FOBJ - Defines New Objects
20. FDRT - Draws all constant u lines
21. FDRWW - Draws all constant w lines
22. FEDAT - Sets up 2 polygon edge attributes (color, edge type)
23. FFNTV - Displays the front view of an object
24. FGBMT - Computes the elements of B-matrix based on global axes
25. FGST - Requests a string input from the user
26. FHMAT - Assembles the global B-matrix for bicubic Hermite patch
27. FHMSH - Uses the B-matrix and computes different points on a bicubic Hermite patch
28. FINIT - Initializes all input devices
29. FINPT - Waits for any input from the user (event mode)
30. FINQU - Allows inquiries about features and returns requested information
31. FLBMT - Computes the elements of B-matrix based on local axes
32. FLFTV - Displays the left side view of an object
33. FMENU - Displays different menu pages in the defined area
34. FMPK1 - Processes the pick-id of the item picked from Menu 1 (FILES)
35. FMPK10 - Processes the pick-id of the item picked from Menu 10 (DEFINE)
36. FMPK2 - Processes the pick-id of the item picked from Menu 2 (COMMANDS)
37. FMPK3 - Processes the pick-id of the item picked from Menu 3 (MODELING)
38. FMPK4 - Processes the pick-id of the item picked from Menu 4 (RENDERING)
39. FMPK5 - Processes the pick-id of the item picked from Menu 5 (VIEWS)
40. FMPK6 - Processes the pick-id of the item picked from Menu 6 (WINDOWS)
41. FMPK7 - Processes the pick-id of the item picked from Menu 7 (SHADING)
42. FMPK8 - Processes the pick-id of the item picked from Menu 8 (SINGLE WINDOWS)
43. FMPK9 - Processes the pick-id of the item picked from Menu 9 (COLOR)
44. FMSSG - Displays messages and scrolls them in the defined area
45. FMT11 - Multiplies a 1x4 and a 4x1 matrix
46. FMT14 - Multiplies a 1x4 and a 4x4 matrix
47. FMT44 - Multiplies a 4x4 and a 4x4 matrix
48. FMULT - Displays three primary views of an object simultaneously
49. FNMRT - Normalizes a vector to unit length
50. FOPEN - Used to initialize graPHIGS
51. FPARS - Parses a string input into a word array
52. FPGAT - Sets up 2 polygon attributes (color, interior style)
53. FPNTS - Computes global B-matrix for object with axis = z-axis
54. FPRMP - Displays prompts to users
55. FPRPK - Processes the menu number to which the item picked belonged.
56. FQURY - Queries number of assemblies and objects and their names
57. FRBMH - Computes the mesh of points lying on a periodic cubic rational B-spline surface
58. FRBMT - Computes the H-matrix and P-matrix for a periodic cubic rational B-spline surface and displays the point mesh
59. FRCLL - Recalls a existing model file
60. FRGTV - Displays the right side view of an object
61. FROTA - Allows viewing transformations on the model
62. FSCAN - Matches a string input with predefined syntax
63. FSCLE - Allows zooming
64. FSCRA - Displays all background areas
65. FSTCP - Maps points on a B-spline surface into its control points
66. FSTOR - Stores all points on patch into polygon forms
67. FSTRG - Seeks an string input from user (request mode)
68. FSUCO - Computes object to be displayed with title screen
69. FTITL - Displays Title screen for “FEATURE”
70. FTOPV - Displays the top view of an object
71. FTRNS - Transforms all points from the global coordinate axes to local axes
72. FUWVT - Computes a 1x4 vector using the vector parameter
73. FWIND - Sets all Windows and Viewports
**PROGRAM DESCRIPTION**

This is the main program for the Feature Based Modeling software. It is used to display the title page and then the menu screen. It passes the command to the required subroutine after an input has been specified.

*By: Ashit R. Gandhi*

*Date: 01/13/89*

*Parameters Used:*

*None*

REAL*4 V(4), SCOL(3)
INTEGER ICLASS,IDEV,IPKID, ASNUM, OBNUM, INNUM
COMMON STRG*50, PMENU(50)*15
COMMON/SHADE/SCOL
COMMON/IDS/ASMM, OBNUM, INNUM

SET SCALE FACTORS

SCOL(1) = 1.0
SCOL(2) = 0.0
SCOL(3) = 0.0

SET COMPONENT ID NUMBERS

ASNUM = 100
OBNUM = 1000
INNUM = 10000

OPEN GRAPHICS
CALL FOPEN

DEFINE WINDOWS AND VIEWPORTS
CALL FHIND

DISPLAY THE TITLE SCREEN
CALL FTITL

DISPLAY THE BASIC MENUS
CALL FSCRA

INITIALIZE ALL INPUTS
CALL FINIT

DISPLAY THE FIRST SCROLL MESSAGES
DO 50 I = 1,5
   CALL FMSSG(I,STRG)
50 CONTINUE

DISPLAY FIRST MENU AND MAKE SURE THAT FIRST PICKS ARE START OR RECALL

INNUM = 1
CALL FMENU2(INNUM, PMENU)
CALL GPUPSINS(1,2)
   CALL FPMP(1)
300   CALL FNPNT(PMENUM, ICLASS, IDEV, IPKID, V, STRG)
      IF (ICLASS .NE. 6) THEN
         STRG = 'INVALID OPTION SELECTED'
         CALL FMSSG(30, STRG)
      ENDIF
      GOTO 300
      ELSEIF (IPKID .NE. 5 .AND. IPKID .NE. 6) THEN
         STRG = 'INVALID OPTION SELECTED'
         CALL FMSSG(30, STRG)
      ENDIF
      GOTO 300
      ELSE
         CALL FPMP(INNUM, IPKID, PMENU)
         CALL GPUPSINS(1, 2)
      ENDIF
      ENDIF

Appendix B - FeatureMod Program Listing
*ONCE FILE HAS BEEN STARTED OR RECALLED ALLOW USER TO SELECT DIFFERENT OPTIONS*

CALL GPUPHS(1,2)

200 CALL FPRMP(2)

*GET INPUT FROM THE USER*

CALL FINPT(PMENU,ICLASS,IDEV,IPKID,V,STRG)

*IF INPUT IS VALUATOR*

IF (ICLASS .EQ. 3) THEN

*IF DEVICE NUMBER IS 4 THEN SCALE OTHERWISE ROTATE THE VIEWS*

IF (IDEV .NE. 4) THEN
   CALL FROTA(V(1),V(2),V(3),IDEV)
   ELSE CALL FSCLE(V(1))
   CALL GPUPHS(1,2)
   ENDIF

ENDIF

*IF INPUT IS PICK*

IF (ICLASS .EQ. 5) THEN
   CALL FPRPK(MNUM,IPKID,PMENU)
   CALL GPUPHS(1,2)
   ENDIF

IF (MNUM .EQ. 0) GOTO 100
GOTO 200

100 STOP
END

**SUBROUTINE BUSH(R1,R2,THK,LOC,ORI,ICOL,BOOL)**

**PROGRAM DESCRIPTION**

**THIS ROUTINE WILL COMPUTE THE CONTROL POINTS REQUIRED FOR CREATING A BUSHING USING B-SPLINE SURFACES.**

**BY: ASHIT R. GANDHI**

**DATE: 01/13/89**

**PARAMETERS USED:**

**R1 = INNER RADIUS OF BUSHING**

**R2 = INNER RADIUS OF BUSHING**

**THK = THICKNESS OF THE BUSHING**

**LOC = LOCATION OF THE BUSHING**

**ORI = ORIENTATION OF THE BUSHING**

**ICOL = COLOR FOR THE BUSHING**

**BOOL = BOOLEAN TYPE**

**SUBROUTINE BUSH(R1,R2,THK,LOC,ORI,ICOL,BOOL)**

INTEGER*4 ICOL

INTEGER*4 ASNUM, OBNUM, INNUM, LTYPE, CNUM

REAL*4 LOC(3), ORI(3), PT(3)

REAL*4 R1, R2, THK

REAL*4 PLOC(3), PORI(3)

CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8

COMMON/IDS/ASNUM,OBNUM,INNUM

COMMON/CMP/ASSEM,OBJECT

COMMON/REND/PLOC,PORI

DO 50 I = 1,3
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)

50 CONTINUE

IF (BOOL .EQ. ' ') LTYPE = 1
IF (BOOL .EQ. '*') LTYPE = 3
IF (BOOL .EQ. '-') LTYPE = 2

SUBROUTINE BUSH(R1,R2,THK,LOC,ORI,ICOL,BOOL)

Appendix B - FeatureMod Program Listing
SUBROUTINE CONE

*** PROGRAM DESCRIPTION ***

** THIS ROUTINE WILL COMPUTE THE CONTROL POINTS REQUIRED TO **
CREATE A CONE USING B- SPLINE SURFACES **

** BY: ASHIT R. GANDHI **

** DATE: 11/12/88 **

** PARAMETERS USED: **

** RAD = RADIUS OF THE CONE {REAL, I/P} **
** LEN = LENGTH OF THE CONE {REAL, I/P} **
** LOC = LOCATION OF THE CONE {REAL, I/P} **
** ORI = EULER ORIENTATION FOR THE CONE {REAL, I/P} **
** ICOL = COLOR FOR THE CONE **
** BOOL = BOOLEAN OPERATOR ON THE CONE **
** ICHK = FLAG FOR NEW FEATURE **
** PLACE = LOCAL POSITION OF CONE **

SUBROUTINE CONE(IFL, RAD, LEN, ORI, ICOL, BOOL, ICHK, PLACE)
OBJECT(CNUM-1000),LOC,ORI)

ENDIF

IF (IFL .NE. 1 .AND. IFL .NE. 2) THEN
   WRITE(6,*)'CONE =:::> FLAG FOR CAPS NOT 1 OR 2'
   WRITE(6,*)'CONE =::=> FLAG SHOULD BE SET TO 1 OR 2'
   RETURN
ENDIF

PI = 3.14159

DEL = PI/6.0

R(1) = 1.0
R(2) = 1.0 - (1.0 - AR)/3.0
R(3) = 1.0 - 2.0*(1.0 - AR)/3.0
R(4) = AR

*COMPUTE THE HOLLOW CONE

DO 300 K = 1,4
   WRITE(10,10)INNUM,'4 4 0 0',ICOL, NJ, NM
   THETA = 0.0
   DO 200 I = 1,4
      DO 100 J = 1,4
         PT(1) = U(J)*LEN + PLACE
         PT(2) = FY(K)*RAD*SIN(THETA)*R(J)
         PT(3) = FZ(K)*RAD*COS(THETA)*R(J)
         PTS(I,J,1) = PT(1)
         PTS(I,J,2) = PT(2)
         PTS(I,J,3) = PT(3)
      100 Continue
   THETA = THETA + DEL
   200 Continue
   CALL FSTCP(U,N,PTS,PNTS)
   CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
   DO 250 I = 1,4
      WRITE(10,*) (PNTS(I,J,1),J=1,4)
      WRITE(10,*) (PNTS(I,J,2),J=1,4)
      WRITE(10,*) (PNTS(I,J,3),J=1,4)
      250 Continue
   IF (IFL .EQ. 1) RETURN

*IF SOLID, COMPUTE THE END CAPS OF THE CONE IF NEEDED

ND = 3
   IF (AR .EQ. 0.0) ND = 5
   DO 700 IJ = 1,4 ND
      600 X = IJ
      WRITE(10,10)INNUM,'4 4 0 0',ICOL, NJ, NM
      THETA = 0.0
      DO 400 J = 1,4
         PT(1) = FX(IJ)*LEN + PLACE
         PT(2) = FY(K)*U(J)*RAD*SIN(THETA)*R(IJ)
         PT(3) = FZ(K)*U(J)*RAD*COS(THETA)*R(IJ)
         PTS(I,J,1) = PT(1)
         PTS(I,J,2) = PT(2)
         PTS(I,J,3) = PT(3)
      400 Continue
   THETA = THETA + DEL
   500 Continue
   CALL FSTCP(U,N,PTS,PNTS)
   CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
   DO 550 I = 1,4
      WRITE(10,*) (PNTS(I,J,1),J=1,4)
      WRITE(10,*) (PNTS(I,J,2),J=1,4)
      WRITE(10,*) (PNTS(I,J,3),J=1,4)
      550 Continue
   600 Continue
   RETURN

10 FORMAT(1X,I5,1X,A7,3(I2))

END

******************************************************************************
******************************************************************************
** SUBROUTINE COUNT(R1,R2,L1,L2,IFL,IFL1,LOC,ORI,COLOR,BOOL) **
******************************************************************************

Appendix B - FeatureMod Program Listing 172
** PROGRAM DESCRIPTION

This routine will compute the control points required for creating a counter bore or a counter sink using B-spline surfaces.

By: Ashit R. Gandhi
Date: 11/12/88

Parameters used:

- **R1** = radius at top of feature
- **R2** = intermediate radius of feature
- **L1** = length between top and intermediate radius
- **L2** = length of feature
- **IFL** = flag indicating caps for counter bore
- **IFL1** = flag indicating caps for counter sink
- **LOCA** = location of the feature
- **ORI** = orientation of the feature
- **COLOR** = color to be given to the feature
- **BOOL** = boolean operator for the feature

SUBROUTINE COUNT (R1, R2, L1, L2, IFL, IFL1, LOCA, ORI, ICOL, BOOL)

INTEGER*4 ICOL
INTEGER*4 ASNUM, OBNUM, INNUM, LTYPE, CNUM
REAL*4 LOC(3), ORI(3), PT(3)
REAL*4 R1, R2, L1, L2
REAL*4 PLOC(3), PORI(3)
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8
COMMON/IDS/ASNLM, OBNUM, INNUM
COMMON/COMP/ASSEM, OBJECT
COMMON/PRTOF/CNUM
COMMON/RENO/PLOC, PORI
DO 50 I = 1, 3
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)
50 CONTINUE

IF (BOOL EQ '++') LTYPE = 1
IF (BOOL EQ '+') LTYPE = 2
IF (BOOL EQ 'x') LTYPE = 3
INNUM = INNUM + 1
IF (IFL EQ 1) THEN
   NSURF = 0
ELSE
   NSURF = 8
ENDIF
IF (IFL1 EQ 1) THEN
   NSURF = NSURF + 8
   CALL FDINS (INNUM, LTYPE, 'C. BORE ', NSURF, CNUM, 1
   OBJECT (CNUM-1000), LOC, ORI)
ELSE
   NSURF = NSURF + 12
   CALL FDINS (INNUM, LTYPE, 'C. SINK', NSURF, CNUM, 1
   OBJECT (CNUM-1000), LOC, ORI)
ENDIF
ICHK = 1
IF (R2 .GT. R1) THEN
   R3 = R2
   R2 = R1
   R1 = R3
ENDIF
AR = R2/R1
IF (IFL1 EQ 2) THEN
   CALL CONE (1, R1, L1, AR, LOC, ORI, ICOL, BOOL, ICHK, 0.)
   CALL CVND (1, R2, L2, LOC, ORI, ICOL, BOOL, ICHK, L1)
ELSE
   CALL CVND (1, R1, L1, LOC, ORI, ICOL, BOOL, ICHK, 0.)
   CALL RING (1, R2, R1, LOC, ORI, ICOL, BOOL, ICHK, L1)
ENDIF
IF (IFL EQ 2) THEN
   CALL RING (0, R1, LOC, ORI, ICOL, BOOL, ICHK, 0.)
   R1 = L1 + L2
   CALL RING (0, R2, LOC, ORI, ICOL, BOOL, ICHK, RL)
ENDIF
RETURN
END
SUBROUTINE CYNDRI

*** BEGIN SUBROUTINE CYNDRI ***

PARAMETERS USED:
** RAD = RADIUS OF THE CYLINDER (REAL, I/P)
** LEN = LENGTH OF THE CYLINDER (REAL, I/P)
** LOC = LOCATION OF THE CYLINDER (REAL, I/P)
** ORI = EULER ORIENTATION FOR THE CYLINDER (REAL, I/P)
** ICOL = COLOR FOR THE CYLINDER
** BOOL = BOOLEAN OPERATOR ON THE CYLINDER
** ICHK = FLAG FOR NEW FEATURE
** PLACE = LOCATION IN LOCAL AXIS

*** END SUBROUTINE CYNDRI ***

SUBROUTINE CYNDRI(IFL, RAD, LEN, LOC, ORI, ICOL, BOOL, ICHK, PLACE)

INTEGER*4 ASMF, OBSM, INNM, LTYPE, CNUM
REAL*4 RAD, LEN, LOC(3), ORI(3), FX(4), FY(4), FZ(4)
REAL*4 U(4), W(4), H(20,20), PT(3), PTS(4,4,3), PNTS(20,20,3)
REAL*4 PLOC(3), PORI(3)
CHARACTER BOOL*1, ASSEM(90*D)*8, OBJECT(9000)*8

COMMON/IDS/ASNUM, OBSM, INNM
COMMON/COMP/ASSEM, OBJECT
COMMON/PRTOF/CNUM
COMMON/REND/PLOC, PORI

DATA FX/-1.0, 0.0, 1.0, 2.0/
DATA FY/ 1.0, 1.0, -1.0, -1.0/
DATA FZ/ 1.0, -1.0, 1.0, -1.0/
DATA U/0.0, 0.33333333, 0.66666667, 1.0/
DATA W/0.0, 0.33333333, 0.66666667, 1.0/
DATA H/400*1.0/

DO 50 I = 1,3
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)
50 CONTINUE

IF (BOOL .EQ. '+') LTYPE = 1
IF (BOOL .EQ. '-') LTYPE = 2
IF (BOOL .EQ. '¤') LTYPE = 3

IF (ICHK .EQ. 0) THEN
   INNM = INNM + 1
   NSURF = 4
   ELSE
      NSURF = 12
      ENDIF
      CALL FDSIN(INNM, LTYPE, 'CYLINDER', NSURF, CNUM,
          OBJECT(CNUM-1000), LOC, ORI)
ENDIF

IF (IFL .EQ. 1 .AND. IFL .NE. 2) THEN
   WRITE(*,*)'CYLINDER ====> FLAG FOR CAPS NOT 1 OR 2'
   RETURN
ENDIF

pi = 3.14159
DELI = PI/6.0

*COMPUTE THE HOLLOW CYLINDER

DO 300 K = 1,4
   WRITE(10,10)INNM, '4 0 0 ', ICOL, NU, NH
   Theta = 0.0
300   DO 200 I = 1,4
      DO 100 J = 1,4
         PT(I) = U(J)*LEN + PLACE
         PT(I) = W(J)*RAD * SIN(Theta)
         PT(I) = FZ(J)*RAD * COS(Theta)
      ENDIF
      PT(I) = PT(I)
   ENDIF

Appendix B - FeatureMod Program Listing
PTS(I,J,2) = PT(2)
PTS(I,J,3) = PT(3)

400 CONTINUE
THETA = THETA + DEL

500 CONTINUE
CALL FSTCP(U,N,PTS,PNT$)
CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)

DO 550 I = 1,4
WRITE(10,*|(PNTS(IJ,1,J=1,4)
WRITE(10,*|(PNTS(IJ,2),J=1,4)
WRITE(10,*|(PNTS(IJ,3),J=1,4)
550 HRITE(10,*)(H0(1,J1,J=1,4)

600 CONTINUE
700 CONTINUE
RETURN

10 FORMAT(1X,I5,1X,A7,3(1X,I2)
END

SUBROUTINE ELPSOD

*********************************************************************************************
** SUBROUTINE ELPSOD(RX,RY,LOC,ORI,ICOL,BOOL) **
** PROGRAM DESCRIPTION **
** THIS ROUTINE COMPUTES THE CONTROL POINTS THAT WOULD BE **
** REQUIRED TO CREATE A ELLIPSOID USING B-SPLINE SURFACES **
** BY: ASHIT R. GANDHI **
** DATE: 11/21/88 **
** PARAMETERS USED: **
** RAD = RADIUS OF THE ELLIPSOID (REAL I/P) **
** LOC = LOCATION OF THE ELLIPSOID (REAL, I/P) **
** *********************************************************************************************

INTEGER*(• ASNUM, OBNUM, INNUM, LTYPE, CNUM
REAL*4 LOC(3), PT(3), PTS(4,4,3), ORI(3)
REAL*4 RX, RY, RX(4), PXS(20,20,3), HY(20,20), XS(2), YS(4), ZS(4)
REAL*4 PLOCIS), PORI(3)
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8
COMMON/PATCH/NUM, NM
COMMON/IDS/ASNUM, OBNUM, INNUM
COMMON/COMP/ASSEM, OBJECT
**SUBROUTINE FADTO**

******************************************************************************
** SUBROUTINE FADTO(IER) **
******************************************************************************
** PROGRAM DESCRIPTION **
** THIS SUBROUTINE WILL BE USED TO ADD TO A EXISTING OBJECT OR ASSEMBLY. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 12/16/88 **
** PARAMETERS USED: **
** IER : ERROR CODE **
******************************************************************************

INTEGER*4 ASNUM, OBNUM, INNUM, CNUM

CHARACTER*8 ONAME, OBJECT(9000)*8, STRG*50, WORD(15)*12

Appendix B - FeatureMod Program Listing 176
SUBROUTINE FASSM

******************************************************************************
******************************************************************************
** SUBROUTINE FASSM(NOBJ, LCWA, ORI) **
** 
** PROGRAM DESCRIPTION **
** 
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING AN **
** ASSEMBLY. CORRECT LOCATION AND ORIENTATION OF THE OBJECT IS ALSO **
** REQUESTED **
** 
** BY: ASHIT R. GANDHI **
** DATE: 11/12/88 **
** 
** PARAMETERS USED: **
** NOBJ = ID OF OBJECT TO BE ASSEMBLED (INTEGER, O/P) **
** LOCA = LOCATION OF THE OBJECT (REAL, O/P) **
** ORI = ORIENTATION OF THE OBJECT (REAL, O/P) **
** 
******************************************************************************
******************************************************************************
*DEFINE GENERIC INFORMATION

NOBJ = 0
ONAME = ',
MSG = ',
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1,ERRIND,UNITS,CSIZE,ASIZE)

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS

PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPEST(7)
CALL GPPOST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*CONVERT ALL NUMBERS TO CHARACTERS FOR DISPLAY

WRITE(LOCA1,'(F7.2)') LOCA(1)
WRITE(LOCA2,'(F7.2)') LOCA(2)
WRITE(LOCA3,'(F7.2)') LOCA(3)
WRITE(ORI1,'(F7.2)') ORI(1)
WRITE(ORI2,'(F7.2)') ORI(2)
WRITE(ORI3,'(F7.2)') ORI(3)

*DRAW ALL TITLE TEXT

POS(1) = 0.62
POS(2) = 0.80
STRG = 'OBJECT = ' //ONAME
CALL GPPKID(1)
CALL GPPAN(POS,17,STRG)

POS(1) = 0.62
POS(2) = 0.67
STRG = 'LOCATION = ' //LOCA1//LOCA2//LOCA3
CALL GPPKID(2)
CALL GPPAN(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.84
STRG = 'ORIENT = ' //ORI1//ORI2//ORI3
CALL GPPKID(3)
CALL GPPAN(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.91
STRG = 'ASSEMBLY COMPLETE'
CALL GPPKID(4)
CALL GPPAN(POS,17,STRG)

POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPPAN(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.15
STRG = 'ABORT'
CALL GPPKID(16)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.02
POS(2) = 0.80
CALL GPAN2(POS,32,MSG)
CALL GPCCLST
*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)
*ASSOCIATE MENU TO VIEW 5
CALL GPDRV(1,5,7,0)
*UPDATE WORKSTATION AND PROMPT USER FOR INPUT
CALL FPRMP(14)
ICLASS = 2
CALL GPPK1(1,1,1,ICLASS,1,1)
*WAIT FOR A PICK
100 CALL GPAW1(1,1,1,ICLASS,IDEV)
   IF (ICLASS .NE. 5) GOTO 100
   CALL GPGETPK(1,1,PPATH)
   IPKID = PPATH(2)
   GOTO (1,2,3,4,5,6,IFKID
   IF ITEM PICKED IS "OBJECT" THEN GET NAME OF OBJECT AND COMPUTE THE
   OBJECT NUMBER
1   CALL FGTST(CHAR)
   NOBJ = 0
   CALL FPARS(CHAR,4,WORD)
   ONAME = WORD(1)(1:8)
   DO 200 I = 1,OBJNUM-1000
      IF (ONAME .EQ. OBJECT(I)) THEN
         NOBJ = I+1000
         MSG = I+1000
         REMIND(22)
   ENDIF
   GOTO 50
200 CONTINUE
   IF ASSEMBLY INFORMATION EXISTS THEN CHECK IF OBJECT IS ALREADY A PART
   OF A ASSEMBLY
      IF (IFL22 .EQ. 22) THEN
         DO 150 J = 1,20000
            READ(L22,10,END=175)ICHK,CHAR1,JUNK,CHAR2
            IF (ICHK .EQ. NOBJ) THEN
               ONAME = ' '
               NOBJ = 0
               MSG = 'OBJECT PART OF ASSEMBLY ' //ASSEM(JUNK-100)
            ENDIF
            READ(L22,10,END=175)R1,R2,R3,R4,R5,R6
         CONTINUE
         150 BACKSPACE(22)
      ENDIF
      175 ENDIF
      GOTO 50
   ENDIF
200 CONTINUE
   IF NON EXISTING OBJECT, GET A NEW NAME
      IF (NOBJ .EQ. 0) THEN
         ONAME = ' 'MSG = 'NON-EXISTENT OBJECT'
      ENDIF
      GOTO 50
   ENDIF
   GET LOCATION INFORMATION
2   CALL FGTST(CLOCA)
      IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9')
         .AND. (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-' )) THEN
         READ(CLOCA,*LOC(1),LOC(2),LOC(3))
      GOTO 50
      ELSE
         GOTO 100
      ENDIF
   GET ORIENTATION INFORMATION
3   CALL FGTST(CORI)
      IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9')
         .AND. (CORI(1:1) .EQ. '+' .OR. CORI(1:1) .EQ. '-' )) THEN
SUBROUTINE FBCKV

******************************************************************************
** SUBROUTINE FBCKV(NVIEW) **
** **
** PROGRAM DESCRIPTION **
** **
** THIS SUBROUTINE WILL DISPLAY THE BACK VIEW OF ANY OBJECT. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 11/07/88 **
** **
** PARAMETERS USED: **
** **
** NVIEW = VIEW IN WHICH THE TOP VIEW IS TO BE DISPLAYED **
** (INTEGER,I/P) **
** **
******************************************************************************

SUBROUTINE FBCKV(NVIEW)
REAL*4 MAT(4,4), MAT1(4,4), MAT2(4,4), SCALEF(3)
COMMON /SCLE/SCALEF
PI = 3.14159
DL = PI

*ROTATE THE VIEW ELEMENT ABOUT THE Y-AXIS BY +180 DEGREES
CALL GPROTY(DL,MAT2)
**SCALE THE MODEL**
CALL GPSC3(SCALEF, MAT1)

**COMPOSE THE TRANSFORMATION**
CALL GPCMT3(MAT2, MAT1, MAT)

**ASSOCIATE THIS ROTATION TO THE VIEW**
CALL GPVMT3(1,NVIEW, MAT)
CALL GPPV(1,NVIEW, 2, 1)
RETURN
END

---

**SUBROUTINE FBMAT**

```
SUBROUTINE FBMAT(IU, IN, MPTS, NPTS, PNTS)

PROGRAM DESCRIPTION

THIS SUBROUTINE WILL ASSEMBLE THE P-MATRIX TO BE USED IN THE
CREATION OF A B-SPLINE PERIODIC CUBIC SURFACE.
AFTER ASSEMBLING THE MATRIX IT WILL COMPUTE THE POINTS ON THE
SURFACE.

BY: ASHIT R. GANDHI

DATE: 11/03/88

PARAMETERS USED:

IU = FLAG INDICATING WHETHER SURFACE IS CLOSED AT U = 0
(INTEGER, I/P)
0 = OPEN
1 = CLOSED

IM = FLAG INDICATING WHETHER SURFACE IS CLOSED AT M = 0
(INTEGER, I/P)
0 = OPEN
1 = CLOSED

MPTS = NUMBER OF POINTS IN THE M-DIRECTION (INTEGER, I/P)

NPTS = NUMBER OF POINTS IN THE N-DIRECTION (INTEGER, I/P)

PNTS = ARRAY OF POINTS DEFINING THE SURFACE

REAL*4 PMAT(4,4,3), PTS(0:19,0:19,3), PT(20,20,3), PNTS(20,20,3)

READ IN VALUES FOR THE CONTROL POINTS

DO 300 I = 0, MPTS - 1
   DO 200 J = 0, NPTS - 1
      DO 100 K = 1, 3
         PTS(I,J,K) = PNTS(I+1,J+1,K)
      100 CONTINUE
   200 CONTINUE
300 CONTINUE

CHECK TO SEE WHETHER THE SURFACE IS CLOSED OR OPEN

IF SURFACE IS OPEN AT U = 0.0 AND OPEN AT W = 0.0
   IF (IU .EQ. 0 .AND. IM .EQ. 0) THEN
      SEND = NPTS - 3
   ENDIF

IF SURFACE IS OPEN AT U = 0.0 AND CLOSED AT W = 0.0
   IF (IU .EQ. 0 .AND. IM .EQ. 1) THEN
      SEND = MPTS - 3
   ENDIF

IF SURFACE IS CLOSED AT U = 0.0 AND OPEN AT W = 0.0
   IF (IU .EQ. 1 .AND. IM .EQ. 0) THEN
      SEND = MPTS - 3
   ENDIF

IF SURFACE IS CLOSED AT U = 0.0 AND CLOSED AT W = 0.0
   IF (IU .EQ. 1 .AND. IM .EQ. 1) THEN
```

---

Appendix B - FeatureMod Program Listing
SUBROUTINE FBMSH

******************************************************************************
** SUBROUTINE FBMSH(PMAT,NU,NM,PT) **
** **
** PROGRAM DESCRIPTION **
** **
** THIS SUBROUTINE USES THE GEOMETRIC COEFFICIENT MATRIX TO **
** CALCULATE THE SPECIFIED NUMBER OF POINTS WHICH WOULD LIE ON **
** THE PERIODIC CUBIC B-SPINE SURFACE **
** **
** BY: ASHIT R. GANDHI **
** **
** DATE: 10/31/88 **
** **
** PARAMETERS USED: **
** **
** PMAT = GEOMETRIC COEFFICIENT MATRIX (REAL,I/P) **
** **
** NU = NUMBER OF POINTS TO BE COMPUTED IN U-DIRECTION **
** (INTEGER,I/P) **
** **
** NM = NUMBER OF POINTS TO BE COMPUTED IN W-DIRECTION **
** (INTEGER,I/P) **
** **
** PT = COORDINATES OF THE POINTS ON THE SURFACE **
** (REAL,O/P) **
** **
******************************************************************************

INTEGER NU, NM
REAL M(4,4), H(4,4)
REAL P(20,20,3)

******************************************************************************
** SUBROUTINE FBMSH(PMAT,NU,NM,PT) **
** **
INTEGER NU, NM
REAL M(4,4), NT(4,4), PMAT(4,4), PZMAT(4,4)
REAL P(20,20,3)

******************************************************************************
** SUBROUTINE FBMSH(PMAT,NU,NM,PT) **
** **
INTEGER NU, NM
REAL M(4,4), H(4,4)
REAL P(20,20,3)

******************************************************************************
*SPLIT THE P-MATRIX INTO ITS RESPECTIVE ELEMENTS*

```fortran
DO 200 I = 1,4
  DO 100 J = 1,4
    PXMAT(I,J) = PMAT(I,J,1)
    PYMAT(I,J) = PMAT(I,J,2)
    PZMAT(I,J) = PMAT(I,J,3)
  100 CONTINUE
200 CONTINUE
```

*CALCULATE THE DELTA CHANGES IN U AND W*

```fortran
DELU = 1.0/FLOAT(NU - 1)
DELW = 1.0/FLOAT(NW - 1)
```

*CALCULATE THE POINTS CORRESPONDING TO DIFFERENT VALUES OF U AND W*

```fortran
U = 0.0
DO 500 I = 1,NU
  M = 0.0
  CALL FMVT(U,VU)
  CALL FMT14(VU,M,F)
  DO 400 J = 1,NW
    CALL FMVT(M,VM)
    CALL FMT14(VM,PXMAT(VU))
    CALL FMT14(VU,S,PT(I,J,1))
    CALL FMT14(VU,PYMAT(VU))
    CALL FMT14(VU,S,PT(I,J,2))
    CALL FMT14(VU,PZMAT(VU))
    CALL FMT11(VU,S,PT(I,J,3))
    M = M + DELM
  400 CONTINUE
  PT(I,J,K) = PT(I,J,K)/36.0
  N = N + DELN
  DO 300 K = 1,3
    PT(I,J,K) = PT(I,J,K)/36.0
  300 CONTINUE
  U = U + DELU
500 CONTINUE
RETURN
END
```

---

**SUBROUTINE FBOTV**

```fortran
SUBROUTINE FBOTV(NVIEW)

REAL*4 MAT(4,4), MAT1(4,4), MAT2(4,4), SCALEF(3)
COMMON /SCLE/ SCALEF
PI = 3.14159
DL = -PI/2.0

*ROTATE THE VIEW ELEMENT ABOUT THE X-AXIS BY -90 DEGREES*
CALL GPROTX(DL,MAT1)

*SCALE THE MODEL*
CALL GPC3(SCALEF,MAT2)
```

---

`SUBROUTINE FBOTV(NVIEW)`

**PROGRAM DESCRIPTION**

This subroutine will display the bottom view of any object.

**BY:** ASHIT R. GANDHI
**DATE:** 11/07/88

**PARAMETERS USED:**
**NVIEW = VIEW IN WHICH THE BOTTOM VIEW IS TO BE DISPLAYED**

**PARAMETERS:**
**PI = 3.14159**
**DL = -PI/2.0**

---

Appendix B - FeatureMod Program Listing 183
SUBROUTINE FBPCH

******************************************************************************
** SUBROUTINE FBPCH(MPTS,NPTS,IM,IM,PNTS,H0,ICOL,LTYPE) **
******************************************************************************
** PROGRAM DESCRIPTION **
** THIS SUBROUTINE WILL DRAW A B-SPLINE SURFACE PATCH FOR **
** CONTROL POINTS SPECIFIED BY THE USER **
** BY: ASHIT R. GANDHI **
** DATE: 11/22/88 **
** PARAMETERS USED: **
** MPTS = NUMBER OF POINTS IN M-DIRECTION (INTEGER, I/P) **
** NPTS = NUMBER OF POINTS IN N-DIRECTION (INTEGER, I/P) **
** IM = FLAG FOR CLOSING IN U-DIRECTION (INTEGER, I/P) **
** IN = FLAG FOR CLOSING IN N-DIRECTION (INTEGER, I/P) **
** PNTS = ARRAY OF CONTROL POINTS DEFINING PATCH (REAL, I/P) **
** H0 = HOMOGENEOUS COORDINATES (REAL, I/P) **
** ICOL = COLOR ATTRIBUTE TO BE USED (INTEGER, I/P) **
** LTYPE = LINE TYPE TO BE USED (INTEGER, I/P) **
******************************************************************************
** SUBROUTINE FBPCH(MPTS,NPTS,IM,IN,PNTS,H0,ICOL,LTYPE) **
******************************************************************************
INTEGER*4 ASNUM,OBNUM,INNUM
REAL*4 PNTS(20,20), H0(20,20), SCALEF(3)
COMMON /SCLE/SCALEF
COMMON /POLY/IPGNR
COMMON /PATCH/MJ,NN
COMMON /IDS/ASMM,OBMM,INMM

SET COUNTER FOR POLYGON NUMBER
IPGN = IPGNR
IF (IPGNR .EQ. 0) THEN
  IPGN = IPGNR
  IPGNR = 1
ENDIF

SET DEFAULT SCALE FACTORS
SCALEF(1) = 1.0
SCALEF(2) = 1.0
SCALEF(3) = 1.0

OPEN CORRECT STRUCTURE AND ATTACH ATTRIBUTES
CALL GPOPST(INNUM)
CALL GPPLCI(ICOL)

COMPUTE THE MATRIX FOR THE SURFACE AND COMPUTE THE POINTS ON IT
CALL FRBMT(IPGNR,IM,IN,MPTS,NPTS,PNTS,H0,ICOL)
CALL GPCLS
IPGNR = IPGN
300 RETURN
END

SUBROUTINE FCOMM

******************************************************************************
** SUBROUTINE FCOMM(IPGN,PGN,ICOL,IOPGN) **
******************************************************************************

Appendix B - FeatureMod Program Listing
**PROGRAM DESCRIPTION**

**THIS ROUTINE SETS UP THE PARAMETERS REQUIRED FOR RENDERING**
A OBJECT OR ASSEMBLY

**BY:** ASHIT R. GANDHI
**DATE:** 01/13/89

**PARAMETERS USED:**

IPGN = POLYGON NUMBER (INTEGER, I/P)
PGN = POLYGON COORDINATES (REAL, I/P)
ICOL = COLOR ATTRIBUTE FOR POLYGON (INTEGER, I/P)
IOPGN = STARTING ELEMENT FOR PART (INTEGER, I/P)

**SUBROUTINE FCOH1(IPGN, PGN, ICOL, IOPGN)**

REAL PGN(3,20000,5), COL(3), XYZSUB(3,20000,5)
INTEGER NPSUB(20000), NPARTS, PRTLMT(2,100), CMPCLR(100)
COMMON /POLAG/ NPARTS, PRTLMT, NSUBT, NPSUB, XYZSUB
COMMON /CPCL/CMPCLR
COMMON /IDS/ ASMM, OBNM, INNM

DATA /DLO/0/

**COMPUTE THE PART NUMBER**
NPARTS = INNM - 10000

**IF A NEW PART THEN COMPUTE FIRST ELEMENT NUMBER**
IF (NOLD .NE. NPARTS) PRTLMT(1, NPARTS) = IOPGN
NOLD = NPARTS

**COMPUTE THE LAST ELEMENT IN THE CURRENT PART**
PRTLMT(2, NPARTS) = IPGN

**ASSIGN COLOR ATTRIBUTE TO CURRENT PART**
CMPCLR(NPARTS) = ICOL

**COMPUTE TOTAL NUMBER OF POLYGONS**
NSUBT = IPGN

**ASSIGN NUMBER OF EDGES TO EACH POLYGON**
DO 100 I = IOPGN, IPGN
NPSUB(I) = 4
100 CONTINUE

**COMPUTE COORDINATES OF POLYGON VERTICES**
DO 400 I = 1,3
   DO 300 J = IOPGN, IPGN
      DO 200 K = 1, NPSUB(J)
         XYZSUB(I, J, K) = PGN(I, J, K)
      200 CONTINUE
   300 CONTINUE
400 CONTINUE

IOPGN = IPGN
RETURN
END

**SUBROUTINE FCTCP(U, B, P)**

**PROGRAM DESCRIPTION**

**THIS ROUTINE WILL CALCULATE THE CONTROL POINTS NEEDED TO CREATE**
A PERIODIC B-SPLINE CURVE THAT PASSES THROUGH FOUR
POINTS SPECIFIED BY THE USER.

**BY:** ASHIT R. GANDHI
**DATE:** 11/12/88

**PARAMETERS USED:**

SUBROUTINE FCTCP(U, B, P)

Appendix B - FeatureMod Program Listing 185
SUBROUTINE FCTCP(U, B, P)
REAL*4 UM(4,4), M(4,4), B1(4,3), P(4,3), VU(4), A(4,4), AINV(4,4)
REAL*4 B1(4), B2(4), B3(4)
REAL*4 P1(4), P2(4), P3(4)

DATA M /-1.0, 3.0,-3.0, 1.0,
     2, 3.0,-6.0, 0.0, 4.0,
     3,-3.0, 3.0, 3.0, 1.0,
     4, 1.0, 0.0, 0.0, 0.0/

FOR EACH VALUE OF U COMPUTE THE PARAMETER VECTOR AND ASSEMBLE THE COMPLETE 4X4 MATRIX
DO 200 I = 1, 4
  CALL FUNVT(U(I), VU)
  DO 100 J = 1, 4
    UM(I, J) = VU(J)
  100 CONTINUE
200 CONTINUE

MULTIPLY MATRIX(U) AND MATRIX(M) TO GET MATRIX(A) AND THEN SCALE MATRIX(A)
CALL FNTV4U(UM, M, A)
DO 400 I = 1, 4
  DO 300 J = 1, 4
    A(I, J) = A(I, J)*27.0
  300 CONTINUE
400 CONTINUE

COMPUTE THE CLOSED FORM INVERSE OF MATRIX(A) AND SCALE MATRIX(AINV)
CALL MATINV(A, AINV)
DO 600 I = 1, 4
  DO 500 J = 1, 4
    AINV(I, J) = AINV(I, J)*162.0
  500 CONTINUE
600 CONTINUE

ASSIGN APPROPRIATE VALUES OF COORDINATES TO THE ARRAY
DO 700 I = 1, 4
  B1(I) = B(I, 1)
  B2(I) = B(I, 2)
  B3(I) = B(I, 3)
700 CONTINUE

COMPUTE THE CONTROL POINTS COORDINATES
CALL FMT4U(AINV, B1, P1)
CALL FMT4U(AINV, B2, P2)
CALL FMT4U(AINV, B3, P3)

ASSIGN APPROPRIATE VALUES OF COORDINATES TO THE ARRAY
DO 800 I = 1, 4
  P1(I, 1) = P1(I)
  P1(I, 2) = P2(I)
  P1(I, 3) = P3(I)
800 CONTINUE

RETURN
END

SUBROUTINE FDASM

********** U = ARRAY OF PARAMETER (U) VALUES CORRESPONDING TO INPUT POINTS **********
********** B = ARRAY OF POINTS THROUGH WHICH THE B-SPLINE CURVE SHOULD PASS **********
********** P = ARRAY OF CONTROL POINTS NEEDED TO GET THE SPECIFIED CURVE **********
********** (REAL/O/P) **********

SUBROUTINE FDASM (IADTO)

PROGRAM DESCRIPTION

THIS ROUTINE CREATES A NEW ASSEMBLY. IT IS ALSO USED TO ADD TO A PREVIOUSLY DEFINED ASSEMBLY.

BY: ASHIT R. GANDHI
DATE: 12/16/88
PARAMETERS USED:

IADTO = FLAG INDICATING NEW OR OLD ASSEMBLY (INTEGER, I/P)

** SUBROUTINE FDASM(IADTO)

INTEGER ASNUM, OBNUM, INNUM, CNUM
REAL ASMAT(4,4), MAT1(4,4), MAT2(4,4), MAT3(4,4)
CHARACTER ANAME*8, OBJECT(9000)*8, STRG*50, WORD(15)*12
COMMON/COMP/ASSEM, OBJECT
COMMON /PRTOF/CNJM
DATA ASMAT/1.0,4*0.0,1.0,4*0.0,1.0,4*0.0,1.0/

IF (IADTO .EQ. 1) GOTO 400

GET NAME OF OBJECT
CALL FPRMP(15)
100 CALL FGSTSTRG)
CALL FPARS(STRG, NM, WORD)
ANAME = WORD(1)(1:8)
IF (ANAME .EQ. 'CANCEL ') RETURN

CHECK TO SEE IF THIS ASSEMBLY ALREADY EXISTS
DO 200 I = 101, ASNUM
   IF (ANAME .EQ. ASSEM(I-100)) THEN
      STRG = 'ASSEMBLY EXISTS. PLEASE ENTER NEW NAME'
      CALL FMSSG(30, STRG)
      GOTO 100
   ENDIF
200 CONTINUE

DO 300 I = 1001, OBNUM
   IF (ANAME .EQ. OBJECT(I-1000)) THEN
      STRG = 'OBJECT WITH SAME NAME EXISTS. PLEASE ENTER NEW NAME'
      CALL FMSSG(30, STRG)
      GOTO 100
   ENDIF
300 CONTINUE

IF NEW OBJECT THEN PROCEED
STRG = 'DEFINE ASSEMBLY //ANAME'
CALL FMSSG(30, STRG)

DEFINE STRUCTURE FOR OBJECT
ASNUM = ASNUM + 1
ASSEM(ASNUM-100) = ANAME
WRITE(31,50) ASNUM, ANAME
CALL GPPOST(ASNUM)
CALL GPMLX3(ASMAT, 1)
CALL GPCLST
CNUM = ASNUM
400 NOBJ = 0
CALL FASSM(NOBJ, OLOC, OORI)
   IF (NOBJ .EQ. 0) RETURN
   IF (NOBJ .EQ. 1) GOTO 400
   CALL FURMT(OORI, MAT1)
   CALL GPTRL3(LOC, MAT1, MAT2, MAT)
   CALL GPCHTS(MAT1, MAT2, MAT)
   CALL GPPOST(CNUM)
   CALL GPMLX3(MAT1, 3)
   CALL GPEXIST(NOBJ)
   CALL GPCLST
   WRITE(22,10) NOBJ, OBJECT(NOBJ-1000), CNUM, ASSEM(CNUM-100)
WRITE(22,20) OORI, OLOC, OORI
GOTO 400
10 FORMAT(1X,2(I4,1X,A8))
20 FORMAT(1X,6(F8.3,1X))
30 FORMAT(1X,13,1X,A8)
SUBROUTINE FDELT

******************************************************************************
** SUBROUTINE FDELT **
** PROGRAM DESCRIPTION **
** THIS SUBROUTINE WILL DELETE A SPECIFIED COMPONENT **
** **
** BY: ASHIT R. GANDHI 
** DATE: 12/18/88 
** PARAMETERS USED: 
** NONE 
**
******************************************************************************

SUBROUTINE FDELT

INTEGER*4 ASMUM, OBSUM, INNUM
CHARACTER*8 ASSEM(900), OBJECT(9000), DCOMP
CHARACTER*15 HORD, STRG
COMMON/IDS/ASSEM, OBSUM, INNUM
COMMON/COMP/ASSEM, OBJECT

* PROMPT USER FOR INPUT
CALL FPRMP(16)

* GET COMPONENT TO BE DELETED
100 CALL FSTSTRG
 CALL FPAR$STRG, HORD)
 DCOMP = WORD(I)(1:8)

* IF "CANCEL" RETURN
 IF (DCOMP .EQ. 'CANCEL') RETURN

* COMPUTE ID OF COMPONENT TO BE DELETED
 DO 200 I = 1, ASMUM
 IF (DCOMP .EQ. ASSEM(I-100)) THEN
 IDDEL = I
 GOTO 400
 ENDIF
 200 CONTINUE

 DO 300 I = 1, OBSUM
 IF (DCOMP .EQ. OBJECT(I-1000)) THEN
 IDDEL = I
 GOTO 400
 ENDIF
 300 CONTINUE

* IF NON EXISTING OBJECT, QUERY FOR NEW NAME
 STRG = 'NON-EXISTING COMPONENT. ENTER NEW NAME.'
 CALL FMSSG(30, STRG)
 GOTO 100

400 CALL GPDLIST(IDDEL)

* COMPLETED REQUEST MESSAGE
 STRG = 'DELETED COMPONENT '/DCOMP
 CALL FMSSG(30, STRG)

500 RETURN

END
SUBROUTINE FDINS(INNUM, LTYPE, INST, NSURF, OBNUM, OBNAME, LOC, DCS)

PROGRAM DESCRIPTION

THIS ROUTINE CREATES NEW FEATURE INSTANCES. IT EXECUTES A
NEW STRUCTURE AND MAKES IT A PART OF THE OBJECT STRUCTURE

BY: ASHIT R. GANDHI

DATE: 12/16/88

PARAMETERS USED:

INNUM = INSTANCE NUMBER
LTYPE = LINE TYPE
INST = NAME OF INSTANCE
NSURF = NUMBER OF PATCHES MAKING UP THE INSTANCE
OBNUM = OBJECT NUMBER THAT INSTANCE IS PART OF
OBNAME = NAME OF PARENT OBJECT
LOC = LOCATION IN OBJECT COORDINATES
ORI = ORIENTATION IN OBJECT COORDINATES

SUBROUTINE FDINS(INNUM, LTYPE, INST, NSURF, OBNUM, OBNAME, LOC, DCS)

NAMELIST /INFO/PLACE, ORIENT

INTEGER4 ASMM, OBNUM, INNUM, LTYPE
REAL4 LOC(3), DCS(3), MAT1(4,4), MAT2(4,4), MAT3(4,4)
REAL4 PLACE(3), ORIENT(3)
CHARACTER4 INST, OBNAME

*COMPUTE THE CORRECT LOCATION AND ORIENTATION

DO 100 I = 1, 3
   PLACE(I) = LOC(I)
   ORIENT(I) = DCS(I)
100 CONTINUE

*ESTABLISH THE CORRECT TRANSFORMATION MATRIX

CALL FURMT(DCS, MAT1)
CALL GPTRL3(LOC, MAT2)
CALL GPCTMS(MAT1, MAT2, MAT3)

*EXECUTE STRUCTURE INSIDE OBJECT STRUCTURE

CALL GPOPST(OBNUM)
CALL GPMLX3(MAT3, 3)
CALL GPCLST(INNUM)
CALL GPCLST
WRITE(10, 10 INNUM, LTYPE, INST, NSURF, OBNUM, OBNAME
WRITE(10, 20) LOC, DCS
RETURN
10 FORMAT(1X, I5, 1X, I2, 1X, A8, 1X, I2, 1X, A8, 1X, A8)
20 FORMAT(1X, 6KF8.3, 1X)

SUBROUTINE FDISP

INTEGER4 ASMM, OBNUM, INNUM
CHARACTER CNAME*8, STRG*50, WORD(15)*12

PROGRAM DESCRIPTION

THIS ROUTINE DISPLAYS THE SPECIFIED OBJECT OR ASSEMBLY.
IF AN UNKNOWN NAME IS GIVEN IT PROMPTS THE USER FOR A NEW NAME.
ALLOWS ONLY ONE COMPONENT TO BE DISPLAYED AT A TIME.

BY: ASHIT R. GANDHI

DATE: 01/13/89

PARAMETERS USED:

NONE

SUBROUTINE FDISP

INTEGER4 ASMM, OBNUM, INNUM
CHARACTER CNAME*8, STRG*50, WORD(15)*12

SUBROUTINE FDISP
*PROMPT USER FOR INPUT
CALL FPRMP{18)

*GET COMPONENT NAME FROM USER
100 CALL FGIST(STRG)
CALL FPARSSTRG,NM,NORD)
CNAME = NORD(1)(1:8)

*IF "CANCEL" RETURN
IF (CNAME .EQ. 'CANCEL ')THEN
RETURN
ENDIF

*CHECK IF COMPONENT ENTERED IS AN OBJECT
NCOMP = 0
DO 200 I = 1001,0BNUM
 IF (CNAME .EQ. OBJECT(I-1000))THEN
 NCOMP = I
 GOTO 400
 ENDIF
200 CONTINUE

*CHECK IF COMPONENT ENTERED IS AN ASSEMBLY
DO 300 I = 101 ASNUM
 IF (CNAME .EQ. ASSEM(I-100))THEN
 NCOMP = I
 GOTO 400
 ENDIF
300 CONTINUE

*IF COMPONENT ENTERED IS NEITHER, THEN PROMPT FOR NEW NAME
STRG = 'NON-EXISTENT COMPONENT. ENTER NEW NAME'
CALL FMSSG(30,STRG)
GOTO 100

*DISASSOCIATE EXISTING COMPONENT FROM THE SCREEN
400 CALL GDPVR{1,7,NOLD)
CALL GDPVR(1,8,NOLD)
CALL GDPVR(1,9,NOLD)
CALL GDPV{1,10,NOLD)
CALL GDPVR(1,11>NOLD,0)

*ASSOCIATE REQUESTED COMPONENT WITH THE SCREEN
CALL GPARV{1,7,NCOMP,0)
CALL GPARV(1,8,NCOMP,0)
CALL GPARV(1,9,NCOMP,0)
CALL GPARV(1,10,NCOMP,0)
CALL GPARV(1,11>NCOMP,0)
CALL GPVP{1,7,2,1)

*COMPLETED REQUESTED MESSAGE
STRG = 'DISPLAYED COMPONENT ' //CNAME
CALL FMSSG{30,STRG)

*SET CURRENTLY VISIBLE STRUCTURE IDS
NOLD = NCOMP
NVIS = NCOMP
RETURN
END
SUBROUTINE FDOBJ

** SUBROUTINE FDOBJIER  
**  
** PROGRAM DESCRIPTION  
**  
** THIS ROUTINE ALLOWS CREATION AND DEFINITION OF AN OBJECT.  
**  
** BY: ASHIT R. GANDHI  
** DATE: 12/16/88  
**  
** PARAMETERS USED:  
**  
** IER = ERROR CODE INDICATING SUCCESS  
**  
**  
***************************************************************

SUBROUTINE FDOBJIER

INTEGER*4 ASNUM, OBNUM, INNUM, CNUM
REAL*4 OBMAT(4,4)
CHARACTER ONAME*8, OBJECT(9000)*8, STRG*50, MORD(15)*12
CHARACTER ASSEM*900)*8
COMMON/COMP:ASSEM,OBJECT
COMMON/IDS: ASNUM, OBNUM, INNUM
COMMON/PRTOF/CNLM
COMMON/VIS/NIUS
DATA OBMAT/1.0 4*0.0;1.0;'s*0.0;1.0;*'•*0.0a1.0/
DATA NOLD/100000/

*GET NAME OF OBJECT
IER = 0
CALL FPRMP(15)
100 CALL FGTST(STRG)
CALL FPARS(STRG,AM,MORD)
ONAME = MORD(11:1:8)

*CHECK TO SEE IF THIS OBJECT ALREADY EXISTS
IF (ONAME .EQ. 'CANCEL') THEN
IER = 1
ENDIF
DO 200 I = 1001, OBNLM
IF (ONAME .EQ. OBJECT(I-1000)) THEN
STRG = 'OBJECT EXISTS. PLEASE ENTER NEW NAME'
CALL FMSSG(30,STRG)
GOTO 100
ENDIF
200 CONTINUE
DO 300 I = 101, ASNUM
IF (ONAME .EQ. ASSEM(I-100)) THEN
STRG = 'ASSEMBLY EXISTS. PLEASE ENTER NEW NAME'
CALL FMSSG(30,STRG)
GOTO 100
ENDIF
300 CONTINUE

*IF NEW OBJECT THEN PROCEED

*DEFINE STRUCTURE FOR OBJECT
OBNUM = OBNUM + 1
CNUM = OBNUM
OBJECT(OBNUM-1000) = ONAME
WRITE(32,10)OBNUM,ONAME
STRG = 'DEFINED OBJECT //'OBJECT(OBNUM-1000)
CALL FMSSG(30,STRG)
CALL GPDBST(OBNUM)
CALL GPMLX3(0BMAT,1)
CALL GPCLST
IF (NOLD .NE. 100000) THEN
SUBROUTINE FDRWU

******************************************************************************
** SUBROUTINE FDRWU(IP, NP, IS, N, MJ, NN, PT)
**
** PROGRAM DESCRIPTION
**
** THIS SUBROUTINE DRAWS THE CONSTANT U LINES FOR A PATCH.
**
** BY: ASHIT R. GANDHI
**
** DATE: 10/12/88
**
** PARAMETERS USED:
**
** IP = THE PATCH NUMBER (INTEGER INPUT)
**
** NP = TOTAL NUMBER OF PATCHES (INTEGER INPUT)
**
** IS = THE SECTION NUMBER (INTEGER INPUT)
**
** N = TOTAL NUMBER OF CROSS SECTIONS (INTEGER INPUT)
**
** MJ = TOTAL NUMBER OF CONSTANT U LINES (INTEGER INPUT)
**
** NN = TOTAL NUMBER OF CONSTANT W LINES (INTEGER INPUT)
**
** PT = ARRAY OF POINTS THROUGH WHICH THE LINES WILL BE DRAWN
**
** (REAL INPUT)
**
******************************************************************************

INTEGER*4 IP, IS, MJ, NN
REAL*4 PT(20, 20, 3), PTS(60)

*DECIDE THE STARTING AND ENDING LINES TO BE DRAWN (TO AVOID REDRAWING)
ISTRT = 1
IEND = NU

*CREATE ARRAY FOR GRAPHICAL INPUT
DO 200 = ISTRT, IEND
   DO 100 M = 1, MJ
      PTS(3*M-2) = PT(L, M, 1)
      PTS(3*M-1) = PT(L, M, 2)
      PTS(3*M) = PT(L, M, 3)
  100 CONTINUE

*DRAW THE LINE
   CALL GPPL3(NN, 3, PTS)
200 CONTINUE

RETURN
END

SUBROUTINE FDRWW

******************************************************************************
** SUBROUTINE FDRWW(IP, NP, IS, N, MJ, NN, PT)
**
** PROGRAM DESCRIPTION
**
** THIS SUBROUTINE DRAWS THE CONSTANT W LINES FOR A PATCH.
**
******************************************************************************
**BY: ASHIT R. GANDHI**
**DATE: 10/12/88**

**PARAMETERS USED:**
**IP = THE PATCH NUMBER (INTEGER, INPUT)**
**NP = TOTAL NUMBER OF PATCHES (INTEGER, INPUT)**
**IS = THE SECTION NUMBER (INTEGER, INPUT)**
**N = TOTAL NUMBER OF CROSS SECTIONS (INTEGER, INPUT)**
**NU = TOTAL NUMBER OF CONSTANT U LINES (INTEGER, INPUT)**
**NW = TOTAL NUMBER OF CONSTANT W LINES (INTEGER, INPUT)**
**PT = ARRAY OF POINTS THROUGH WHICH THE LINES WILL BE DRAWN**
**IN (REAL, INPUT)**

**SUBROUTINE FDRPN(IP, NP, IS, N, NU, NW, PT)**

INT p = INTEGER*4, IP, NP, IS, N, NU, NW, PT
REAL*4 PT(20, 20, 3), PTS(60)

*DECIDE THE STARTING AND ENDING LINES TO BE DRAWN (TO AVOID REDRAWING)*
IF (IP .EQ. NP .AND. IPEND .EQ. NW - 1)
   ISTART = 1
   IEND = NW

*CREATE ARRAY FOR GRAPHICAL INPUT*
DO 200 M = ISTART, IEND
   DO 100 L = 1, NW
      PTS(3*M-2) = PT(L, M, 1)
      PTS(3*M-1) = PT(L, M, 2)
      PTS(3*M ) = PT(L, M, 3)
   100 CONTINUE

*DRAW THE LINE*
CALL GPPL3(N, S, PTS)
200 CONTINUE
RETURN
END

---

**SUBROUTINE FEDAT**

**DESCRIPTION:**
This subroutine sets up polygon edge attributes.

**BY:** Ashit R. Gandhi
**DATE:** 10/07/88

**PARAMETERS USED:**
**IEDEL : EDGE FLAG 1=OFF, 2=ON**
**IEDLT : EDGE LINE TYPE**

**SUBROUTINE FEDAT (IEDEL, IEDLT, IEDCL)**

SET UP EDGE FLAG
CALL GPEF(IEDEL)

SET EDGE LINE TYPE
CALL GPELT(IEDLT)

SET EDGE LINE COLOR
CALL GPEC(IECL)
RETURN
END
**SUBROUTINE FFNTV**

******************************************************************************
** SUBROUTINE FFNTV(NVIEW) **
** **
** PROGRAM DESCRIPTION **
** **
** THIS SUBROUTINE WILL DISPLAY THE TOP VIEW OF ANY OBJECT. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 11/07/88 **
** **
** PARAMETERS USED: **
** **
** NVIEW = VIEW IN WHICH THE TOP VIEW IS TO BE DISPLAYED **
** (INTEGER, I/P) **
** **
******************************************************************************

SUBROUTINE FFNTV(NVIEW)

REAL*6 MAT(4,4), MAT1(4,4), MAT2(4,4), SCALEF(3)
COMPLEX /SCALF/SCALEF
PI = 3.14159
DL = 0.0

*ROTATE THE VIEW ELEMENT ABOUT THE X-AXIS BY +90 DEGREES
CALL GPROTX(DL,MAT2)

*SCALE THE MODEL
CALL GSCP3(SCALEF,MAT1)

*COMPOSE THE TRANSFORMATION
CALL GPCMT3(MAT2,MAT1,MAT)

*ASSOCIATE THIS ROTATION TO THE VIEW
CALL GPVM3(1,NVIEW,MAT)
CALL GPVPI(1,NVIEW,2,1)
RETURN
END

******************************************************************************
** SUBROUTINE FGTST **
** **
** PROGRAM DESCRIPTION **
** **
** THIS SUBROUTINE GETS A STRING INPUT FROM THE USER. THE STRING **
** INPUT IS TAKEN FROM AN ACTIVATED AREA RESERVED FOR SUCH INPUT **
** **
** BY: ASHIT R. GANDHI **
** DATE: 11/06/88 **
** **
** PARAMETERS USED: **
** **
** STRG = STRING INPUT (CHARACTER, O/P) **
** **
******************************************************************************

SUBROUTINE FGTST(STRG)

COMMON CSIZE
INTEGER*4 L
REAL*4 CSIZE(3), AREA(6)
CHARACTER STRG(*), EMTY*50, PREF*9, MSSG*59

DATA EMTY/' '/
STRG = ' 

*PUT STRING IN REQUEST MODE
CALL GPSTMO(1,1,1,2)

*GET STRING
L=50
100 CALL GPRGST(1,1,50,2,LR,STRG)
IF (LR .EQ. 0) GOTO 100

*CONVERT STRING TO ALL CAPITALS
CALL FCAPSSTRG
CALL GPVP(1,6,2,1)

*PUT STRING BACK IN EVENT MODE
CALL GPSTMO(1,1,1,2)
RETURN
END

** SUBROUTINE FINIT **

*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE INITIALIZES ALL INPUT DEVICES. ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 10/26/88 ***
*** PARAMETERS USED: ***
*** **

SUBROUTINE FINIT
INTEGER*4 PPATH(3)
REAL*4 ASIZE(6),CSIZE(3),DATA(12)
REAL*4 PAREA(6),SAREA(6),VARA(6)
CHARACTER**50
DATA EMTY/' '/

INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1,ERRIND,UNIT$,CSIZE,ASIZE)

PREPARE FOR PICK INPUT
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.35*CSIZE(1)
PAREA(3)=0.15*CSIZE(2)
PAREA(4)=0.5*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

PREPARE FOR STRING INPUT
CALL GPSTM0(1,1,1,2)

PLACE STRING DEVICE IN THE REQUEST MODE
CALL GPSTMO(1,1,1,2)
SUBROUTINE FINPT

*SETUP STRING INPUT AREA

<table>
<thead>
<tr>
<th>AREA</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.007*CSIZE(1)</td>
</tr>
<tr>
<td>2</td>
<td>0.845*CSIZE(1)</td>
</tr>
<tr>
<td>3</td>
<td>0.15*CSIZE(2)</td>
</tr>
<tr>
<td>4</td>
<td>0.17*CSIZE(2)</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>CSIZE(3)</td>
</tr>
</tbody>
</table>

*INITIALIZE THE STRING DEVICE

CALL GPINST(1,1,50,EMTY,1,SAREA,50,1,0,EMTY)

*PLACE STRING DEVICE IN THE EVENT MODE

CALL GPSTMO(1,1,3,2)

*DEFINE PICK AREA

<table>
<thead>
<tr>
<th>AREA</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>CSIZE(1)</td>
</tr>
<tr>
<td>3</td>
<td>CSIZE(2)</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>CSIZE(3)</td>
</tr>
</tbody>
</table>

*PLACE VALUATOR DEVICES IN THE EVENT MODE

*INITIALIZE VALUATORS TO GET VALUES FROM 0 TO 360

DO 100 I = 1,3
  VAREA(3) = CSIZE(2) - 0.01*I
  CALL GPVLMO(1,I,1,1)
  CALL GPINVL(1,I,0,1,VAREA,0,360,0,DATA)
  CALL GPVLMO(1,I,3,2)
100 CONTINUE

VAREA(3) = CSIZE(2) - 0.01*4
CALL GPVLMO(1,4,1,1)
CALL GPINVL(1,4,1,1,VAREA,0.01,5,0,DATA)
CALL GPVLMO(1,4,3,2)
RETURN
END

SUBROUTINE FINPT

PROGRAM DESCRIPTION

THIS SUBROUTINE WILL WAIT FOR AN INPUT FROM THE USER. ONCE THE
INPUT IS GIVEN IT WILL RETURN INFORMATION ABOUT THE KIND OF
INPUT THAT WAS RECEIVED AND ITS VALUES TO BE FURTHER PROCESSED.

BY: ASHIT R. GANDHI
DATE: 10/26/88

PARAMETERS USED:

PMENU = MENU LIST THAT IS CURRENTLY DISPLAYED
ICLASS = CLASS OF INPUT DEVICE (INTEGER,O/P)
IDEV = DEVICE NUMBER IF VALUATOR INPUT (INTEGER,O/P)
IPKID = PICK-ID OF ITEM, IF PICKED FROM MENU (INTEGER,O/P)
V = ARRAY OF VALUES FROM VALUATOR (REAL(8),O/P)
STRG = STRING VALUE, IF STRING INPUT (CHARACTER*50,O/P)

SUBROUTINE FINPT(PMENU,ICLASS,IDEV,IPKID,V,STRG)

INTEGER*4 CLASS: YÜDE, ECHO, IPKID, PPATH(3)
REAL*4 PAREAI6),C$IZE(S),A$IZE(6), DATAKIZ)
REAL*4 SAREA(6),VAREA(6) V(8)
CHARACTER STRG*50, EMTY*50, PREF*9, MSGS*59, PMENU(50)*15

Appendix B - FeatureMod Program Listing
SET PICK FILTER TO MAKE CLASS DETECTABLE
CALL GPPKF(1,1,1,CLASS,1,2)

WAIT AN EVENT
200 CALL GPAHEV(1,1,ICLASS,IDEV)

*IF INPUT CLASS IS "LOCATOR"
  IF (ICLASS .EQ. 1) THEN
    MSG = 'LOCATOR NOT IMPLEMENTED'
    CALL FMSSG(30,MSG)
  ENDIF

*IF INPUT CLASS IS "STROKE"
  IF (ICLASS .EQ. 2) THEN
    MSG = 'STROKE NOT IMPLEMENTED'
    CALL FMSSG(30,MSG)
  ENDIF

*IF INPUT CLASS IS "VALUATOR"
  IF (ICLASS .EQ. 3) THEN
    GET THE VALUE FROM THE VALUATOR
    CALL GPGLVL(VLU)
    UPDATE ARRAY OF VALUES FOR VALUATOR
    V(IDEV) = VLU
  ENDIF

*IF INPUT CLASS IS "CHOICE"
  IF (ICLASS .EQ. 4) THEN
    MSG = 'CHOICE NOT IMPLEMENTED'
    CALL FMSSG(30,MSG)
  ENDIF

*UPDATE WORKSTATION (TRAVERSE ALL VIEWS)
  CALL GPUPWS(1,2)
*GET PICK
  CALL GPGT PK(1,1,PPATH)
*IPKID IS SECOND OF PPATH
  IPKID = PPATH(2)
  IF (IPKID .EQ. 1) GOTO 200
ENDIF

*IF INPUT CLASS IS "STRING"
  IF (ICLASS .EQ. 6) THEN
    WRITE(6,*) 'STRING INPUT'
    L = 50
*GET THE STRING INPUT*

```fortran
CALL GPRST(L,LR,STRG)
IF (LR .EQ. 0) GOTO 200
CALL FPARS(STRG,NN,NORD)
CALL FCAPS(NORD)
```

*ECHO THE STRING INPUT*

```fortran
IF (MORDER(1:15) .EQ. 'HCOPY') THEN
  CALL HRDCPY(1,2)
  CALL GPUPNS(2,2)
  CALL HRDCPY(1,2)
  CALL GPUPNS(2,2)
  STRG = 'HARD COPY GENERATED'
  CALL FMSSG(30,STRG)
END IF
```

RETURN
END

### SUBROUTINE FINGU

```fortran
** SUBROUTINE FINGU(FNAME,TYPE) **
** **
** PROGRAM DESCRIPTION **
** THIS SUBROUTINE ALLOWS A USER TO INQUIRE INFORMATION ON **
** DIFFERENT FEATURES. DIFFERENT LEVELS AND TYPES OF INQUIRIES **
** ARE POSSIBLE. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 01/30/89 **
** **
** PARAMETERS USED: **
** FNAME : FEATURE NAME ON WHICH INFORMATION IS REQUESTED. (I/P, CHARACTER*)
** TYPE : TYPE AND LEVEL OF INQUIRY REQUESTED. (I/P, CHARACTER*3) **
** **
** SUBROUTINE FINGU(FNAME,TYPE)
** **
** CHARACTER FNAME*8, LCHAR*73, HEAD(6)*12, TYPE*3
** DATA HEAD /'FEATURE NAME', 'ENGLISH DEFINITION', 'GROUP CLASSIFICATION',
** 1 'SYNONYMS', 'MATHEMATICAL', 'DEFAULT MODE'/
** DATA NHEAD/6/
** DATA NIN,NOUT/B,10/
** IFLAG = 0
** DO 100 I = 1,20000
** READ(NIN,10) LCHAR
** IF (LCHAR(1:12) .EQ. HEAD(1)) THEN
** IF (LCHAR(16:23) .EQ. FNAME) THEN
** IFLAG = 1
** GOTO 200
** END IF
** END IF
** 100 CONTINUE
** ELSE
** WRITE(NOUT,*) 'FEATURE DOES NOT EXIST IN LIBRARY'
** GOTO 500
** END IF
** IF (TYPE .NE. 'ALL') THEN
** DO 300 I = 1,1000
** READ(NIN,10) LCHAR
** IF (TYPE .EQ. LCHAR(1:3)) THEN
** IFLAG = 2
** WRITE(NOUT,*) LCHAR
** END IF
** IF (LCHAR(1:3) .EQ. 'END') GOTO 450
** 300 CONTINUE
** ELSE
** DO 400 I = 1,1000
** READ(NIN,10) LCHAR
** WRITE(NOUT,*) LCHAR
** IFLAG = 2
** 400 CONTINUE
** 500 CONTINUE
** END
** **
** END SUBROUTINE FINGU **
```

Appendix B - FeatureMod Program Listing
SUBROUTINE FLFTV

*** SUBROUTINE FLFTV(NVIEW) ***
*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE WILL DISPLAY THE LEFT SIDE VIEW OF ANY OBJECT. ***
*** ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/07/88 ***
*** PARAMETERS USED: ***
*** NST = STRUCTURE ASSOCIATED WITH THE VIEW (INTEGER I/P) ***
*** NVIEW = VIEW IN WHICH THE TOP VIEW IS TO BE DISPLAYED ***
*** (INTEGER I/P) ***

SUBROUTINE FLFTV(NVIEW)

REAL*4 MAT(4,4); MAT1(4,4), MAT2(4,4), SCLEF(3)
COMMON/SCLE/SCALEF
PI = 3.14159
DL = PI/2.0

*ROTATE THE VIEW ELEMENT ABOUT THE Y-AXIS BY +90 DEGREES
CALL GPROTY(DL,MAT2)

*SCALE THE MODEL
CALL GPSC3(SCALEF,MAT1)

*COMPOSE THE TRANSFORMATION
CALL GPCMT3(MAT2,MAT1,MAT)

*ASSOCIATE THIS ROTATION TO THE VIEW
CALL GPVM3(1,NVIEW,MAT)
CALL GPVP(1,NVIEW,2,1)
RETURN
END

SUBROUTINE FMENU2

*** SUBROUTINE FMENU2(NUM,PMENU) ***
*** PROGRAM DESCRIPTION ***
*** THIS ROUTINE DISPLAYS THE REQUIRED MENU ON THE SCREEN. ***
*** IT RETURNS A STRING OF MENU ITEMS THAT ARE CURRENTLY ***
*** BEING DISPLAYED. ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 10/12/88 ***
*** PARAMETERS USED: ***
** COMMON CSIZE
REAL*4 POS(2), CSIZE(6), PAREA(6)
INTEGER*4 CHOICE, PPATH(3)
CHARACTER MENU(50,50)*15, MSG*15, PMENU(50)*15
*  TITLE TEST,LENGTH

* DATA LNT/15/
DATA MENU1,1/ ' FILES '/'
DATA MENU1,2/ ' ---- '/'
DATA MENU1,3/ ' COMMANDS '/'
DATA MENU1,4/ '< file '/'
DATA MENU1,5/ ' COMMAND FILE '/'
DATA MENU1,6/ ' COMMANDS '/'
DATA MENU1,7/ ' ---- '/'
DATA MENU1,8/ ' EXIT '/'
DATA MENU1,9/ ' BACKGROUNDS '/'
DATA MENU1,10/ ' LANGUAGE I/P '/'
DATA MENU1,11/ ' RENDERING '/'
DATA MENU1,12/ ' VIEWS '/'
DATA MENU1,13/ ' WINDOWS '/'
DATA MENU1,14/ ' DISPLAY '/'
DATA MENU1,15/ ' FILES '/'
DATA MENU1,16/ ' FILES '/'
DATA MENU1,17/ ' FILES '/'
DATA MENU1,18/ ' FILES '/'
DATA MENU1,19/ ' FILES '/'
DATA MENU1,20/ ' FILES '/'
DATA MENU1,21/ ' FILES '/'
DATA MENU1,22/ ' FILES '/'
DATA MENU1,23/ ' FILES '/'
DATA MENU1,24/ ' FILES '/'
DATA MENU1,25/ ' FILES '/'
DATA MENU1,26/ ' FILES '/'
DATA MENU1,27/ ' FILES '/'
DATA MENU1,28/ ' FILES '/'
DATA MENU1,29/ ' FILES '/'
DATA MENU1,30/ ' FILES '/'
DATA MENU1,31/ ' FILES '/'
DATA MENU1,32/ ' FILES '/'
DATA MENU1,33/ ' FILES '/'
DATA MENU1,34/ ' FILES '/'
DATA MENU1,35/ ' FILES '/'
DATA MENU1,36/ ' FILES '/'
DATA MENU1,37/ ' FILES '/'
DATA MENU1,38/ ' FILES '/'
DATA MENU1,39/ ' FILES '/'
DATA MENU1,40/ ' FILES '/'
DATA MENU1,41/ ' FILES '/'
DATA MENU1,42/ ' FILES '/'
DATA MENU1,43/ ' FILES '/'
DATA MENU1,44/ ' FILES '/'
DATA MENU1,45/ ' FILES '/'
DATA MENU1,46/ ' FILES '/'
DATA MENU1,47/ ' FILES '/'
DATA MENU1,48/ ' FILES '/'
DATA MENU2,1/ ' COMMANDS '/'
DATA MENU2,2/ ' ---- '/'
DATA MENU2,3/ ' EXIT '/'
DATA MENU2,4/ ' BACKGROUNDS '/'
DATA MENU2,5/ ' LANGUAGE I/P '/'
DATA MENU2,6/ ' RENDERING '/'
DATA MENU2,7/ ' VIEWS '/'
DATA MENU2,8/ ' WINDOWS '/'
DATA MENU2,9/ ' DISPLAY '/'
DATA MENU2,10/ ' INQUIRE '/'
DATA MENU2,11/ ' H.COPY '/'
DATA MENU2,12/ ' FILES '/'
DATA MENU2,13/ ' FILES '/'
DATA MENU2,14/ ' FILES '/'
DATA MENU2,15/ ' FILES '/'
DATA MENU2,16/ ' FILES /'

Appendix B - FeatureMod Program Listing
Appendix B - FeatureMod Program Listing

DATA THMENU(6, 38)';
DATA THMENU(6, 15)';
DATA THMENU(6, 19)';
DATA THMENU(6, 16)';
DATA THMENU(6, 40)';
DATA THMENU(6, 27)';
DATA THMENU(6, 41)';
DATA THMENU(6, 18)';
DATA THMENU(6, 42)';
DATA THMENU(6, 19)';
DATA THMENU(6, 43)';
DATA THMENU(6, 20)';
DATA THMENU(6, 44)';
DATA THMENU(6, 21)';
DATA THMENU(6, 45)';
DATA THMENU(6, 46)';
DATA THMENU(6, 23)';
DATA THMENU(6, 47)';
DATA THMENU(6, 24)';
DATA THMENU(6, 48)';
DATA THMENU(7, 1)' SHADING
DATA THMENU(7, 25)';
DATA THMENU(7, 27)';
DATA THMENU(7, 26)';
DATA THMENU(7, 27)';
DATA THMENU(7, 28)';
DATA THMENU(7, 29)';
DATA THMENU(7, 3)' RETURN
DATA THMENU(7, 4)';
DATA THMENU(7, 5)' CONSTANT
DATA THMENU(7, 6)';
DATA THMENU(7, 7)' RESET
DATA THMENU(7, 8)';
DATA THMENU(7, 9)' SINGLE
DATA THMENU(7, 10)';
DATA THMENU(7, 11)';
DATA THMENU(7, 12)';
DATA THMENU(7, 13)';
DATA THMENU(7, 14)';
DATA THMENU(7, 15)';
DATA THMENU(7, 16)';
DATA THMENU(7, 17)';
DATA THMENU(7, 18)';
DATA THMENU(7, 19)';
DATA THMENU(7, 20)';
DATA THMENU(7, 21)' FRONT
DATA THMENU(7, 22)';
DATA THMENU(7, 23)' TOP
DATA THMENU(7, 24)';
DATA THMENU(7, 25)' RIGHT SIDE
DATA THMENU(7, 26)' ISOMETRIC
DATA THMENU(7, 27)' BACK
DATA THMENU(7, 28)' BOTTOM
DATA THMENU(7, 29)' LEFT SIDE
DATA THMENU(7, 30)' BOTTOM
DATA THMENU(7, 31)' LEFT SIDE
DATA THMENU(7, 32)' BACK
DATA THMENU(7, 33)' RIGHT SIDE
DATA THMENU(7, 34)' TOP
DATA THMENU(7, 35)' ISOMETRIC
DATA THMENU(7, 36)' FRONT
DATA THMENU(7, 37)' TOP
DATA THMENU(7, 38)' RIGHT SIDE
DATA THMENU(7, 39)' ISOMETRIC
DATA THMENU(7, 40)' FRONT
DATA THMENU(7, 41)' TOP
DATA THMENU(7, 42)' RIGHT SIDE
DATA THMENU(7, 43)' ISOMETRIC
DATA THMENU(7, 44)' FRONT
DATA THMENU(7, 45)' TOP
DATA THMENU(7, 46)' RIGHT SIDE
DATA THMENU(7, 47)' ISOMETRIC
DATA THMENU(7, 48)' FRONT

Appendix B - FeatureMod Program Listing 203
DATA T-menu(8,37) /* */
DATA T-menu(8,36) /* */
DATA T-menu(8,39) /* */
DATA T-menu(8,16) /* */
DATA T-menu(8,17) /* */
DATA T-menu(8,8) /* */
DATA T-menu(8,18) /* */
DATA T-menu(8,4) /* */
DATA T-menu(8,19) /* */
DATA T-menu(8,20) /* */
DATA T-menu(8,40) /* */
DATA T-menu(8,21) /* */
DATA T-menu(8,8) /* */
DATA T-menu(8,22) /* */
DATA T-menu(8,41) /* */
DATA T-menu(8,23) /* */
DATA T-menu(8,24) /* */
DATA T-menu(8,42) /* */
DATA T-menu(8,25) /* */
DATA T-menu(8,43) /* */
DATA T-menu(8,26) /* */
DATA T-menu(8,44) /* */
DATA T-menu(8,27) /* */
DATA T-menu(8,45) /* */
DATA T-menu(8,28) /* */
DATA T-menu(8,46) /* */
DATA T-menu(8,29) /* */
DATA T-menu(8,47) /* */
DATA T-menu(8,30) /* */
DATA T-menu(8,48) /* */

DATA T-menu(9,1) /* COLOR */
DATA T-menu(9,25) /* ----- */
DATA T-menu(9,1) /* RETURN */
DATA T-menu(9,27) /* */
DATA T-menu(9,4) /* */
DATA T-menu(9,28) /* MAXIMUM RED */
DATA T-menu(9,29) /* */
DATA T-menu(9,6) /* MAXIMUM GREEN */
DATA T-menu(9,30) /* */
DATA T-menu(9,7) /* MAXIMUM BLUE */
DATA T-menu(9,31) /* */
DATA T-menu(9,8) /* MAXIMUM YELLOW */
DATA T-menu(9,32) /* */
DATA T-menu(9,9) /* MAGENTA */
DATA T-menu(9,33) /* */
DATA T-menu(9,10) /* TURQUOISE */
DATA T-menu(9,34) /* */
DATA T-menu(9,11) /* DARK MAGENTA */
DATA T-menu(9,35) /* */
DATA T-menu(9,12) /* DARK GREEN */
DATA T-menu(9,36) /* */
DATA T-menu(9,13) /* DARK BLUE */
DATA T-menu(9,37) /* */
DATA T-menu(9,14) /* BACKGROUND GREY */
DATA T-menu(9,38) /* */
DATA T-menu(9,15) /* LIGHT GREY */
DATA T-menu(9,39) /* */
DATA T-menu(9,16) /* PEACOCK GREEN */
DATA T-menu(9,40) /* */
DATA T-menu(9,17) /* ORANGE */
DATA T-menu(9,41) /* */
DATA T-menu(9,18) /* BLACK */
DATA T-menu(9,42) /* */
DATA T-menu(9,19) /* */
DATA T-menu(9,43) /* */
DATA T-menu(9,20) /* RESET */
DATA T-menu(9,44) /* */
DATA T-menu(9,21) /* */
DATA T-menu(9,45) /* */
DATA T-menu(9,22) /* */
DATA T-menu(9,46) /* */
DATA T-menu(9,23) /* */
DATA T-menu(9,47) /* */
DATA T-menu(9,24) /* */
DATA T-menu(9,48) /* */

DATA T-menu(10,1) /* DEFINE */
DATA T-menu(10,25) /* ----- */
DATA T-menu(10,26) /* */
DATA T-menu(10,3) /* return */
DATA T-menu(10,27) /* */
DATA T-menu(10,4) /* */
DATA T-menu(10,28) /* SLAB */
DATA T-menu(10,29) /* */
DATA T-menu(10,6) /* SPHERE */
DATA T-menu(10,30) /* */
DATA T-menu(10,7) /* CYLINDER */
DATA T-menu(10,31) /* */
DATA T-menu(10,8) /* RING */
DATA T-menu(10,32) /* */
DATA T-menu(10,9) /* CONE */
DATA T-menu(10,33) /* */
DATA T-menu(10,10) /* PYRAMID */
DATA T-menu(10,34) /* */
DATA T-menu(10,11) /* WEDGE */
DATA T-menu(10,35) /* */
DATA T-menu(10,12) /* ELLIPSOID */

Appendix B - FeatureMod Program Listing
SUBROUTINE FMPK1

:: SUBROUTINE FMPK1(MNUM,IPKID,PMENU) ::
:: PROGRAM DESCRIPTION ::
** IF AN ITEM WAS PICKED FROM MENU NUMBER 1, THEN THE CORRECT **
** ACTION SHOULD BE TAKEN. **
** BY: ASHIT R. GANDHI **
** DATE: 11/04/88 **
** PARAMETERS USED: **
** MNUM = MENU NUMBER (INTEGER, I/P) **
** IPKID = 1
** PICK ID OF THE ITEM THAT WAS PICKED (INTEGER,I/P)
**
** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU
** (CHARACTER,I/P,O/P)
**
** SUBROUTINE FMPK1(MNUM,IPKID,PMENU)
**
** INTEGER*4 MNUM, IPKID
** REAL*4 PMENU(50)
** LOGICAL*4 FLCHK
** CHARACTER STRG*50, PRE*30, FTRG*4, ANS*1, LCHAR*72
** CHARACTER*8 FILEIO, FILE21, FILE22, FILE31, FILE52
** COMMON/POLY/IPGNR
** COMMON/MODEL/FILE22
** DATA FLFLAG/O/

*GOTO THE CORRECT ID THAT WAS PICKED
1  GOTO(11,12,13,14,15,16,17,18,19,20,21)IPKID
*IF IPKID .GT. 11 RETURN
   RETURN
*MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN
*MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN
*MNUM = 1, IPKID = 3 ITEM PICKED = COMMANDS
*IN THIS CASE JUST DISPLAY MENU NUMBER 2
13  MNUM = 2
    CALL FMENU2(MNUM,PMENU)
    RETURN
*MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN
*MNUM = 1, IPKID = 5 ITEM PICKED = START
*IN THIS CASE ASK FOR STRING INPUT
*CHECK IF A MODEL FILE ALREADY OPEN (FLFLAG = 0, NO FILE OPEN)
* (FLFLAG = 1, FILE OPEN)
15  IF (FLFLAG .EQ. 1) THEN
   STRG = 'MODEL FILE IS CURRENTLY OPEN'
   CALL FMSSGI(30,STRG)
   STRG = 'IF YOU OPEN A NEW FILE, PREVIOUS WORK'
   CALL FMSSGI(30,STRG)
   STRG = 'DONE IN THIS SESSION WILL BE LOST'
   CALL FMSSGI(30,STRG)
   CALL FGSTANS
   IF (ANS .NE. 'Y') RETURN
   CLOSE(10)
   CALL GPDLST(20)
   CALL GPDLST(21)
   ENDIF
*IF FILE IS TO BE OPENED GET THE FILE NAME
151 CALL FPRM(3)
    CALL FGSTI(FTRG)
    DO 152 I = 1,4
        IF (FTRGI:I) .EQ. '.' FTRGI:I) = '0'
152 CONTINUE
*CHECK IF THIS FILE ALREADY EXISTS
    FILE10 = FTRG//0010
    INQUIRE(FILE=FILE10,EXIST=FLCHK)
    IF (FLCHK) THEN
        STRG = 'MODEL FILE ALREADY EXISTS'
        CALL FMSSGI(30,STRG)
        STRG = 'CANNOT OPEN EXISTING FILE'
        CALL FMSSGI(30,STRG)
        STRG = 'ENTER NEW NAME'
        CALL FMSSGI(30,STRG)
        GOTO 151
    ENDIF
*IF A NEW FILE HAS TO BE CREATED
FILE21 = FTRG//'0021'
FILE22 = FTRG//'0022'
FILE31 = FTRG//'0031'
FILE32 = FTRG//'0032'
OPEN(UNIT=10,FILE=FILE10,STATUS='NEW')
OPEN(UNIT=21,FILE=FILE21,STATUS='NEW')
OPEN(UNIT=22,FILE=FILE22,STATUS='NEW')
OPEN(UNIT=31,FILE=FILE31,STATUS='NEW')
OPEN(UNIT=32,FILE=FILE32,STATUS='NEW')

*SET FILE FLAG ON
FILEFLAG = 1
IPGNR = 0
STRG = 'STARTED MODEL'//FTRG
CALL FMSSG(30,STRG)
RETURN

*IPKID = 6 ITEM PICKED = RECALL
*IN THIS CASE ASK FOR STRING INPUT
*CHECK IF A MODEL FILE ALREADY OPEN (FILEFLAG = 0, NO FILE OPEN)
* (FILEFLAG = 1, FILE OPEN)
16 IF (FILEFLAG .EQ. 1) THEN
  STRG = 'MODEL FILE IS CURRENTLY OPEN'
  CALL FMSSG(30,STRG)
  STRG = 'IF YOU RECALL A NEW FILE, PREVIOUS WORK'
  CALL FMSSG(30,STRG)
  STRG = 'DONE IN THIS SESSION WILL BE LOST'
  CALL FMSSG(30,STRG)
  CALL FPMPB()
  CALL FGTST(FTRG)
  IF (ANS .NE. 'Y') RETURN
  CLOSE(10)
  CALL GDLSTIZ0)
  CALL GDLSTIZ1)
  ENDIF
161 CALL FGTST(FTRG)
162 DO 162 I = 1,6
  IF (FTRG(I:I) .EQ. ' ') FTRG(I:I) = '0'
  CONTINUE

*CHECK IF THIS FILE EXISTS
FILE10 = FTRG//'0010'
FILE21 = FTRG//'0021'
FILE22 = FTRG//'0022'
FILE31 = FTRG//'0031'
FILE32 = FTRG//'0032'
I10 = 0
INQUIRE(FILE=FILE10,EXIST=FLCHK)
IF (FLCHK) THEN
  I10 = 1
  OPEN(UNIT=10,FILE=FILE10,STATUS='OLD')
ELSE
  STRG = 'MODEL FILE DOES NOT EXIST'
  CALL FMSSG(30,STRG)
  STRG = 'CANNOT RECALL NEW EXISTING FILE'
  CALL FMSSG(30,STRG)
  STRG = 'ENTER NEW NAME'
  CALL FMSSG(30,STRG)
GOTO 161
ENDIF
INQUIRE(FILE=FILE21,EXIST=FLCHK)
IF (FLCHK) THEN
  I21 = 1
  OPEN(UNIT=21,FILE=FILE21,STATUS='OLD')
ELSE
  I21 = 0
  OPEN(UNIT=21,FILE=FILE21,STATUS='NEW')
ENDIF
INQUIRE(FILE=FILE22,EXIST=FLCHK)
IF (FLCHK) THEN
  I22 = 1
  OPEN(UNIT=22,FILE=FILE22,STATUS='OLD')
ELSE
  I22 = 0
  OPEN(UNIT=22,FILE=FILE22,STATUS='NEW')
ENDIF
INQUIRE(FILE=FILE31,EXIST=FLCHK)
IF (FLCHK) THEN
  I31 = 1
  OPEN(UNIT=31,FILE=FILE31,STATUS='OLD')
ELSE
  I31 = 0
  OPEN(UNIT=31,FILE=FILE31,STATUS='NEW')
ENDIF
ENDIF
INQUIRE(FILE=FILE32,EXIST=FLCHK)
IF (FLCHK) THEN
  I32 = 1
  OPEN(UNIT=32,FILE=FILE32,STATUS='OLD')
ELSE
  I32 = 0
  OPEN(UNIT=32,FILE=FILE32,STATUS='NEW')
ENDIF

*IF A OLD FILE HAS TO BE RECALLED
STRG = 'RECALLING MODEL '//FTRG
CALL FMMSG(30,STRG)

*USE RECALL PROCEDURE TO BRING UP MODEL
CALL FRSET
CALL FRCALL(i10,i21,i22,i31,i32)

FLFLAG = 1
STRG = 'MODEL RECALLED'
CALL FMMSG(30,STRG)
RETURN

*MNUN = 1, IPKID = 7 IS NOT A POSSIBILITY
17 RETURN

*MNUN = 1, IPKID = 8 ITEM PICKED = FILE
*IN THIS CASE, ASK FOR STRING INPUT

*CHECK IF A MODEL FILE ALREADY OPEN (FLFLAG = 0; NO FILE OPEN)
* (FLFLAG = 1, FILE OPEN)
18 IF (FLFLAG .EQ 0) THEN
  STRG = 'NO MODEL FILE EXISTING'
  CALL FMMSG(30,STRG)
  STRG = 'CANNOT STORE EMPTY MODEL'
  CALL FMMSG(50,STRG)
  RETURN
ENDIF

*INQUIRE NAME OF CURRENT MODEL
INQUIRE(UNIT=10,NAME=FTRG)
STRG = 'CURRENT MODEL NAME IS '//FTRG
CALL FMMSG(30,STRG)
CALL FPRMPl9)
CALL FGDS(10)
IF (ANS .EQ 'Y') THEN
  STRG = 'SAVING MODEL UNDER FILE '//FTRG
  CALL FMMSG(30,STRG)
  CALL GPDLST(20)
  CLOSE(10)
  FLFLAG = 0
  STRG = 'MODEL SAVED'
  CALL FMMSG(30,STRG)
  RETURN
ENDIF

CALL FPRMPl5)
FLFLAG = 0
CALL FGDS(10)
CALL GPDLST(20)

*USE STORE PROCEDURE TO SAVE UP MODEL
* CALL FSAVE

*MNUN = 1, IPKID = 9 ITEM PICKED = PARAMETER
19 STRG = 'PARAMETER OPTION NOT IMPLEMENTED'
CALL FMMSG(30,STRG)
RETURN

*MNUN = 1, IPKID = 10 IS NOT A POSSIBILITY
20 RETURN

*MNUN = 1, IPKID = 11 ITEM PICKED = COMMAND FILE
21 CALL FPRMPl6)
CALL FGDS(FTRG)
STRG = 'EXECUTING COMMAND FILE '//FTRG
CALL FMMSG(30,STRG)

Appendix B - FeatureMod Program Listing 208
*USE EXECUTE FILE PROCEDURE TO CREATE UP MODEL
* CALL FEXEC
  STRG = 'COMMAND FILE NOT IMPLEMENTED'
  CALL FMSSG(10,STRG)
  RETURN
16t1 FORMAT(A72)
END

SUBROUTINE FMPK10

*********************************************************************************************
** SUBROUTINE FMPK10(NMUM,IPKID,PMENU)
** PROGRAM DESCRIPTION
** IF AN ITEM HAS PICKED FROM MENU NUMBER 10, THEN THE CORRECT
** ACTION SHOULD BE TAKEN.
**
** BY: ASHIT R. GANDHI
** DATE: 11/21/88
**
** PARAMETERS USED:
** NMUM = MENU NUMBER (INTEGER, I/P)
** IPKID = PICK ID OF THE ITEM THAT HAS PICKED (INTEGER, I/P)
** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU (CHARACTER, I/P)

SUBROUTINE FMPK10(NMUM,IPKID,PMENU)

INTEGE NNMUM, IPKID
REAL*4 PMENU(50), LOC(3), ORI(3), LEN
CHARACTER STRG*50, PRE*20, FTRG*8, ANS*1, BOOL*1

GOTO THE CORRECT ID THAT HAS PICKED
1 GOTO(11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26)IPKID

IF IPKID .GT. 8 RETURN
RETURN

NMUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN

NMUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN

NMUM = 1, IPKID = 3 ITEM PICKED = RETURN
13 NMUM = 3
  CALL FMENU2(NMUM,PMENU)
  RETURN

NMUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN

NMUM = 1, IPKID = 5 ITEM PICKED = SLAB
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
15 CALL MSLAB(A,B,C,LOC,ORI,ICOL,BOOL)
  RETURN

NMUM = 1, IPKID = 6 ITEM PICKED = SPHERE
16 CALL MSHERE(RAD,LOC,ICOL,BOOL)
  RETURN

NMUM = 1, IPKID = 7 ITEM PICKED = CYLINDER
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
17 CALL MCYNDR(RAD,LEN,LOC,ORI,ICOL,BOOL)
  RETURN

Appendix B - FeatureMod Program Listing  209
SUBROUTINE FMPK2

***********************************************************************
SUBROUTINE FMPK2(MNUM,IPKID,PMENU)

PROGRAM DESCRIPTION

IF AN ITEM WAS PICKED FROM MENU NUMBER 2, THEN THE CORRECT
ACTION SHOULD BE TAKEN.

BY: ASHIT R. GANDHI

DATE: 11/05/88

PARAMETERS USED:

MNUM = MENU NUMBER (INTEGER, I/P)
IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P)
PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU

(CHARACTER, I/P, O/P)

***********************************************************************
SUBROUTINE FMKP2(MNUM, IPKID, PMENU)
INTEGER MNUM, IPKID
REAL*4 PMENU(50)
CHARACTER STRG*50, PRE*20, FTRG*8, AN$*1
LOGICAL*4 FLCHK
*GOTO THE CORRECT ID THAT WAS PICKED
1 GOTO(11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26)IPKID
*IF IPKID .GT. 15 RETURN
RETURN
*MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN
*MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN
*MNUM = 1, IPKID = 3 ITEM PICKED = EXIT
*MAKE SURE ALL WORK HAS BEEN STORED
*IF NOT ASK WHETHER IT IS OK TO EXIT WITHOUT STORING
13 INQUIRE(UNIT=10, NAME=FTRG)
IF(FTRG .NE. ' 'THEN
   STRG='MODEL ' // FTRG // ' CURRENTLY OPEN'
   CALL FMSSG(30, STRG)
   STRG='CURRENT MODEL WILL BE LOST IF YOU EXIT'
   CALL FMSSG(30, STRG)
   CALL FPRMP(10)
   CALL FGTSTIANS(ANS)
   IF(ANS .NE. 'Y') RETURN
   CALL FCLSE
   ELSE CALL FPRMP(10)
   CALL FGTSTIANS(ANS)
   IF(ANS .NE. 'Y') RETURN
   CALL FCLSE
ENDIF
   STRG = 'EXIT NOT IMPLEMENTED'
   CALL FMSSG(30, STRG)
RETURN
*MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN
*MNUM = 1, IPKID = 5 ITEM PICKED = MODELING
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
15 MNUM = 3
   CALL FMENU2(MNUM, PMENU)
RETURN
*MNUM = 1, IPKID = 6 ITEM PICKED = LANGUAGE INPUT
16 MNUM = 6
   STRG = 'NATURAL LANGUAGE INPUT NOT IMPLEMENTED'
   CALL FMSSG(30, STRG)
   RETURN
*MNUM = 1, IPKID = 7 IS NOT A POSSIBILITY
17 RETURN
*MNUM = 1, IPKID = 8 ITEM PICKED = RENDERING
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
18 MNUM = 7
   CALL FMENU2(MNUM, PMENU)
   RETURN
*MNUM = 1, IPKID = 9 ITEM PICKED = VIEWS
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
19 MNUM = 5
   CALL FMENU2(MNUM, PMENU)
   RETURN
*MNUM = 1, IPKID = 10 ITEM PICKED = WINDOWS
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
20 MNUM = 6

Appendix B - FeatureMod Program Listing
CALL FMENU2(MNUM,PMENU)
RETURN
*MNUM = 1, IPKID = 11 IS NOT A POSSIBILITY
21 RETURN
*MNUM = 1, IPKID = 12 ITEM PICKED = DISPLAY
22 CALL FDISP
RETURN
*MNUM = 1, IPKID = 13 ITEM PICKED = INQUIRE
23 CALL FQUERY
RETURN
*MNUM = 1, IPKID = 14 ITEM PICKED = H. COPY
24 CALL HCOPY(1,2)
   STRG = 'HARD COPY GENERATED'
   CALL FMSSG(30,STRG)
RETURN
*MNUM = 1, IPKID = 15 IS NOT A POSSIBILITY
25 RETURN
*MNUM = 1, IPKID = 16 ITEM PICKED = FILES
   IN THIS CASE DISPLAY APPROPRIATE SCREEN
26 MNUM = 1
   CALL FMENU2(MNUM,PMENU)
RETURN
END

SUBROUTINE FMPK3

**********************************************************************************
** SUBROUTINE FMPK3(MNUM,IPKID,PMENU) **
** PROGRAM DESCRIPTION **
** IF AN ITEM HAS PICKED FROM MENU NUMBER 3, THEN THE CORRECT **
** ACTION SHOULD BE TAKEN. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 11/05/88 **
** PARAMETERS USED: **
** **
** MNUM = MENU NUMBER (INTEGER, I/P) **
** IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER,I/P) **
** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU **
** (CHARACTER,I/P,O/P) **
** **
**********************************************************************************

INTEGER*, MNUM, IPKID, CNUM
REAL*4 PMENU(50)
CHARACTER STRG*50, PRE*20, FTRG*8
COMMON/PRTOF/CNUM
*GOTO THE CORRECT ID THAT WAS PICKED
1   GOTO(11,12,13,14,15,16,17,18,19,20,21,IPKID)
*IF IPKID .GT. 10 RETURN
RETURN
*MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN
*MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
21 RETURN
SUBROUTINE FMPK4

*** SUBROUTINE FMPK4(MNH, IPKID, PMENU) ***
*** PROGRAM DESCRIPTION ***
*** IF AN ITEM WAS PICKED FROM MENU NUMBER 4, THEN THE CORRECT ***
*** ACTION SHOULD BE TAKEN. ***
SUBROUTINE FMPK4(MNUM,IPKID,PMENU)
    INTEGER*4 MNUM, IPKID
    REAL*4 PMENU(50)
    CHARACTER STRG*50, PRE*20, FTRG*8
    COMMON/VIS/NVIS
    GOTO THE CORRECT ID THAT WAS PICKED
1   GOTO(11,12,13,14,15,16,17,18,19,20,21,22)IPKID
1   IF IPKID .GT. 12 RETURN
      RETURN
1   MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11  RETURN
1   MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12  RETURN
1   MNUM = 1, IPKID = 3 ITEM PICKED = RETURN
13  MNUM = 2
14  CALL FMENJ2(MNUM,PMENU)
      RETURN
1   MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14  RETURN
1   MNUM = 1, IPKID = 5 ITEM PICKED = COLOR
15  MNUM = 9
16  CALL FMENJ2(MNUM,PMENU)
      RETURN
1   MNUM = 1, IPKID = 6 IS NOT A POSSIBILITY
16  RETURN
1   MNUM = 1, IPKID = 7 ITEM PICKED = HIDDEN LINES
17  CALL FSHDE(-1)
      RETURN
1   MNUM = 1, IPKID = 8 ITEM PICKED = HIDDEN SURFACE
18  CALL FSHDE(-1)
      RETURN
1   MNUM = 1, IPKID = 9 IS NOT A POSSIBILITY
19  RETURN
1   MNUM = 1, IPKID = 10 ITEM PICKED = SHADING
19  CALL FMENJ2(MNUM,PMENU)
      RETURN
1   MNUM = 1, IPKID = 11 IS NOT A POSSIBILITY
20  RETURN
1   MNUM = 1, IPKID = 12 ITEM PICKED = RESET
21  RETURN
1   CALL GPDRV(1,7,21)
21  CALL GPARV(1,7,NVIS,0)
      RETURN
SUBROUTINE FMPK5

SUBROUTINE FMPK5(MNUM, IPKID, PMENU)

PROGRAM DESCRIPTION

IF AN ITEM WAS PICKED FROM MENU NUMBER 5, THEN THE CORRECT ACTION SHOULD BE TAKEN.

BY: ASHIT R. GANDHI

DATE: 11/05/88

PARAMETERS USED:

MNUM = MENU NUMBER (INTEGER, I/P)
IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P)
PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU (CHARACTER, I/P, O/P)

GOT0 THE CORRECT ID THAT WAS PICKED
1 GOTO(11,12,13,14,15,16,17,18,19)IPKID

IF IPKID .GT. 9 RETURN

RETURN

*MNJM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN

*MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN

*MNUM = 1, IPKID = 3 ITEM PICKED = RETURN
IN THIS CASE DISPLAY APPROPRIATE SCREEN
13 MNUM = 2
CALL FMENU2(MNUM, PMENU)
RETURN

*MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN

*MNUM = 1, IPKID = 5 ITEM PICKED = MOVE
15 STRG = 'MOVE NOT IMPLEMENTED'
CALL FMSSG(30, STRG)
RETURN

*MNUM = 1, IPKID = 6 ITEM PICKED = ROTATE
16 STRG = 'ROTATE NOT IMPLEMENTED'
CALL FMSSG(30, STRG)
RETURN

*MNUM = 1, IPKID = 7 ITEM PICKED = ZOOM
17 STRG = 'ZOOM NOT IMPLEMENTED'
CALL FMSSG(30, STRG)
RETURN

*MNUM = 1, IPKID = 8 IS NOT A POSSIBILITY
18 RETURN

*MNUM = 1, IPKID = 9 ITEM PICKED = RESET

Appendix B - FeatureMod Program Listing
SUBROUTINE FMPK6

*** SUBROUTINE FMPK6(MNUM,IPKID,PMENU) ***
*** PROGRAM DESCRIPTION ***
*** IF AN ITEM WAS PICKED FROM MENU NUMBER 6, THEN THE CORRECT ***
*** ACTION SHOULD BE TAKEN. ***
*** ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/05/88 ***
*** ***
*** PARAMETERS USED: ***
*** MNUM = MENU NUMBER (INTEGER, I/P) ***
*** IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P) ***
*** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU ***
*** (CHARACTER, I/P, O/P) ***
*** ***
*** SUBROUTINE FMPK6(MNUM,IPKID,PMENU) ***
INTEGER=MNUM, IPKID
REAL=PMENU(50)
CHARACTER STRG=50, PRE=20, FTRG=8
*GOTO THE CORRECT ID THAT WAS PICKED
1  GOTO(11,12,13,14,15,16,17,18)IPKID
*IF IPKID .GT. 8 RETURN
RETURN
*MMNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN
*MMNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN
*MMNUM = 1, IPKID = 3 ITEM PICKED = RETURN
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
13  MNUM = 2
CALL FMENU2(MNUM,PMENU)
RETURN
*MMNUM = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN
*MMNUM = 1, IPKID = 5 ITEM PICKED = SINGLE
15  CALL GPVCH(1,9,2,1,1,2,11,2,1,1)
 CALL GPVCH(1,9,2,1,1,2,11,2,1,1)
 CALL GPVCH(1,9,2,1,1,2,11,2,1,1)
 CALL GPVCH(1,9,2,1,1,2,11,2,1,1)
 CALL GPVCH(1,9,2,1,1,2,11,2,1,1)
 MNUM = 8
CALL FMENU2(MNUM,PMENU)
RETURN
*MMNUM = 1, IPKID = 6 ITEM PICKED = MULTIPLE
*DISPLAY APPROPRIATE SCREEN
16  CALL GPVCH(1,7,5,1,1,2,11,2,1,1)
 CALL GPVCH(1,7,5,1,1,2,11,2,1,1)
 CALL GPVCH(1,7,5,1,1,2,11,2,1,1)
 CALL GPVCH(1,7,5,1,1,2,11,2,1,1)
 CALL FMULT
RETURN

*MNUN = 1, IPKID = 7 IS NOT A POSSIBILITY
17 RETURN

*MNUN = 1, IPKID = 8 ITEM PICKED = RESET
18 STRG = 'RESET NOT IMPLEMENTED'
   CALL FMSSG(30, STRG)
RETURN
END

SUBROUTINE FMPK7

***************************************************************************
** SUBROUTINE FMPK7(MNUM, IPKID, PMENU)
***************************************************************************
**
** PROGRAM DESCRIPTION
**
** IF AN ITEM WAS PICKED FROM MENU NUMBER 7, THEN THE CORRECT ACTION SHOULD BE TAKEN.
**
** BY: ASHIT R. GANDHI
**
** DATE: 11/05/88
**
** PARAMETERS USED:
**
** MNUN = MENU NUMBER (INTEGER, I/P)
**
** IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P)
**
** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU
**
** (CHARACTER, I/P, O/P)
**
***************************************************************************

INTEGER=MNUM, IPKID
REAL=P MENU(50)
CHARACTER STRG=50, PRE=20, FTRG=8

*GOTO THE CORRECT ID THAT WAS PICKED
1 GOTO(11, 12, 13, 14, 15, 16, 17, 18, 19) IPKID
*IF IPKID .GT. 9 RETURN
RETURN

*MNUN = 1, IPKID = 1 IS NOT A POSSIBILITY
11 RETURN

*MNUN = 1, IPKID = 2 IS NOT A POSSIBILITY
12 RETURN

*MNUN = 1, IPKID = 3 ITEM PICKED = RETURN
*IN THIS CASE DISPLAY APPROPRIATE SCREEN
13 MNUM = 6
   CALL P MENU(7, PMENU)
RETURN

*MNUN = 1, IPKID = 4 IS NOT A POSSIBILITY
14 RETURN

*MNUN = 1, IPKID = 5 ITEM PICKED = CONSTANT
15 CALL FSHDE(1)
RETURN

*MNUN = 1, IPKID = 6 ITEM PICKED = PHONG
16 STRG = 'PHONG SHADING NOT IMPLEMENTED'
   CALL FMSSG(30, STRG)
RETURN

*MNUN = 1, IPKID = 7 ITEM PICKED = GOURAUD
**SUBROUTINE FMPK8**

SUBROUTINE FMPK8(MNUM, IPKID, PMENJ)

PROGRAM DESCRIPTION

IF AN ITEM HAS PICKED FROM MENU NUMBER 8, THEN THE CORRECT ACTION SHOULD BE TAKEN.

BY: ASHIT R. GANDHI

DATE: 11/05/88

PARAMETERS USED:

MNUM = MENU NUMBER (INTEGER, I/P)

= 1

IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P)

PMENJ = ARRAY OF ITEMS AVAILABLE FOR THIS MENU

(CHARACTER, I/P, O/P)


GOTO THE CORRECT ID THAT WAS PICKED

IF IPKID .GT. 11 RETURN

RETURN

MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY

RETURN

MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY

RETURN

MNUM = 1, IPKID = 3 ITEM PICKED = RETURN

IN THIS CASE DISPLAY APPROPRIATE SCREEN

MNUM = 6

CALL FMENU2(MNUM, PMENJ)

RETURN

MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY

RETURN

MNUM = 1, IPKID = 5 ITEM PICKED = FRONT

CALL FFNTW(7)

RETURN

MNUM = 1, IPKID = 6 ITEM PICKED = TOP

CALL FTP(7)

RETURN
*MNUM = 1, IPKID = 7 ITEM PICKED = RIGHT SIDE
17    CALL FRTV(7)
    RETURN

*MNUM = 1, IPKID = 8 ITEM PICKED = ISOMETRIC
18    CALL FISOV(7)
    RETURN

*MNUM = 1, IPKID = 9 ITEM PICKED = BACK
19    CALL FBCKV(7)
    STRG = 'BACK VIEW NOT IMPLEMENTED'
    CALL FMSSG(30,STRG)
    RETURN

*MNUM = 1, IPKID = 10 ITEM PICKED = BOTTOM
20    CALL FBOTV(7)
    RETURN

*MNUM = 1, IPKID = 11 ITEM PICKED = LEFT SIDE
21    CALL FLFTV(7)
    RETURN
END

:: SUBROUTINE FMPK9 ::
:: PROGRAM DESCRIPTION ::
** IF AN ITEM HAS PICKED FROM MENU NUMBER 7, THEN THE CORRECT **
** ACTION SHOULD BE TAKEN. **
** BY: ASHIT R. GANDHI **
** DATE: 11/05/88 **
:: PARAMETERS USED: ::
** MNUM = MENU NUMBER (INTEGER, I/P) **
** = 1 **
** IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER, I/P) **
** PMENU = ARRAY OF ITEMS AVAILABLE FOR THIS MENU **
** (CHARACTER,I/P,O/P) **
::

SUBROUTINE FMPK9(MNUM, IPKID, PMENU)

INTEGER*4 MNUM, IPKID
REAL*4 PMENU(50), COL(3), SCOL(3)
CHARACTER STRG*50, PRE*20, FTRG*8
COMON/SHADE/SCOL

DATA ICST/2/
DATA NCOL/1/

GOTO THE CORRECT ID THAT WAS PICKED
1   GOTO(11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
& 28,29,30)IPKID

*IF IPKID .GT. 9 RETURN
    RETURN
*MNUM = 1, IPKID = 1 IS NOT A POSSIBILITY
11   RETURN
*MNUM = 1, IPKID = 2 IS NOT A POSSIBILITY
12   RETURN
*MNUM = 1, IPKID = 3 ITEM PICKED = RETURN
*IN THIS CASE DISPLAY APPROPRIATE SCREEN

Appendix B - FeatureMod Program Listing

13 INHM = 6
CALL PHMENU(MNUM,MNUM)
RETURN

14 **MNUM = 1, IPKID = 4 IS NOT A POSSIBILITY**
RETURN

15 **MNUM = 1, IPKID = 5 ITEM PICKED = MAXIMUM RED**
   COL(1) = 1.0
   COL(2) = 0.0
   COL(3) = 0.0
   GOTO 35

16 **MNUM = 1, IPKID = 6 ITEM PICKED = MAXIMUM GREEN**
   COL(1) = 0.0
   COL(2) = 0.0
   COL(3) = 1.0
   GOTO 35

17 **MNUM = 1, IPKID = 7 ITEM PICKED = MAXIMUM BLUE**
   COL(1) = 0.0
   COL(2) = 0.0
   COL(3) = 1.0
   GOTO 35

18 **MNUM = 1, IPKID = 8 ITEM PICKED = MAXIMUM YELLOW**
   COL(1) = 1.0
   COL(2) = 1.0
   COL(3) = 0.0
   GOTO 35

19 **MNUM = 1, IPKID = 9 ITEM PICKED = MAGENTA**
   COL(1) = 1.0
   COL(2) = 0.0
   COL(3) = 1.0
   GOTO 35

20 **MNUM = 1, IPKID = 10 ITEM PICKED = TURQUOISE**
   COL(1) = 0.0
   COL(2) = 1.0
   COL(3) = 1.0
   GOTO 35

21 **MNUM = 1, IPKID = 11 ITEM PICKED = DARK MAGENTA**
   COL(1) = 0.3
   COL(2) = 0.0
   COL(3) = 0.3
   GOTO 35

22 **MNUM = 1, IPKID = 12 ITEM PICKED = DARK GREEN**
   COL(1) = 0.0
   COL(2) = 0.1
   COL(3) = 0.0
   GOTO 35

23 **MNUM = 1, IPKID = 13 ITEM PICKED = DARK BLUE**
   COL(1) = 0.0
   COL(2) = 0.0
   COL(3) = 0.5
   GOTO 35

24 **MNUM = 1, IPKID = 14 ITEM PICKED = BACKGROUND GREY**
   COL(1) = 0.25
   COL(2) = 0.25
   COL(3) = 0.25
   GOTO 35

25 **MNUM = 1, IPKID = 15 ITEM PICKED = LIGHT GREY**
   COL(1) = 0.4
   COL(2) = 0.2
   COL(3) = 0.5
   GOTO 35

26 **MNUM = 1, IPKID = 16 ITEM PICKED = PEACOCK GREEN**
   COL(1) = 0.0
   COL(2) = 0.0
   COL(3) = 0.6
   GOTO 35

**MNUM = 1, IPKID = 17 ITEM PICKED = ORANGE**
SUBROUTINE FMSSG

*** SUBROUTINE FMSSG (NUM, MSG) ***

REAL POS(2)
CHARACTER MSGG(30)*72, TXT(9)*72, MSG(*)

*DATA FOR MESSAGE LENGTH
DATA LNT/50/, DATA TXT/9*/

*DATA FOR MESSAGES TO BE DISPLAYED
DATA MSGGI/"WELCOME TO "FEATURE"
DATA MSGG3/"A FEATURE BASED GEOMETRIC MODELING SYSTEM"
DATA MSGG4/"VERSION 1.0"
DATA MSGG5/"VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY"
DATA MSGG6/"DEVELOPED AT"
DATA MSGG7/"STARTING FILE"
DATA MSGG8/"DEFINING OBJECT"
DATA MSGG9/"ONLY DEFINE AND ADD TO ACTIVE AT PRESENT"
DATA MSGG10/"EDIT, MODIFY AND ERASE NOT ACTIVE"
DATA MSGG11/"OBJECT DEFINED"
DATA MSGG12/"ONLY DEFINE AND ADD TO OBJECT"
DATA MSGG13/"OBJECT ADDED TO"
DATA MSGG14/"OBJECT ADDED TO"
DATA MSGG15/"FEATURE LIST 1"
DATA MSGG16/"EXITING FROM THE FEATURE MENU 1"
DATA MSGG17/"FEATURE BOX"
DATA MSGG18/"ONLY FEATURE BOX IS ACTIVE"
DATA MSGG19/"FEATURE LIST 2"
DATA MSGG20/"EXITING FROM THE FEATURE MENU 2"
DATA MSGG21/"FEATURE LIST 2 NOT SUPPORTED"
DATA MSGG22/"EXITING BOX"
DATA MSGG23/"DIMENSIONING OBJECT"

Appendix B - FeatureMod Program Listing
DATA MSGG(24)/'MOVING OBJECT'/
DATA MSGG(25)/'MOVING OBJECT NOT SUPPORTED'/
DATA MSGG(26)/'RETURN'/

*DEFINE TEXT LOCATION
  POS(1)=0.02
  POS(2)=0.85

*SCROLL ALL TEXT
  MSGG(30) = MSG
  TXT(1)=TXT(2)
  TXT(2)=TXT(3)
  TXT(3)=TXT(4)
  TXT(4)=TXT(5)
  TXT(5)=TXT(6)
  TXT(6)=TXT(7)
  TXT(7)=TXT(8)
  TXT(8)=TXT(9)
  TXT(9)=MSGG(NUM)

*OPEN STRUCTURE FOR MESSAGE ITEMS
  CALL GPEST(5)

*OPEN STRUCTURE 5
  CALL GPOPST(5)

*SET TEXT COLOR TO YELLOW
  CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
  CALL GPAHSC(0.50)

*DRAW ALL TEXT
  DO 100 I=1,8
     CALL GPANZ(POS,LNT,TXT(I))
     POS(2)=POS(2)-0.1
  100 CONTINUE

*SET TEXT COLOR TO GREEN
  CALL GPTXCI(3)

*DRAW TEXT
  CALL GPANZ(POS,LNT,TXT(9))

*CLOSE STRUCTURE
  CALL GPCLST

*ASSOCIATE THE STRUCTURE 5 WITH A VIEW 5
  CALL GPARV(1,5,5,0)

*SET VIEW PRIORITY
  CALL GPVP(1,5,2,1)

*UPDATE WORKSTATION
  CALL GPUPW(1,2)
RETURN
END

SUBROUTINE FMT11

******************************************************************************
******************************************************************************
** SUBROUTINE FMT11(A,B,C) **
** PROGRAM DESCRIPTION **
** THIS ROUTINE MULTIPLIES A 1X4 MATRIX WITH A 4X1 MATRIX TO GIVE **
** A SINGLE SCALAR ANSWER. **
** BY: ASHIT R. GANDHI **
** DATE: 10/08/88 **
** PARAMETERS USED: **
** A = 1X4 MATRIX {REAL,I/P} **
** B = 4X1 MATRIX {REAL,I/P} **
******************************************************************************

Appendix B - FeatureMod Program Listing 222
**SUBROUTINE FMT14**

**SUBROUTINE FMT44**
SUBROUTINE FMULT

*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE WILL ALLOW THE DISPLAY OF MULTIPLE VIEWS OF ANY OBJECT. ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/07/88 ***
*** PARAMETERS USED: ***
*** NONE ***

*APPLY VIEW TRANSFORMATION TO EACH VIEW*
CALL FNTV(8)
CALL FTOPV(9)
CALL FRTV(10)
CALL FIS0V(11)
RETURN
END

SUBROUTINE FNORM(A,B,C)

*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE WILL NORMALIZE A GIVEN VECTOR TO A VECTOR WITH PREDEFINED MAGNITUDE. ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 10/08/88 ***
*** PARAMETERS USED: ***
*** A = X-ELEMENT OF THE VECTOR (REAL,INPUT,OUTPUT) ***
*** B = Y-ELEMENT OF THE VECTOR (REAL,INPUT,OUTPUT) ***
*** C = Z-ELEMENT OF THE VECTOR (REAL,INPUT,OUTPUT) ***

SUBROUTINE FNORM(A,B,C)
REAL A,B,C,RMAG
RMAG = SQRT(A**2 + B**2 + C**2)
IF (RMAG .EQ. 0.0)THEN
RETURN
ENDIF
A = A/RMAG
B = B/RMAG
C = C/RMAG

Appendix B · FeatureMod Program Listing
**SUBROUTINE FOPEN**

* SUBROUTINE DESCRIPTION *
** THIS SUBROUTINE ALLOWS THE USER TO INITIALIZE PHIGS. IT OPENS**
** PHIGS, WORKSTATION AND LOADS THE COLOR TABLE.**
** BY: ASHIT R. GANDHI**
** DATE: 10/07/88**
**
* PARAMETERS USED: *
** ERFILE : NAME OF ERROR FILE **
** WSTYPE : WORKSTATION TYPE **
** CONNID : WORKSTATION CONNECTION ID **
** ICST : START INDEX INTO COLOR TABLE **
** NCOL : TOTAL NUMBER OF COLORS TO BE LOADED **
** COLORS : ARRAY OF R,G,B VALUES FOR COLORS LOADED **

* COMMON CSIZE *

REAL*4 COLORS(48), CSIZE(3)
INTEGER*4 ICST, NCOL, ASIZE(3)
CHARACTER*8 ERFILE, WSTYPE, CONNID

* DATA COLORS/0., 0., 0.,
1. 1., 1., 1.,
2. 1., 0., 0.,
3. 0., 1., 0.,
4. 0., 0., 1.,
5. 1., 1., 0.,
6. 1., 0., 1.,
7. 0., 1., 1.,
8. .3, 0., .3,
9. 0., .2, 0.,
0. 0., 0., .5,
1. 0., 0., 0.,
2. .4, .5, .5,
3. 0., 1., .6,
4. 1., .3, .3,
5. 0., 1., .6/

* DATA ICST/0/
* DATA NCOL/16/
* DATA ERFILE/'SY5PRINT'/
* DATA WSTYPE/'5080 '/
* DATA CONNID/'IBM5080 '/

* OPEN PHIGS *
CALL GPOPPH(ERFILE,0)
* OPEN WORKSTATION *
CALL GPOPWS(1,CONNID,WSTYPE)
* LOAD COLOR TABLE *
CALL GPCR(1,ICST,NCOL, COLORS)
* INQUIRE DISPLAY SURFACE SIZE *
CALL GPQADS(1,ERRIND,UNITS,CSIZE,ASIZE)
RETURN
END

SUBROUTINE FPARS(LCHAR, NN, NORD)
This subroutine will take a line and split it into words that are a part of it. Additionally, it also gives the number of words that form that line.

By: Ashit R. Gandhi
Date: 07/15/88

Parameters Used:
- LCHAR - 72 character string line (input)
- NM - number of words forming the line (output)
- WORD - array of words that form the line (output)
- FC - first character of the word string
- LC - last character of the word string
- IM - flag to indicate state of word
  0 - end of word/blank space
  1 - continuous word

Variables Used:
- SEPA - array of separators

Subroutine FPARS(LCHAR, NM, WORD)
CHARACTER LCHAR*72, CHAR*1, NORD(15)*12, SEPA(13)*1
INTEGER FC(20), LC(20)
DATA NSEPA/13/
NN = 0
IM = 0
DO 200 I = 1, 72
CHAR = LCHAR(I:I)
DO 100 J = 1, NSEPA
IF (CHAR .EQ. SEPA(J)) THEN
LCHAR(I:I) = ' '
GOTO 200
ENDIF
100 CONTINUE
200 CONTINUE
IF (CHAR .NE. ' ' .AND. IM .EQ. 0) THEN
NN = NN + 1
FC(NM) = I
IM = 1
ENDIF
IF (CHAR .EQ. ' ' .AND. IM .EQ. 1) THEN
LC(NM) = I
IM = 0
WORD(NM) = LCHAR(FC(NM):LC(NM))
ENDIF
300 CONTINUE
RETURN
END

SUBROUTINE FPGAT

This subroutine sets up polygon attributes.

By: Ashit R. Gandhi
SUBROUTINE FPRMP

*PROGRAM DESCRIPTION*
THIS SUBROUTINE PROMPTS MESSAGES IN THE PRE-DEFINED AREA.

*BY: ASHIT R. GANDHI*
DATE: 10/08/88

*PARAMETERS USED:*
NUM = MESSAGE NUMBER (INTEGER, I/P)

*SUBROUTINE FPRMP NUM*

INTEGER*4 LNT, NUM
REAL*4 POS(2)
*LENGTH OF PROMPT MESSAGES*
DATA LNT/50/
*PROMPT MESSAGES*
DATA PROMPT(1)/'SELECT START OR RECALL TO BEGIN '/
DATA PROMPT(2)/'SELECT OPTION '/
DATA PROMPT(3)/'ENTER NAME OF THE FILE TO START '/
DATA PROMPT(4)/'ENTER NAME OF THE FILE TO STORE '/
DATA PROMPT(5)/'ENTER NAME OF THE COMMAND FILE TO BE EXECUTED '/
DATA PROMPT(6)/'DO YOU WANT TO OPEN A NEW FILE? (Y/N) '/
DATA PROMPT(7)/'DO YOU WANT TO RECALL A MODEL FILE? (Y/N) '/
DATA PROMPT(8)/'SAVE UNDER SAME NAME? (Y/N) '/
DATA PROMPT(9)/'DO YOU STILL WANT TO EXIT? (Y/N) '/
DATA PROMPT(10)/'ARE YOU SURE? (Y/N) '/
DATA PROMPT(11)/'INVALID OPTION SELECTED '/
DATA PROMPT(12)/'MODEL HAS NOT BEEN SAVED. OK TO EXIT? (Y/N) '/
DATA PROMPT(13)/'SELECT PROPER PARAMETERS AND THEN EXECUTE '/
DATA PROMPT(14)/'ENTER NAME OF OBJECT TO DEFINE '/
DATA PROMPT(15)/'ENTER NAME OF COMPONENT TO BE DELETED '/
DATA PROMPT(16)/'ENTER NAME OF COMPONENT TO ADD TO '/
DATA PROMPT(17)/'ENTER NAME OF COMPONENT TO DISPLAY '/

*DEFINE TEXT LOCATION*
POS(1) = .03
POS(2) = .52

*OPEN STRUCTURE FOR PROMPT ITEMS*
CALL GPEST(4)

*OPEN STRUCTURE 4*
CALL GPOPST(4)

*SET TEXT COLOR TO TURQUOISE*
CALL GPTXI(7)
SUBROUTINE FPRPK

*** SUBROUTINE FPRPK(MNUM,IPKID,PMENU) ***
*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE CALLS THE APPROPRIATE MENU ROUTINE TO PROCESS ***
*** THE ITEM THAT WAS PICKED. ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/04/88 ***
*** PARAMETERS USED: ***
** MNUM = MENU NUMBER THAT WAS LAST DISPLAYED (INTEGER,1/P) **
** IPKID = PICK ID OF THE ITEM THAT WAS PICKED (INTEGER,1/P) **
** PMENU = ARRAY OF ITEMS PRESENT ON THE LAST MENU DISPLAYED **
** (CHARACTER,1/P) **
*** SUBROUTINE FPRPK(MNUM,IPKID,PMENU) ***
INTEGER MNUM, IPKID
CHARACTER STRG=50, PMENU(50)*15
*GO TO CORRECT MENU SELECTION AREA
GOTO(1,2,3,4,5,6,7,8,9,10)*MNUM
*IF MNUM .GT. 10 RETURN
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 1
1 CALL FMPK1(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 2
2 CALL FMPK2(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 3
3 CALL FMPK3(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 4
4 CALL FMPK4(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 5
CALL FMMPK5(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 6
CALL FMMPK6(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 7
CALL FMMPK7(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 8
CALL FMMPK8(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 9
CALL FMMPK9(MNUM,IPKID,PMENU)
RETURN
*IF LAST MENU DISPLAYED IN NUMBER 10
CALL FMMPK10(MNUM,IPKID,PMENU)
RETURN
END

********************************************************************
SUBROUTINE FQUERY
********************************************************************

INTEGER ASNUM, OBNUM, INNUM
CHARACTER*8 ASSEM(900), OBJECT(9000)
COMMON/IDS/ASNUM, OBNUM, INNUM
COMMON/COMP/ASSEM, OBJECT

*COMPUTE NUMBER OF OBJECTS AND ASSEMBLIES PRESENT IN THE MODEL

NAMS = ASNUM - 100
NOBJ = OBNUM - 1000

*PRINT REQUESTED INFORMATION AND ASSEMBLY NAMES
WRITE(6,'(T4), NASM = ASNUM - 100
WRITE(6,'(T4), OBJ = OBNUM - 1000
WRITE(6,'(T4), ASM = ASNUM - 100
WRITE(6,'(T4), OBT = OBNUM - 1000

*PRINT REQUESTED INFORMATION AND OBJECT NAMES
WRITE(6,'(T4), NASM = ASNUM - 100
WRITE(6,'(T4), NOBJ = OBNUM - 1000
WRITE(6,'(T4), ASM = ASNUM - 100
WRITE(6,'(T4), OBT = OBNUM - 1000

Appendix B - FeatureMod Program Listing
***SUBROUTINE FRBMH (PMAT, HMAT, NU, NH, PT)***

**PROGRAM DESCRIPTION**

This subroutine uses the geometric coefficient matrix to calculate the specified number of points which would lie on the periodic cubic rational B-spline surface.

**By:** ASHIT R. GANDHI

**Date:** 11/07/88

**Parameters Used:**

- PMAT = geometric coefficient matrix (REAL,I/P)
- HMAT = height matrix (REAL,I/P)
- NU = number of points to be computed in U-direction (INTEGER,I/P)
- NH = number of points to be computed in W-direction (INTEGER,I/P)
- PT = coordinates of points on the computed surface (REAL,O/P)

*****

**SPLIT THE P-MATRIX INTO ITS RESPECTIVE ELEMENTS**

DO 200 I = 1, 4
  DO 100 J = 1, 4
    PXMAT(I,J) = PMAT(I,J,1)
    PYMAT(I,J) = PMAT(I,J,2)
    PZMAT(I,J) = PMAT(I,J,3)
  100 CONTINUE
  200 CONTINUE

**CALCULATE THE DELTA CHANGES IN U AND H**

DELU = 1.0/FLOAT(NU - 1)
DELH = 1.0/FLOAT(NH - 1)

**CALCULATE THE POINTS CORRESPONDING TO DIFFERENT VALUES OF U AND H**

CALL FMT44(M, PXMAT, PMT)
CALL FMT44(PMT, MT, PXMAT)
CALL FMT44(M, PYMAT, PMT)
CALL FMT44(PMT, MT, PYMAT)
CALL FMT44(M, PZMAT, PMT)
CALL FMT44(PMT, MT, PZMAT)
CALL FMT44(M, HMAT, PMT)
CALL FMT44(PMT, MT, HMAT1)

U = 0.0
DO 500 I = 1, NU
  M = 0.0
  CALL FOMAT(U, VU)
  500 CONTINUE

Appendix B - FeatureMod Program Listing
CALL FMT14(VU,PXMAT,FX)
CALL FMT14(VU,PYMAT,FY)
CALL FMT14(VU,PZMAT,FZ)
CALL FMT14(VU;HMAT1;FH)
DO 400 J = 1,NW
   CALL FUWVT(W,VW)
   CALL FMT-11( FX,VW,PT(I,J,1))
   CALL FMT11(FY,VW,PT(I,J,2))
   CALL FMT11(FZ,VW,PT(I,J,3))
   CALL FMT11(FH,VW,H(I,J))
N = N + DELN
DO 300 K = 1,3
   PT(I,J,K) = PT(I,J,K)/H(I,J)
300 CONTINUE
400 CONTINUE
U = U + DELU
500 CONTINUE
RETURN
END

SUBROUTINE FRBMT
*****************************************************************************
*** SUBROUTINE FRBMT(IPGN, IU, IM, MPTS, NPTS, PNTS, H0, ICOL) ***
*** PROGRAM DESCRIPTION ***
*** THIS SUBROUTINE WILL ASSEMBLE THE P-MATRIX TO BE USED IN THE ***
*** CREATION OF A RATIONAL B-SPLINE (PERIODIC, CUBIC) SURFACE. ***
*** AFTER ASSEMBLING THE MATRIX IT WILL COMPUTE THE POINTS ON THE ***
*** SURFACE. ***
*** ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/03/88 ***
*** PARAMETERS USED: ***
*** IU = FLAG INDICATING WHETHER SURFACE IS CLOSED AT U = 0 ***
*** (INTEGER; I/P) ***
*** OPEN ***
*** CLOSED ***
*** IM = FLAG INDICATING WHETHER SURFACE IS CLOSED AT W = 0 ***
*** (INTEGER; I/P) ***
*** OPEN ***
*** CLOSED ***
*** MPTS = NUMBER OF POINTS IN THE M-DIRECTION (INTEGER; I/P) ***
*** NPTS = NUMBER OF POINTS IN THE N-DIRECTION (INTEGER; I/P) ***
*** PNTS = ARRAY OF POINTS DEFINING THE SURFACE (REAL; I/P) ***
*** H0 = ARRAY OF WEIGHTS ASSIGNED TO EACH POINT (REAL; O/P) ***
*** ICOL = COLOR TO BE ASSIGNED TO INSTANCE ***
*****************************************************************************
SUBROUTINE FRBMT(IPGN, IU, IM, MPTS, NPTS, PNTS, H0, ICOL)
INTEGER Asnum, Obnum, Innum
REAL PMAT(4,4), Pts(0:19,0:19,3), Pt(20,20,3), Pnts(20,20,3)
REAL H0(20,20), Hmat(4,4), H1(0:19,0:19), PGN(3,20000,5)
REAL Ploc(3), Pori(3)
COMMON/PATCH/Nu, Nu
COMMON/IDS/Asnum, Obnum, Innum
COMMON/Rend/Ploc, Pori
IF (IPGN .EQ. 0) THEN
   Iopgn = 1
ELSE
   Iopgn = Iopgn + 1
ENDIF
*READ IN VALUES FOR THE RATIONAL CONTROL POINTS
DO 300 I = 0, MPTS-1
   DO 200 J = 0, NPTS-1
      H1(I,J) = H0(I+1,J+1)
   DO 100 K = 1,3
      Pts(I,J,K) = H0(I+1,J+1)*Pnts(I+1,J+1,K)
   100 CONTINUE
200 CONTINUE
300 CONTINUE
*CHECK TO SEE WHETHER THE SURFACE IS CLOSED OR OPEN
*IF SURFACE IS OPEN AT U = 0.0 AND OPEN AT W = 0.0
   IF (IU .EQ. 0 .AND. IM .EQ. 0) THEN
      Send = MPTS - 3
Appendix B - FeatureMod Program Listing 231
TEND = NPTS - 3
ENDIF

*IF SURFACE IS OPEN AT U = 0.0 AND CLOSED AT M = 0.0
IF (IU .EQ. 0 .AND. IM .EQ. 1) THEN
  SEND = MPTS - 3
  TEND = NPTS
ENDIF

*IF SURFACE IS CLOSED AT U = 0.0 AND OPEN AT M = 0.0
IF (IU .EQ. 1 .AND. IM .EQ. 0) THEN
  SEND = MPTS
  TEND = NPTS - 3
ENDIF

*IF SURFACE IS CLOSED AT U = 0.0 AND CLOSED AT M = 0.0
IF (IU .EQ. 1 .AND. IM .EQ. 1) THEN
  SEND = MPTS
  TEND = NPTS
ENDIF

*COMPUTE ALL ELEMENTS ON THE SURFACE
*COMPUTE ELEMENTS IN THE S-DIRECTION
  DO 800 I = 1,SEND
*COMPUTE ELEMENTS IN THE T-DIRECTION
    DO 700 J = 1,TEND
      SET UP THE H-MATRIX AND P-MATRIX FOR EACH ELEMENT
        DO 600 II = 1,4
          DO 500 JJ = 1,4
            IX = MOD((1+II-2),MPTS)
            IY = MOD((J+JJ-2),NPTS)
            HMAT(II,JJ) = H1(IX,IY)
            PMAT(II,JJ,LL) = PTS(IX,IY,LL)
          CONTINUE
        CONTINUE

      *COMPUTE ALL POINTS ON EACH SURFACE ELEMENT
        IF (NU .EQ. 0) NU = 8
        IF (NH .EQ. 0) NH = 8
        CALL FRBMH(PMAT,HMAT,NJ,IN,PT)
      *DRAW THE CONSTANT U AND M LINES
        CALL FDRBM(HMAT,PT)
        CALL FDRPM(PMAT,PT)
      *STORE POLYGON DATA
        CALL FSTOR(IPGN,IPN,IN,IPM,IPN,PT)
      CONTINUE
  CONTINUE

SET UP RENDERING PARAMETERS
  CALL FCPRM(IPGN,IPN,IPM,ICOL,IOPGN)
RETURN
11 FORMAT(1X,8(F6.2,1X))
END

SUBROUTINE FRCLL

******************************************************************************
** SUBROUTINE FRCLL(I10,I21,I22,I31,I32)                                **
** PROGRAM DESCRIPTION                                                   **
** THIS ROUTINE WILL RECALL A EXISTING MODEL.                           **
** BY: ASHIT R. GANDHI                                                 **
** DATE: 01/13/69                                                        **
** PARAMETERS USED:                                                     **
** I10 : FLAG FOR EXISTENCE OF UNIT = 10                                **
******************************************************************************

Appendix B - FeatureMod Program Listing 232
** I21 : FLAG FOR EXISTENCE OF UNIT = 21
** I22 : FLAG FOR EXISTENCE OF UNIT = 22
** I31 : FLAG FOR EXISTENCE OF UNIT = 31
** I32 : FLAG FOR EXISTENCE OF UNIT = 32

*INITIALIZE POLYGON NUMBER AND SCALE PARAMETERS
IPGNR = 0
IPGN = IPGNR
SCALEF(1) = 1.0
SCALEF(2) = 1.0
SCALEF(3) = 1.0
WRITE(6,*)'ENTER FRCLL'
CALL TIMEON

*IF UNIT 31 DOES NOT EXIST SKIP THE NEXT PART
IF (I31 .EQ. 0) THEN
  NASM = 100
  GOTO 200
ENDIF

*READ ASSEMBLY NAME AND NUMBER INFORMATION
DO 100 I = 1,900
  READ(I31,10,END=200)NASM,ASSEM(I)
100 CONTINUE
200 BACKSPACE(I31)
ASNUM = NASM

*IF UNIT 21 DOES NOT EXIST SKIP THE NEXT PART
IF (I21 .EQ. 0) THEN
  GOTO 400
ENDIF

*READ ASSEMBLY LOCATION AND ORIENTATION INFORMATION
DO 300 I = 1,ASNUM
  READ(I21,10,END=400)NASM,ASSEM(I)
  READ(I21,40,END=400)LOC,ORI
  CALL FURTORTI MAT1)
  CALL GPTRL3(LOC MAT2)
  CALL GPCMT3(MAT1 MAT2,MAT3)
  CALL GPOPST(INASM)
  CALL GPGLX3(MAT3,3)
  CALL GPCLST
300 CONTINUE
400 BACKSPACE(I21)

*IF UNIT 32 DOES NOT EXIST SKIP THE NEXT PART
IF (I32 .EQ. 0) THEN
  NOBJ = 1000
  GOTO 600
ENDIF

*READ OBJECT NUMBER AND NAME INFORMATION
DO 500 I = 1,NOBJ
  READ(I32,20,END=600)NOBJ,OBJECT(I)
  CALL POPSFXNOBJ
  CALL GPCLST
500 CONTINUE
600 BACKSPACE(I32)
OBNUM = NOBJ

*IF UNIT 22 DOES NOT EXIST SKIP THE NEXT PART
IF (I122 .EQ. 0) THEN
  GOTO 800
ENDIF
*READ OBJECT LOCATION AND ORIENTATION WITHIN AN ASSEMBLY

DO 700 I = 1, OBNUM
    READ(22,30,END=800)LOBJ,ODUM,NASH,ADUM
    READ(22,40,END=800)LOC,ORI
    CALL FURMTI(ORI, MAT1)
    CALL GPTRLSI(LOC, MAT2)
    CALL GPCMT3(MAT1, MAT2, MAT3)
    CALL GPSTI(NASH)
    CALL GPMX3(MAT, MAT3)
    CALL GPXST(NOBJ)
    CALL GPCLST
    700 CONTINUE

800 BACKSPACE(22)
*IF UNIT 32 DOES NOT EXIST SKIP THE NEXT PART

IF (I10 .EQ. 0) THEN
    NINS = 0
    COTO 1200
ENDIF

*READ INSTANCE INFORMATION
*INFORMATION ON INSTANCE NUMBER, TYPE, PARENT OBJECT, SURFACES, COLOR,
*CONTROL POINTS AND HEIGHT INFORMATION

DO 1100 I = 1, NINST
    READ(10,50,END=1200)NINS, LTYPE, INST, NSURF, NOBJ, ODUM
    READ(10,40,END=1200)LOC, ORI
    LNUM = NINS
    CALL FURMTI(ORI, MAT1)
    CALL GPTRLSI(LOC, MAT2)
    CALL GPCMT3(MAT1, MAT2, MAT3)
    CALL GPSTI(NOBJ)
    CALL GPMX3(NINS)
    CALL GPCLST
    DO 1000 J = 1, NSURF
        READ(10,*,END=1200)ININS, MPTS, NPTS, IU, IM, ICOL, NI, NM
        DO 900 K = 1, NPTS
            READ(10,*,END=1200)PNTS(L, K, 1), L = 1, MPTS
        DO 800 K = 1, NPTS
            READ(10,*,END=1200)H0(L, K, 1), L = 1, MPTS
        DO 700 K = 1, NPTS
            READ(10,*,END=1200)H1(L, K, 1), L = 1, MPTS
        END DO
        900 CONTINUE
        CALL GPSTI(NINS)
        CALL GPPLCII(ICOL)
        CALL FURM(T(IPGN, IU, IM, MPTS, PNTS, H0, ICOL)
        CALL GPCLST
        IPGNR = IPGN
    END DO
1000 CONTINUE
WRITE(6,*)'INSTANCE NO. = ', I
1100 CONTINUE
1200 BACKSPACE(10)
RETURN
10 FORMAT(I8,1X,A8)
20 FORMAT(I8,2(I3,1X,A8))
30 FORMAT(I8,2(I3,1X,A8))
40 FORMAT(I8,6(F8.3,1X))
50 FORMAT(I8,6(F8.3,1X))
END

SUBROUTINE FRGTV

***************************************************************************
** SUBROUTINE FRGTV(NVIEW) **
** **
** PROGRAM DESCRIPTION **
** **
** THIS SUBROUTINE WILL DISPLAY THE RIGHT SIDE VIEW OF ANY OBJECT. **
** **
** BY: ASHIT R. GANDHI **
** DATE: 11/07/88 **
** **
** PARAMETERS USED: **
** **
** NST = STRUCTURE ASSOCIATED WITH THE VIEW (INTEGER, I/P) **
** NVIEW = VIEW IN WHICH THE TOP VIEW IS TO BE DISPLAYED **
** (INTEGER, I/P) **
** **
***************************************************************************

Appendix B - FeatureMod Program Listing 234
SUBROUTINE FROTA

*** PROGRAM DESCRIPTION ***

** THIS SUBROUTINE APPLIES THE CORRECT ROTATION REQUESTED ON THE **
** VIEW DISPLAYING A SINGLE WINDOW **

** BY: ASHIT R. GANDHI **
** DATE: 11/25/88 **

** PARAMETERS USED: **

** NST = NOT REQUIRED **
** X = ROTATION ABOUT THE X AXIS (REAL,I/P) **
** Y = ROTATION ABOUT THE Y AXIS (REAL,I/P) **
** Z = ROTATION ABOUT THE Z AXIS (REAL,I/P) **
** IDEV = CORRESPONDING DEVICE NUMBER (INTEGER,I/P) **

*** SUBROUTINE FROTA(NST,X,Y,Z,IDEV) ***

REAL MAT(4,4), MAT1(4,4), RX, RY, RZ, PI
REAL OLDMAT(4,4)

DATA MAT/1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0/
DATA MAT1/1.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0/
DATA RX,RY,RZ/0.0,0.0,0.0/
PI = 3.14159

*QUERY THE OLD TRANSFORMATION MATRIX
CALL GPQCVX(1,1,IER OLDMAT,AJXK1,AJXK2,AJXK3,AJXK4,
AJXK5,AJXK6,AJXK7,AJXK8,AJXK9)

*ROTATION REQUESTED IS ABOUT THE X AXIS
IF (IDEV .EQ. 1) THEN
RX = PI*X/180.0
DX = RX - OX
OX = RX
CALL GPRTX(DX,MAT1)
CALL GPCMT3(MAT1,OLDMAT,MAT)
GOTO 100
ENDIF

*ROTATION REQUESTED IS ABOUT THE Y AXIS
IF (IDEV .EQ. 2) THEN
RY = PI*Y/180.0
DY = RY - OY
OY = RY
CALL GPRTY(DY,MAT1)
CALL GPCMT3(MAT1,OLDMAT,MAT)
END
SUBROUTINE FSCAN

**********************************************************************
** SUBROUTINE FSCAN (LCHAR) **
** PROGRAM DESCRIPTION **
** THIS PROGRAM ALLOWS THE USER TO INPUT A STRING FILE THAT **
** WILL BE PROCESSED AND SENDS THE COMMAND FOR FURTHER ACTION **
** TO FEATUREMOD **
** BY: ASHIT R. GANDHI **
** DATE: 01/13/89 **
** PARAMETERS USED: **
** LCHAR : CHARACTER STRING INPUT BY THE USER **
** **
**********************************************************************

SUBROUTINE FSCAN(LCHAR)

CHARACTER LCHAR*72, WORD(15)*12, NAME*12, PREF*12
CHARACTER*4 VERB(36), TERM(119), FILL(24), POSI(12), SURF(6)
DATA NV, NT, NF, NP, NS/36, 119, 24, 12, 6/
DATA VERB/ 'ADD', 'COMP', 'COPY', 'CREA', 'DISP',
1 'DRAW', 'EDIT', 'ENTR', 'ERAS', 'END', 'FRAM', 'HELP',
1 'INCL', 'JOIN', 'LAME', 'LIST', 'MIX', 'NAME', 'PAUS',
1 'PLAC', 'PLOT', 'POSI', 'RECA', 'RENA', 'REPL', 'RUN',
1 'SAVE', 'SCL', 'SET', 'STOP', 'SUBT', 'USE', 'VIEW',
1 'ZOOM'/
DATA TERM/ 'ANGL', 'ARC ', 'ARCH ', 'AXLE ', 'BAR', 'BEAM ', 'BEAR ', 'BEND ',
1 'BULG', 'BOLT', 'BONE ', 'BOSS', 'BOX', 'BRAC', 'BRAI', 'BRAC ',
1 'CHAS', 'CHOR', 'CLEA', 'CLEV', 'CLIP', 'CLOS', 'COLL', 'COLG',
1 'CONE', 'CORN', 'CORN', 'CORN', 'CORN', 'CORN', 'CORN', 'CUBE',
1 'CUP', 'CYLI', 'DEPR', 'DEPT', 'DISK', 'DONE', 'DRIL', 'DROP',
1 'ELL', 'FACE', 'FAST', 'FILL', 'FIN ', 'FLAN', 'FURL', 'GAP',
1 'GASK', 'GRIP', 'GRIO', 'HAND ', 'HELI', 'HEMI', 'HOLE ', 'HOO',
1 'JOIN', 'JOUR', 'JUG ', 'KNEF ', 'KEY ', 'KEST ', 'KNOB ', 'KNUR',
1 'LUG ', 'MESH ', 'NECK ', 'NEED ', 'NUTC ', 'NUT ', 'PAD ', 'PEEN',
1 'PIN ', 'PIPE ', 'PLAT ', 'POCK ', 'PROT ', 'REEL ', 'RIB ', 'RING',
1 'RIVE ', 'ROUN ', 'SCR ', 'SHAF ', 'SLAB ', 'SLEE ', 'SLIT ',
1 'SLOT ', 'SPIC ', 'SPI ', 'SPIN ', 'SPRT ', 'SPRO ', 'STOR ', 'TAP ',
1 'TCON ', 'THRE ', 'TOLE ', 'TOOT ', 'TUR ', 'TRIP ', 'TUBE ', 'TUS',
1 'UND ', 'VALL ', 'WASH ', 'WEB ', 'WEDG ', 'WEED ', 'WELD ', 'WHEE'/
DATA FILL/ 'A', 'AN ', 'AND ', 'AS ', 'AT ', 'CALL ', 'FOR ', 'FROM',
1 'IN ', 'INTO ', 'IT ', 'LOC ', 'NAME ', 'OP ', 'ON ', 'DRIE',
1 'OUT ', 'PARA ', 'THE ', 'THIS ', 'TO ', 'UP ', 'UPTO ', 'WITH '/
DATA POSI/ 'BACK ', 'BOTT ', 'CENT ', 'END ', 'FRON ', 'HORI ', 'INSI ', 'LEFT',
1 'RIGH ', 'STAR ', 'TOP ', 'VERT '/
DATA SURF/ 'CONC ', 'CONV ', 'CORN ', 'EDGE ', 'FACE ', 'SMOO '/
NIN = 5
NOUT = 6

CONVERT STRING INPUT TO UPPER CASE
CALL FCAPS(LCHAR)

ERASE THE WORD ARRAY
50  DO 800 I = 1,12, NORD(I) = ' '800 CONTINUE

*PARSE THE LINE INPUT INTO A NORD ARRAY
CALL FPARS(LCHAR,NN,NORD)

*CHECK FOR "STOP"
IF (WORD(1) .EQ. 'STOP')GOTO 99

*SCAN FOR FILLER WORDS
K = 0
DO 200 I = 1,NN
   DO 100 J = 1,NF
      IF (WORD(I)(1:4) .EQ. FILL(J)) THEN
         K = K + 1
         WORD(I) = ' '
      ENDIF
   100 CONTINUE
200 CONTINUE

*REMOVE FILLER WORDS
250 DO 400 I = 1,NN-K
   IF (WORD(I) .EQ. ' ') THEN
      DO 300 J = LW
         WORD(J) = WORD(J+1)
   300 CONTINUE
   IF (WORD(I) .EQ. ' ') GOTO 250
400 CONTINUE

NA = WN - K
NNR = NN
CALL FMODEL(LCHAR,NNR,NORD)
RETURN

*SCAN FIRST WORD FOR VERB, IF NO MATCH EXISTS GIVE ERROR
IV = 0
DO 500 J = 1,NV
   IF (WORD(I)(1:4) .EQ. VERB(J)) THEN
      IV = J
      WORD(I) = VERB(J)
   ENDIF
500 CONTINUE
IF (IV .EQ. 0) THEN
   LCHAR = WORD(1)//' ' CALL FMSSG(30,LCHAR)
   LCHAR = 'ERROR: NO VERB MATCH'
   CALL FMSSG(30,LCHAR)
   LCHAR = 'RE-ENTER COMMAND STRING OR USE ANOTHER MODE OF I/P'
   CALL FMSSG(30,LCHAR)
   GOTO 99
ENDIF

NN = NN - 1

*SCAN SECOND WORD FOR GEOMETRICAL TERM
IT = 0
DO 600 J = 1,NT
   IF (WORD(2)(1:4) .EQ. TERM(J)) THEN
      IT = J
      WORD(2) = TERM(J)
   ENDIF
600 CONTINUE
IF (IT .EQ. 0) THEN
   LCHAR = 'ERROR: NO TERM MATCH'
   CALL FMSSG(30,LCHAR)
   LCHAR = WORD(1)//' ' WHAT ?'
   CALL FMSSG(30,LCHAR)
   READ(NIN,30) WORD(2)
ENDIF

*SCAN THIRD WORD FOR FEATURE NAME
625 NAME = WORD(3)
CALL FNAME(NAME,650)
GOTO 675
650 LCHAR = 'ERROR: DUPLICATE NAME'
   CALL FMSSG(30,LCHAR)
   LCHAR = 'GIVE NEW NAME FOR '//WORD(2)
   CALL FMSSG(30,LCHAR)
   READ(NIN,30) WORD(3)
GOTO 625
*SCAN FOURTH WORD FOR POSITION TERM*

675  ID = 0

   DO 700 J = 1, NP
      IF(WORD(4)(1:4) .EQ. POSI(J)) THEN
         ID = J
         WORD(4) = POSI(J)
      ENDIF

700  CONTINUE

   IF (ID .EQ. 0) THEN
      LCHAR = 'ERROR: NO POSITION MATCH'
      CALL FMSSG(30, LCHAR)
      LCHAR = NOR0(4) / '°'
      CALL FMSSG(30, LCHAR)
      LCHAR = NORD(1) / NOR0(2) / ' WHERE?'
      READ(NIN, 50) WORD(4)
   ENDIF

   IF (NIRM .LT. 4) NIRM = 4
   NRITE(NOUT) = (NOR0(I), I = 1, NIRM)

99  RETURN

10  FORMAT(A72)

20  FORMAT(15(1X, A12))

30  FORMAT(A4)

END

SUBROUTINE FNAME(NAME)

INTEGER FNUM
CHARACTER NAME*12, OBJNAME(200)*4

DATA FNUM/1/

IF(NAME(1:4) .EQ. ' ') RETURN

DO 100 I = 1, FSM-1
   IF(NAME(1:4) .EQ. OBJME(I)) RETURN

100  FNUM = FNWI + 1

OBJNME(FNUM) = NAME(1:4)

RETURN

END

SUBROUTINE FSCLE(SCALE)

PROGRAM DESCRIPTION

** THIS ROUTINE ALLOWS A USER TO ZOOM INTO AN OBJECT THAT IS **
** ASSIGNED TO VIEW 7. **

**

** BY: ASHIT R. GANDHI **

** DATE: 01/13/89 **

** PARAMETERS USED: **

** SCALE = SCALE VALUE TO BE APPLIED TO ZOOM **

**

*COMPUTE NEW SCALE FACTOR*

SCALE = SCALE / SCALEF

SUBROUTINE FSCLE(SCALE)

REAL SCALE, SCLRAT(3)
REAL OLOMAT(4, 4), MAT1(4, 4), MAT2(4, 4)
REAL AJUNK1(4), AJUNK2(6), AJUNK3(3)
REAL SCALEF(3)

COMMON /SCLE/ SCALEF

SCLRAT(1) = SCALE / SCALEF(1)
SCLRAT(2) = SCALE / SCALEF(2)
SCLRAT(3) = SCALE / SCALEF(3)

Appendix B - FeatureMod Program Listing 238
**SUBROUTINE FSCRA**

REAL*4 BOX1(8), BOX2(8), BOX3(8), BOX4(8), BOX5(8), BOX6(8), BOX7(8)
REAL*4 POS1(2), POS2(2), POS3(2), POS4(2), POS5(2)
CHARACTER TXT1*4, TXT2*6, TXT3*1, TXT4*4, TXT5*7

*BOX FOR "MENU" AREA
DATA BOX1/70., 89., 99., 99., 99., 70., 99./

*BOX FOR TITLE MONTING BOX

*BOX FOR SCROLL AREA

*BOX FOR INPUT STRING AREA
DATA BOX4/-99., -70., 69., -70., 69., -65., -99., -65./

*BOX FOR PROMPT AREA
DATA BOX5/-99., -64., 69., -64., 69., -59., -99., -59./

*MAIN SCREEN BOX

*BOX FOR MENU
DATA BOX7/70., -70., 69., 99., 99., 70., 88./

*DATA FOR TEXT AND TEXT LOCATIONS
DATA POS1/85., 94./
DATA TXT1/'MENU'/
DATA POS2/85., -84./
DATA TXT2/'V.P.I.'/
DATA POS3/85., -88./
DATA TXT3/'&'/
DATA POS4/85., -92./
DATA TXT4/'S.U.'/
DATA POS5/85., -75./
DATA TXT5/'FEATURE'/
CALL GPOPST(3)
*SET UP POLYGON ATTRIBUTES
CALL FDATI(1,2)
CALL FEDATI(2,1)

*DRAW POLYGONS REPRESENTING DIFFERENT AREAS ON THE SCREEN
CALL GPPG2(1,4,2,BOX1)
CALL GPPG2(1,4,2,BOX2)
CALL GPPG2(1,4,2,BOX3)
CALL GPPG2(1,4,2,BOX4)
CALL GPPG2(1,4,2,BOX5)
CALL GPPG2(1,4,2,BOX6)

*SET UP TEXT ATTRIBUTES AND DRAW THEM
CALL GPTXPR(3)
CALL GPTXAL(3,4)
CALL GPCHH(5)
CALL GPTWC(5)
CALL GPACFO(1,1)
CALL GPTXFO(1,1)
CALL GPTX2(POS1,4,TXT1)
CALL GPTX2(POS1,4,TXT2)
CALL GPTX2(POS1,4,TXT3)
CALL GPTX2(POS1,4,TXT4)
CALL GPTX2(POS1,4,TXT5)

CALL GPCLS

*ASSOCIATE STRUCTURE TO VIEW AND UPDATE WORKSTATION
CALL GPARYV(1,2,3,0.)
CALL GPAYP(1,2,3,0.)
CALL GPUPGR(1,2,3,0.)
RETURN
END

SUBROUTINE FSTCP

**********************************************************************************************************
SUBROUTINE FSTCP(U,H,B,P)

PROGRAM DESCRIPTION

THIS ROUTINE WILL CALCULATE THE CONTROL POINTS NEEDED TO CREATE A PERIODIC BICUBIC B-SPLINE SURFACE THAT PASSES THROUGH SIXTEEN POINTS SPECIFIED BY THE USER.

BY: ASHIT R. GANDHI
DATE: 11/12/88

PARAMETERS USED:
U = ARRAY OF PARAMETER (U) VALUES CORRESPONDING TO INPUT POINTS
W = ARRAY OF PARAMETER (W) VALUES CORRESPONDING TO INPUT POINTS
B = ARRAY OF POINTS THROUGH WHICH THE B-SPLINE CURVE SHOULD PASS
P = ARRAY OF CONTROL POINTS NEEDED TO GET THE SPECIFIED CURVE

**********************************************************************************************************

REAL*4 U(4), VU(4), WM(4), WM(4), UM(4,4), HM(4,4)
REAL*4 AU(4,4), AUN(4,4), AUN(4,4), AM(4,4), AM(4,4)
REAL*4 B(4,4), P1(4,4), P1(4,4), P2(4,4), P3(4,4), P4(4,4)

DATA M /-1.0, 3.0, -3.0, 1.0, 1.0, 0.0, 2.0, 0.0, 1.0, 0.0, 1.0, 0.0/
DATA MT /-1.0, 3.0, -3.0, 1.0, 1.0, 0.0, 2.0, 0.0, 1.0, 0.0, 1.0, 0.0/
FOR EACH VALUE OF U AND H COMPUTE THE PARAMETER VECTOR AND ASSEMBLE THE
COMPLETE 4X4 MATRICES
    DO 200 I = 1,4
    CALL FUNVT(U(I),VU)
    CALL FUNVT(H(I),VN)
    DO 100 J = 1,4
        UM(I,J) = VU(J)
        HMT(I,J) = VN(J)
    100 CONTINUE
200 CONTINUE

MULTIPLY MATRIX(U) AND MATRIX(H) TO GET MATRIX(AU) AND THEN SCALE
MULTIPLY MATRIX(MT) AND MATRIX(NT) TO GET MATRIX(AM) AND THEN SCALE
MULTIAX(AM)
    CALL FMT44(UM,M,AU)
    CALL FMT44(HM,NT,AN)
    DO 400 I = 1,4
        DO 500 J = 1,4
            AU(I,J) = AU(I,J)*27.0
            AN(I,J) = AN(I,J)*27.0
        500 CONTINUE
400 CONTINUE

COMPUTE THE CLOSED FORM INVERSE OF MATRIX(AU) AND SCALE MATRIX(AUINV)
COMPUTE THE CLOSED FORM INVERSE OF MATRIX(AM) AND SCALE MATRIX(AMINV)
    CALL MATINV(AU,AUINV)
    CALL MATINV(AM,AMINV)
    DO 600 I = 1,4
        DO 500 J = 1,4
            AUINV(I,J) = AUINV(I,J)*162.0
            AMINV(I,J) = AMINV(I,J)*162.0
        500 CONTINUE
600 CONTINUE

ASSIGN APPROPRIATE VALUES OF COORDINATES TO THE ARRAY
    DO 800 I = 1,4
        DO 700 J = 1,4
            B1(I,J) = B1(I,J,1)
            B2(I,J) = B2(I,J,2)
            B3(I,J) = B3(I,J,3)
        700 CONTINUE
800 CONTINUE

COMPUTE THE CONTROL POINTS COORDINATES
    CALL FMT44(AUINV,B1,P1)
    CALL FMT44(P1,AMINV,P1)
    CALL FMT44(AUINV,B2,P2)
    CALL FMT44(P2,AMINV,P2)
    CALL FMT44(AUINV,B3,P3)
    CALL FMT44(P3,AMINV,P3)

ASSIGN APPROPRIATE VALUES OF COORDINATES TO THE ARRAY
    DO 1000 I = 1,4
        DO 900 J = 1,4
            P1(I,J,1) = P1(I,J,1)
            P1(I,J,2) = P1(I,J,2)
            P1(I,J,3) = P1(I,J,3)
        900 CONTINUE
1000 CONTINUE
RETURN
END

SUBROUTINE FSTOR

*******************************************************************************
*******************************************************************************
** SUBROUTINE FSTOR(IPGN,I,J,MU,MH,PT,PGN) **
**
** PROGRAM DESCRIPTION **
**
** THIS SUBROUTINE WILL TAKE ALL POINTS EXISTING ON A PATCH AND **
** STORE THEM SO THAT A POLYGON COULD PASS THROUGH THEM. **
**
** BY: ASHIT R. GANDHI **
** DATE: 10/24/88 **
**
** PARAMETERS USED: **
**
*******************************************************************************
*******************************************************************************

Appendix B - FeatureMod Program Listing 241
** I = THE PATCH NUMBER (INTEGER,I/P)  
** J = SURFACE NUMBER (INTEGER,I/P)  
** NU = NUMBER OF LINES IN THE U DIRECTION (INTEGER,I/P)  
** NM = NUMBER OF LINES IN THE M DIRECTION (INTEGER,I/P)  
** P = POINTS IN THE PATCH (REAL,NM-1,JO)  
** PGN = POINTS REPRESENTING THE POLYGON (REAL(3,20000,5),O/P)  
**  
SUBROUTINE FSTROR(I,P,GN;I,J,NU,NM,PT,PGN)  
INTEGER*64 I,J,NU,NM  
REAL*4 PT(20,20,3), PGN(3,20000,5), PTS(3)  
REAL*4 PLOC(3), PORI(3)  
COMMON/REND/PLOC,PORI  
**COMPUTE 4 END POINTS FOR EACH POLYGON  
**FIRST TRAVERSE THE M DIRECTION TO CREATE (NM-1) POLYGONS  
**INCREMENT POLYGON NUMBER  
IPGN = IPGN + 1  
**COMPUTE THE FIRST TWO POINTS DEFINING THE POLYGON  
**TRAVERSE X,Y AND Z COORDINATES  
PGN(I2,IPGN,I1) = PTS(I2)  
**CONTINUE  
100 CONTINUE  
**COMPUTE THE LAST TWO POINTS DEFINING THE POLYGON  
**TRAVERSE X,Y AND Z COORDINATES  
PGN(I2,IPGN,I1) = PTS(I2)  
**CONTINUE  
300 CONTINUE  
400 CONTINUE  
500 CONTINUE  
600 CONTINUE  
RETURN  
END.  

SUBROUTINE FSTRG  

PROGRAM DESCRIPTION  
THIS SUBROUTINE GETS A STRING INPUT FROM THE USER. THE STRING INPUT IS TAKEN FROM AN ACTIVATED AREA RESERVED FOR SUCH INPUT  
BY: ASHIT R. GANDHI  
DATE: 10/08/88  
PARAMETERS USED:
** ** STRG = STRING INPUT (CHARACTER, O/P) ** **

*************************************************************************
SUBROUTINE FSTRG(STRG)
*************************************************************************
COMMON CSIZE
INTEGER*4 L
REAL*4 CSIZE(3), AREA(6)
CHARACTER STRG*50, EMTY*50, PREF*9, M$SG*59
DATA EMTY/' '/
STRG = '

*SETUP STRING INPUT AREA
AREA(1)=0.007*CSIZE(1)
AREA(2)=0.845*CSIZE(1)
AREA(3)=0.15*CSIZE(2)
AREA(4)=0.17*CSIZE(2)
AREA(5)=0.0
AREA(6)=CSIZE(3)

*INITIALIZE THE STRING DEVICE
CALL GPSTM0(1111112)

*INITIALIZE THE STRING DEVICE
CALL GPINST(111 150,EMTY,1,AREA,50,1,0,EMTY)

*PLACE STRING DEVICE IN THE REQUEST MODE
CALL GPSTM0(111,3,2)

*PROCESS THE STRING EVENT
*AWAIT EVENT
3 CALL GPANEV(1000,1,ICLA,IDEV)

*GET STRING
L=50
CALL GPGTST(L,LR,STRG)
IF(STRG.EQ.EMTY)GOTO 3

*RETURN STRING
PREF = 'STRING = 'M$SG = PREF//STRG
CALL FM$SG(30,M$SG)
CALL GPVPl16,2,1)
RETURN
END

*************************************************************************
SUBROUTINE FTITL
*************************************************************************

* PROGRAM DESCRIPTION *
* THIS SUBROUTINE SETS UP AND DRARS THE TITLE SCREEN FOR FEATURE.*

* BY: ASHIT R. GANDHI *
* DATE: 10/08/88 *
* PARAMETERS USED:
* NONE *

*************************************************************************
SUBROUTINE FTITL
INTEGER*4 LNT1, LNT2, LNT3
REAL*4 POS(2), POS2(2), POS3(2), PTS(12)
SUBROUTINE FTPPV

******************************************************************************
******************************************************************************
** SUBROUTINE FTPPV(INVIM) **
** **
** PROGRAM DESCRIPTION **
******************************************************************************
******************************************************************************
*THIS SUBROUTINE WILL DISPLAY THE TOP VIEW OF ANY OBJECT.*

BY: ASHIT R. GANDHI

DATE: 11/07/88

PARAMETERS USED:

** NVIEW = VIEW IN WHICH THE TOP VIEW IS TO BE DISPLAYED **

(INTEGER, I/P) **

********************************************************************************

** SUBROUTINE FTOPV(NVIEW)**

REAL*4 MAT1(4,4), MAT1(4,4), MAT2(4,4), SCALEF(3)

COMMON /SCALE/SCALEF

PI = 3.14159

DL = PI/2.0

*ROTATE THE VIEW ELEMENT ABOUT THE X-AXIS BY 90 DEGREES

CALL GPROTX(DL,MAT2)

*SCALE THE MODEL

CALL GPSCS(SCALEF,MAT1)

*COMPOSE THE TRANSFORMATION

CALL GPCMT3(MAT2,MAT1,MAT)

*ASSOCIATE THIS ROTATION TO THE VIEW

CALL GPVMT3(1,NVIEW,MAT)

CALL GPVP(1,NVIEW,2,1)

RETURN

END

********************************************************************************

** SUBROUTINE FTRNS **

********************************************************************************

** SUBROUTINE FTRNS(LOC,DCS,PTS) **

** PROGRAM DESCRIPTION **

** THIS SUBROUTINE WILL TRANSFORM THE COORDINATES OF A POINT IN **

THE X-O-Y PLANE TO THE CORRESPONDING POINT ON AN ARBITRARY **

PLANE. THE VARIABLE CONTAINING THE OLD LOCATION OF THE POINT **

IS RETURNED WITH THE NEW VALUES.

** BY: ASHIT R. GANDHI **

** DATE: 10/08/88 **

** PARAMETERS USED: **

** LOC = LOCATION OF THE ORIGIN OF NEW PLANE (REAL(3),I/P) **

** DCS = ANGULAR ROTATIONS OF PLANE WITH RESPECT TO THE AXES. **

** (INTEGER, I/P) **

** PTS = LOCATION OF THE POINT ON THE OLD (NEW) PLANE. **

** (REAL(3),I/P,0/P) **

**

********************************************************************************

** SUBROUTINE FTRNS(LOC,DCS,PTS) **

REAL*4 DCS(3)

REAL*4 LOC(3), PTS(3), R(3,3), V(3), ROT(3)

PI = 3.14159

*CONVERT THE ROTATIONS FROM ANGLE TO RADIANS

DO 100 I = 1,3

ROT(I) = DCS(I)*PI/180.

100 CONTINUE

*CALCULATE THE UNIVERSAL ROTATION MATRIX

CZ = COS(ROT(3))

SZ = SIN(ROT(3))

CY = COS(ROT(2))
SY = SINS(ROT(I))
CX = COS(ROT(I))
SX = SIN(ROT(I))

RI1,1) = CY*CZ
RI1,2) = -CY*SZ
RI1,3) = SY

RI2,1) = SX*SY*CZ + CX*SZ
RI2,2) = -SX*SY*SZ + CX*CZ
RI2,3) = -SX*CY

RI3,1) = -CX*SY*CZ + SX*SZ
RI3,2) = CX*SY*SZ + SX*CZ
RI3,3) = CX*CY

**MATRIX MULTIPLICATION TO GET NEW ROTATED COORDINATES**

00 300 I = 1 3
V(I) = 0.0
DO 200 J = 1 3
V(I) = V(I) + R(I,J)*PTS(J)
200 CONTINUE
300 CONTINUE

**CALCULATE THE TRANSLATIONS AND RETURN VALUES TO OLD VARIABLE**

DO 400 I = 1,3
PTS(I) = V(I) + LOC(I)
400 CONTINUE
RETURN
END

:: SUBROUTINE FUNVTIRV,V ::
:: PROGRAM DESCRIPTION ::
:: THIS ROUTINE CREATES THE U AND W VECTORS FOR DIFFERENT VALUES ::
:: OF THE PARAMETER ::
:: BY: ASHIT R. GANDHI ::
:: DATE: 10/08/88 ::
:: PARAMETERS USED: ::
:: RV = SCALER VALUE OF PARAMETER (REAL,I/P) ::
:: V = 4X1 MATRICES (REAL,O/P) ::

SUBROUTINE FUNVTIRV(RV,V)
REAL RV, V(4)
V(1) = RV**3
V(2) = RV**2
V(3) = RV
V(4) = 1.0
RETURN
END

:: SUBROUTINE FWIND ::
:: PROGRAM DESCRIPTION ::
:: THIS SUBROUTINE SETS UP THE WINDOWS AND VIEWPORTS. ::
:: BY: ASHIT R. GANDHI ::
:: DATE: 10/07/88 ::
:: PARAMETERS USED: ::
:: MIN1 : ARRAY CONTAINING WINDOW COORDINATES ::
:: MIN2 : ARRAY CONTAINING WINDOW COORDINATES ::
:: MIN3 : ARRAY CONTAINING WINDOW COORDINATES ::
:: MIN4 : ARRAY CONTAINING WINDOW COORDINATES ::
** WIN5 : ARRAY CONTAINING WINDOW COORDINATES **

** WIN6 : ARRAY CONTAINING WINDOW COORDINATES **

** WIN7 : ARRAY CONTAINING WINDOW COORDINATES **

** VPT1 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT2 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT3 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT4 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT5 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT6 : ARRAY CONTAINING VIEWPORT COORDINATES **

** VPT7 : ARRAY CONTAINING VIEWPORT COORDINATES **

** PT : ARRAY CONTAINING COORDINATES OF VIEWING **

**************************************************************************************

SUBROUTINE FWIND

REAL*4 WIN1(4), VPT1(6), PT1(3)

REAL*4 WIN2(4), VPT2(6), PT2(3)

REAL*4 WIN3(4), VPT3(6), PT3(3)

REAL*4 WIN4(4), VPT4(6), PT4(3)

REAL*4 WIN5(4), VPT5(6), PT5(3)

REAL*4 WIN6(4), VPT6(6), PT6(3)

REAL*4 WIN7(4), VPT7(6), PT7(3)

REAL*4 WIN8(4), VPT8(6), PT8(3)

REAL*4 WIN9(4), VPT9(6), PT9(3)

REAL*4 WIN10(4), VPT10(6), PT10(3)

REAL*4 WIN11(4), VPT11(6), PT11(3)

*WINDOW AND VIEWPORT FOR TITLE SCREEN

DATA WIN1/-20.0,20.0,-20.0,20.0/
DATA VPT1/0.0,0.0,0.0,1.0,0.0,1.0/
DATA PT1/0.0,0.0,300.0/

*WINDOW AND VIEWPORT FOR SCREEN "A"

DATA WIN2/-100.0,100.0,-100.0,100.0/

*WINDOW AND VIEWPORT FOR PROMPT AREA

DATA WIN9/0.0,1.0,0.0,1.0/
DATA VPT9/0.005,0.845,0.18,0.205/

*WINDOW AND VIEWPORT FOR STRING INPUT AREA

DATA WIN6/0.0,1.0,0.0,1.0/
DATA VPT6/0.005,0.845,0.15,0.175/

*WINDOW AND VIEWPORT FOR SCROLL AREA

DATA WIN5/0.0,1.0,0.0,1.0/
DATA VPT5/0.005,0.845,0.005,0.145/

*WINDOW AND VIEWPORT FOR MENU AREA

DATA WIN4/0.0,1.0,0.0,1.0/
DATA VPT4/0.85,0.455,0.15,0.940/

*WINDOW AND VIEWPORT FOR DISPLAY AREA

DATA WIN7/-100.0,100.0,-100.0,100.0/
DATA VPT7/0.005,0.845,0.21,0.995,0.0,1.0/

*WINDOW AND VIEWPORT FOR MULTIPLE DISPLAY AREA

*FRONT VIEW

DATA WIN8/-100.0,100.0,-100.0,100.0/
DATA VPT8/0.005,0.425,0.21,0.6025,0.0,1.0/

*TOP VIEW

DATA WIN9/-100.0,100.0,-100.0,100.0/
DATA VPT9/0.005,0.425,0.6025,0.995,0.0,1.0/

*RIGHT SIDE VIEW

DATA WIN10/-100.0,100.0,-100.0,100.0/
DATA VPT10/0.425,0.845,0.21,0.6025,0.0,1.0/

*ISOMETRIC VIEW

DATA WIN11/-100.0,100.0,-100.0,100.0/
DATA VPT11/0.425,0.845,0.6025,0.995,0.0,1.0/

*SET UP WINDOWS AND VIEWPORTS

CALL GPVM2P1(1,1,WIN1,VPT1,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,2,WIN2,VPT2,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,3,WIN3,VPT3,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,4,WIN4,VPT4,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,5,WIN5,VPT5,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,6,WIN6,VPT6,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,7,WIN7,VPT7,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,8,WIN8,VPT8,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,9,WIN9,VPT9,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,10,WIN10,VPT10,1,PT1,20,200.0,-200.0)

CALL GPVM2P1(1,11,WIN11,VPT11,1,PT1,20,200.0,-200.0)

*ACTIVATE THE VIEWS

CALL GPVCH1(1,1,1,1,2,1,1,2,1,2)

CALL GPVCH1(1,2,1,1,2,1,2,1,2)
**SUBROUTINE HEMIS**

The subroutine `HEMIS` computes the control points that would be required to create a hemisphere using B-spline surfaces.

**By:** Ashit R. Gandhi

**Date:** 11/21/88

**Parameters Used:**
- **RX** = Radius of the hemisphere
- **IFL** = Flag for new feature
- **ORI** = Orientation of hemisphere
- **LOC** = Location of the sphere (real, i/p)
- **ICOL** = Color for hemisphere (real, i/p)
- **BOOL** = Boolean type for hemisphere

**Subroutine HEMIS(RX, IFL, LOC, ORI, ICOL, BOOL)**

```fortran
INTEGER*4 ASNUM, OBNUM, INNUM, LTYPE, CNM
REAL*4 LOCl(3), PT(3), PTS(4,4,3), ORI(3)
REAL*4 UI(4), WI(4), PNTS(20,20,3), HS(2), US(4), ZS(4)
REAL*4 RX, IFL
REAL*4 PLOC(3), PORI(3)
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8
COMMON/PATCH/NUM,NUM, INNUM, LTYPE, CNM
COMMON/IDS/ASNUM, OBNUM, INNUM
COMMON/COMP/ASSEM, OBJECT
COMMON/PRPT/CNUM
COMMON/REND/PLOC, PORI
DATA U/0.0, 0.33333333, 0.66666667, 1.0/
DATA H/0.0, 0.33333333, 0.66666667, 1.0/
DATA HS/1.0, -1.0/
DATA US/1.0, -1.0, -1.0, 1.0/
DATA ZS/1.0, -1.0, -1.0, 1.0/
DO 50 I = 1, 4
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)
50 CONTINUE
PI = 3.14159
IF (BOOL .EQ. 'k') LTYPE = 1
IF (BOOL .EQ. 'b') LTYPE = 2
INNUM = INNUM + 1
IF (IFL .EQ. 2) THEN
   NSURF = 8
ELSE
   NSURF = 4
ENDIF
CALL FODINS(INNUM, LTYPE, 'HEMISPHERE', NSURF, CNUM, 1, OBJECT(CNUM-1000), LOC, ORI)
DELTA = PI/6.0
DO 600 I = 1, 11
   DO 500 IJ = 1, 4
      Theta = 0.0
      RETURN
END
```
DO 200 I = 1,4
ALPHA = 0.0
DO 100 J = 1,4
PT(1) = RX*COS(THETA)*COS(ALPHA)*X$(IK)
PT(2) = RX*SIN(THETA)*Y$(IK)
PT(3) = RX*COS(THETA)*SIN(ALPHA)*Z$(IK)
PTS(I,J,1) = PT(I)
PTS(I,J,2) = PT(J)
PTS(I,J,3) = PT(3)
ALPHA = ALPHA + DELTA
100 CONTINUE
THETA = THETA + DELTA
200 CONTINUE
CALL FSTCP(U,N,PTS,PT$(I,J))
CALL FBPCH('+',ICOL,LTYPE)
WRITE(10,10)INNUM,'4 4 0 0',ICOL,NU,NM
DO 250 I = 1,4
WRITE(10,*)PTS(I,J,1),J=1,4
WRITE(10,*)PTS(I,J,2),J=1,4
WRITE(10,*)PTS(I,J,3),J=1,4
WRITE(10,*)HO(1,J),J=1,4
250 CONTINUE
300 CONTINUE
600 CONTINUE
ICHK = 1
IF (IFL .EQ. Z) CALL RNG(0.0,RX,LOC,ORI,ICOL,BOOL,ICHK,0.)
RETURN

SUBROUTINE MBUSH

*** PROGRAM DESCRIPTION ***
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING **
** A BUSHING IN ITS CORRECT LOCATION AND ORIENTATION **
** BY: ASHIT R. GANDHI **
** DATE: 11/12/88 **
** PARAMETERS USED: **
** R1 = INNER RADIUS **
** R2 = OUTER RADIUS **
** THK = THICKNESS OF THE BUSHING **
** LOCA = LOCATION OF THE BUSHING **
** ORI = ORIENTATION OF THE BUSHING **
** COLOR = COLOR TO BE GIVEN TO THE BUSHING **
** BOOL = BOOLEAN OPERATOR FOR THE BUSHING **

*** COMMON/PATCH/NJ,NM ***

REAL*4 LOCA(3),POS(2),ORI(3)
REAL*4 ASIZE(6),CSIZE(3),DATA(12)
REAL*4 PAREA(6),SAREA(6),VAREA(6)
REAL*4 R1,R2
INTEGER*4 COLOR,PPATH(3)
CHARACTER STRG*32,LOCAL*7,LOCAL2*7,LOCAL3*7,LOCAL*26
CHARACTER CORI*7,ORI2*7,ORI3*7,ORI*26
CHARACTER BOOL*1,CCOLOR*1
COMMON/PATCH/NJ,NM

*DEFINE GENERIC MODEL*
R1 = 1.0
R2 = 2.0
THK = 1.0
NJ = 4
NM = 4
LOCAL(1) = 0.0
LOCAL(2) = 0.0
LOCAL(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'
COLOR = 2

SUBROUTINE MBUSH(R1,R2,THK,LOCA,ORI,COLOR,BOOL)
*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPQADS(1,ERRIND,UNITS,CSIZE,ASIZE)

* PREPARE FOR PICK INPUT *

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.005*CSIZE(1)
PAREA(2)=0.045*CSIZE(1)
PAREA(3)=0.005*CSIZE(2)
PAREA(4)=0.145*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPEST(7)
CALL GPPOPST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*CONVERT NUMBERS TO TEXT
WRITE(CR1,'(F7.2)')R1
WRITE(CR2,'(F7.2)')R2
WRITE(CTH1,'(F7.2)')TH1
WRITE(CNU,'(I2)')NU
WRITE(CNW,'(I2)')NW
WRITE(CCOLOR,'(I2)')COLOR
WRITE(CLOCA1,'(F7.2)')LOCA(1)
WRITE(CLOCA2,'(F7.2)')LOCA(2)
WRITE(CLOCA3,'(F7.2)')LOCA(3)
WRITE(CORI1,'(F7.2)')ORI(1)
WRITE(CORI2,'(F7.2)')ORI(2)
WRITE(CORI3,'(F7.2)')ORI(3)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'BUSH = ' //BOOL
CALL GPPKID(1)
CALL GPAN2(POS,9,STRG)

POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = ' //CCOLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)

POS(1) = 0.62
POS(2) = 0.84
STRG = 'LOCATION = ' //CLOCA1//CLOCA2//CLOCA3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = ' //CORI1//CORI2//CORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.15
STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80
STRG = 'NU = '//CNU
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.32
POS(2) = 0.67
STRG = 'NN = '//CNU
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'R1 = '//CR1
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67
STRG = 'R2 = '//CR2
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.54
STRG = 'THICK = '//CTHK
CALL GPPKID(11)
CALL GPAN2(POS,16,STRG)
CALL GPCLST
*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPORV(1,5,5)
*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,7,0)
*UPDATE WORKSTATION
CALL FPRMP(14)
ICLASS = 2
CALL GPPKF(1,1,1,ICLASS,1,1)
*WAIT FOR A PICK
100 CALL GPAVM(1,1,ICLASS,IDEV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPCTPK(1,1,PPATH)
IPKID = PPATH(2)
GOTO (1,2,3,4,5,6,7,8,9,10,11)IPKID
1 CALL FGST(BOOL)
IF (BOOL .EQ. '+' .AND. BOOL .EQ. '-' .AND. BOOL .EQ. '*') THEN
GOTO 100
ENDIF
GOTO 50
2 CALL FGSTT(CCOLOR)
IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
READICCOLOR = ICOLOR
IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
GOTO 50
ELSE
GOTO 100
ENDIF
ELSE
GOTO 100
ENDIF
3 CALL FGSTT(CLOCA)
IF (CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9')
& OR (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-') THEN
READCLOCA = ICLOCA
GOTO 50
ELSE
GOTO 100
ENDIF
CALL FGTSTICORI
IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9')
   .OR. (CORI(1:1) .EQ. '*' .OR. CORI(1:1) .EQ. '-') )
   THEN
   READ(CORI,*)ORI(1),ORI(2),ORI(3)
   GOTO 50
ELSE
   GOTO 100
ENDIF
5 CALL BUSH(R1,R2,THK,LOCA,ORI,COLOR,BOOL)
6 GOTO 200
7 CALL FGTSTICNU
IF ((CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9')
    THEN
   READ(CNU,*)INU
   IF(INU.GT.1.AND.INU.LT.20)INU = INU
   GOTO 50
ELSE
   GOTO 100
ENDIF
8 CALL FGTSTICNN
IF ((NN(1:1) .GE. '0' .AND. NN(1:1) .LE. '9')
    THEN
   READ(NN,*)NN
   IF(NN.GT.1.AND.NN.LT.20)NN = NN
   GOTO 50
ELSE
   GOTO 100
ENDIF
9 CALL FGTSTICRI
IF ((CR1(1:1) .GE. '0' .AND. CR1(1:1) .LE. '9')
    THEN
   READ(CR1,*)R1
   GOTO 50
ELSE
   GOTO 100
ENDIF
10 CALL FGTSTICR2
IF ((CR2(1:1) .GE. '0' .AND. CR2(1:1) .LE. '9')
    THEN
   READ(CR2,*)R2
   GOTO 50
ELSE
   GOTO 100
ENDIF
11 CALL FGTSTICTHK
IF ((CTHK(1:1) .GE. '0' .AND. CTHK(1:1) .LE. '9')
    THEN
   READ(CTHK,*)THK
   GOTO 50
ELSE
   GOTO 100
ENDIF
*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200 CALL GPPKMO(1,1,1,2)
*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.995*CSIZE(1)
PAREA(3)=0.15*CSIZE(2)
PAREA(4)=0.94*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)
*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)
*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GDPVR(1,5,7)
*ASSOCIATE MENU TO VIEW 5
CALL GPARY(1,5,5,0)
RETURN
END

SUBROUTINE MCONE

***************************
** SUBROUTINE MCONE(RS,LEN,LOCA,ORI,COLOR,BOOL) **
*------------------------------
PROGRAM DESCRIPTION

THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING A CONE IN ITS CORRECT LOCATION AND ORIENTATION

BY: ASHIT R. GANDHI

DATE: 11/12/88

PARAMETERS USED:

RS = RADIUS OF THE CONE
LEN = LENGTH OF THE CONE
LOCA = LOCATION OF THE CONE
ORI = ORIENTATION OF THE CONE
COLOR = COLOR TO BE GIVEN TO THE CONE
BOOL = BOOLEAN OPERATOR FOR THE CONE

PARAMETERS USED: RS, LEN, LOCA, ORI, COLOR, BOOL

SUBROUTINE MCONE(IRs, LEN, LOCA, ORI, COLOR, BOOL)

REAL*4 LOCA(3), ORI(3)
REAL*4 PAREA(6), SAREA(6), VAREA(6)
REAL*4 RS, LEN, AR
INTEGER*4 COLOR, PPATH
CHARACTER STRG*32, CLOCA*7, CLOCA2, CLOCA3, CORI*26
CHARACTER CORI*26, CORI1, CORI2, CORI3, CORI*26
CHARACTER BOOL*1, CCOLOR*2
CHARACTER CRS*7, CLEN*7, CAR*7, CNU*7, CNM*7

COMMON/PATCH/RST, NUM

*DEFINE GENERIC MODEL
RS = 1.0
LEN = 5.0
AR = 0.0
CTYPE = 'NO CAPS'
IFL = 1
NU = 4
NM = 4
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPQADS(1, ERRIND, UNITS, CSIZE, ASIZE)

* PREPARE FOR PICK INPUT *
*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1, 1, 1, 2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.025*CSIZE(2)
PAREA(3) = 0.015*CSIZE(3)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPPNPK(1, 1, 0, PPATH, 1, PAREA, 0, DATA, 1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1, 1, 1, 2)

*OPEN STRUCTURE
CALL GPESTR(1, 7)
CALL GPADCT(7)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*DRAW ALL TITLE TEXT
POSl1) = 0.62
POSl2) = 0.80

STRG = 'CONE' = '/BOOL
CALL GPPKID(1)
CALL GPAN2(POS,9,STRG)
WRITE(CRS,'(F7.2)'),RS
WRITE(CLEN,'(F7.2)'),LEN
WRITE(CAR,'(F7.2)'),AR
WRITE(CNU,'(I2)'), NU
WRITE(CLOCA1,'(F7.2)'),LOCAl1
WRITE(CLOCA2,'(F7.2)'),LOCAl2
WRITE(CLOCA3,'(F7.2)'),LOCAl3
WRITE(CORI1,'(F7.2)'),ORI(1)
WRITE(CORI2,'(F7.2)'),ORI(2)
WRITE(CORI3,'(F7.2)'),ORI(3)

POSl1) = 0.62
POSl2) = 0.67

WRITE(CCOLOR,'(I2)'),COLOR
STRG = 'COLOR' = '/CCOLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)

POSl1) = 0.62
POSl2) = 0.54

STRG = 'LOCATION' = '/CLOCA1/CLOCA2/CLOCA3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)

POSl1) = 0.62
POSl2) = 0.41

STRG = 'ORIENT' = '/CORI1/CORI2/CORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)

POSl1) = 0.62
POSl2) = 0.28

STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)

POSl1) = 0.62
POSl2) = 0.18

STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)

POSl1) = 0.32
POSl2) = 0.80

STRG = 'NU' = '/CNU
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)

POSl1) = 0.32
POSl2) = 0.67

STRG = 'NUM' = '/CNM
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)

POSl1) = 0.02
POSl2) = 0.80

STRG = 'RADIUS' = '/CRS
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)

POSl1) = 0.02
POSl2) = 0.67

STRG = 'LENGTH' = '/CLEN
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)

POSl1) = 0.02
POSl2) = 0.54

STRG = 'ARATIO' = '/CAR
CALL GPPKID(11)
CALL GPAN2(POS,16,STRG)

POSl1) = 0.02
POSl2) = 0.41

STRG = 'TYPE' = '/CTYPE
CALL GPPKID(12)
CALL GPAN2(POS,16,STRG)
CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)

*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,7,0)

*UPDATE WORKSTATION
CALL FPRMP(14)

ICLASS = 2
CALL GPPKF(1,1,1,ICLASS,1,1)

*WAIT FOR A PICK
100 CALL GPAREV(1,1,ICLASS,1DEV)
   IF (ICLASS .NE. 5) GOTO 100
   CALL GPGETK(1,1,PPATH)
   IPKID = PPATH(2)
   GOTO (1,2,3,4,5,6,7,8,9,10,11,12)IPKID

1 CALL FGTST(BOOL)
   IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
      GOTO 100
   ENDIF
   GOTO 50

2 CALL FGTST(CCOLOR)
   IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
      READ(CCOLOR,*) CCOLOR
      IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
         GOTO 50
      ELSE
         GOTO 100
      ENDIF
   ELSE
      GOTO 100
   ENDIF

3 CALL FGTST(CLOCA)
   IF (CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') THEN
      READ(CLOCA,*) CLOCA
      IF (CW .GT. 1 .AND. CW .LT. 20) CW = CW
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

4 CALL FGTST(CORI)
   IF (CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9') THEN
      READ(CORI,*) CORI
      IF (ORI .GT. 1 .AND. ORI .LT. 20) ORI = ORI
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

5 CALL CONE(IFL,RS,LEN,AR,LOCA,ORI,COLOR,BOOL,0,0,0)
   GOTO 200

7 CALL FGTST(CNU)
   IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
      READ(CNU,*) CNU
      IF (INU .GT. 1 .AND. INU .LT. 20) INU = INU
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

8 CALL FGTST(CNH)
   IF (CNH(1:1) .GE. '0' .AND. CNH(1:1) .LE. '9') THEN
      READ(CNH,*) CNH
      IF (INH .GT. 1 .AND. INH .LT. 20) INH = INH
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

9 CALL FGTST(CRS)
   IF (CRS(1:1) .GE. '0' .AND. CRS(1:1) .LE. '9') THEN
      READ(CRS,*) CRS
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

10 CALL FGTST(CLEN)
IF (CLEN(1:1) .GE. '0' .AND. CLEN(1:1) .LE. '9') THEN
    READ(CLEN(1:1)),LEN
    GOTO 50
ELSE
    GOTO 100
ENDIF

11  CALL FGTST(CAR)
    IF (CAR(1:1) .GE. '0' .AND. CAR(1:1) .LE. '9') THEN
        READ(CAR(1:1)),AR
        GOTO 50
    ELSE
        GOTO 100
    ENDIF

12  IF (CTYPE .EQ. 'NO CAPS') THEN
        CTYPE = 'CAPPED'
        IFL = 2
        GOTO 50
    ELSE
        CTYPE = 'NO CAPS'
        IFL = 1
        GOTO 50
    ENDIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200 CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.95*CSIZE(1)
PAREA(3)=0.15*CSIZE(2)
PAREA(4)=0.95*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,0,PATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,7)

*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,5,0)
RETURN
END

SUBROUTINE MCOUNT

*******************************************************************************
*** SUBROUTINE MCOUNT(R1,R2,L1,L2,IFL,IFL1,LOCA,ORI,COLOR,BOOL) ***
*** PROGRAM DESCRIPTION ***
*** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING ***
*** A COUNTER BORE OR A COUNTER SINK IN ITS CORRECT LOCATION ***
*** AND ORIENTATION ***
*** BY: ASHIT R. GANDHI ***
*** DATE: 11/12/88 ***
*** PARAMETERS USED: ***
*** R1 = RADIUS AT TOP OF FEATURE ***
*** R2 = INTERMEDIATE RADIUS OF FEATURE ***
*** L1 = LENGTH BETWEEN TOP AND INTERMEDIATE RADIUS ***
*** L2 = LENGTH OF FEATURE ***
*** IFL = FLAG INDICATING CAPS FOR COUNTER BORE ***
*** IFL1 = FLAG INDICATING CAPS FOR COUNTER SINK ***
*** LOCA = LOCATION OF THE FEATURE ***
*** ORI = ORIENTATION OF THE FEATURE ***
*** COLOR = COLOR TO BE GIVEN TO THE FEATURE ***
*** BOOL = BOOLEAN OPERATOR FOR THE FEATURE ***
*******************************************************************************

*******************************************************************************

REAL*4 LOCA(3),POS(2),ORI(5),R1,R2,L1,L2
REAL*4 ASIZE(6),CSIZE(3),DATA(12)
REAL*4 PAREA(6),SAREA(6),VAREA(6)

Appendix B  - FeatureMod Program Listing 256
INTEGER*4 COLOR, PPATH(3)
CHARACTER STRG*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA*26
CHARACTER CORI1*7, CORI2*7, CORI3*7, CORI*26
CHARACTER BOOL*1, CCOLOR*2
COMMON/PATH/NJROI

*DEFINE GENERIC MODEL
R1 = 2.0
R2 = 1.0
L1 = 1.0
L2 = 2.0
NU = 6
NM = 6
CTYPE = 'NO CAPS'
IFL = 1
CTYPE1 = 'BORE'
IFL1 = 1
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1, ERRIND, UNITS, CSIZE, ASIZE)

* PREPARE FOR PICK INPUT

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,0, PPATH, 1, PAREA, 0, DATA, 1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPEST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(1)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'COUNT = '+BOOL
CALL GPAN2(POS, 9, STRG)

WRITE(CR1,'(F7.2)')R1
WRITE(CR2,'(F7.2)')R2
WRITE(CL1,'(F7.2)')L1
WRITE(CL2,'(F7.2)')L2
WRITE(CNU,'(I2)')NU
WRITE(CNW,'(I2)')NW
WRITE(CCOLOR,'(I2)')COLOR
WRITE(CLOCA1,'(F7.2)')LOCA(1)
WRITE(CLOCA2,'(F7.2)')LOCA(2)
WRITE(CLOCA3,'(F7.2)')LOCA(3)
WRITE(CORI1,'(F7.2)')ORI(1)
WRITE(CORI2,'(F7.2)')ORI(2)
WRITE(CORI3,'(F7.2)')ORI(3)
POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = '/COLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)
POS(1) = 0.62
POS(2) = 0.84

STRG = 'LOCATION = '/CLOC1//CLOC2//CLOC3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.41

STRG = 'ORIENT = '/CORI1//CORI2//CORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.28

STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.28

STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.62
POS(2) = 0.80

STRG = 'MJ = '/CMJ
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.67

STRG = 'NW = '/CNW
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.80

STRG = 'RAD1 = '/CR1
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.67

STRG = 'RAD2 = '/CR2
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.54

STRG = 'LEN1 = '/CL1
CALL GPPKID(11)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.41

STRG = 'LEN2 = '/CL2
CALL GPPKID(12)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.28

STRG = 'TYPE = '/CTYPE
CALL GPPKID(13)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.62
POS(2) = 0.15

STRG = 'COUNT = '/CTYPE1
CALL GPPKID(14)
CALL GPAN2(POS,16,STRG)

CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDVR(1,5,5)

*ASSOCIATE MENU TO VIEW 5

Appendix B - FeatureMod Program Listing
CALL GPARV(1,5,7,0)

*UPDATE WORKSTATION
CALL FPRMPI,14)
ICLASS = 5
CALL GPPKF(1,1,1,ICLASS,1,1)

*Await for a pick
100 CALL GPAWEV(1,1,ICLASS,1DEV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPGPKF(1,1,PPATH)
IPKID = PPATH(2)
GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13,14)IPKID

1 CALL FGTST(BOOL)
IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
  GOTO 100
ENDIF
GOTO 50

2 CALL FGTST(CCOLOR)
IF (CCOLOR(1:1) .GT. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
  READ(CCOLOR,*) CCOLOR
  IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
    GOTO 50
  ELSE
    GOTO 100
  ENDIF
ELSE
  GOTO 100
ENDIF

3 CALL FGTST(CLOCA)
IF (CLOCA(1:1) .GT. '0' .AND. CLOCA(1:1) .LE. '9') THEN
  READ(CLOCA,*) LOCA(1),LOCA(2),LOCA(3)
  IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
    GOTO 50
  ELSE
    GOTO 100
  ENDIF
ENDIF

4 CALL FGTST(CORI)
IF (CORI(1:1) .GT. '0' .AND. CORI(1:1) .LE. '9') THEN
  READ(CORI,*) ORI(1),ORI(2),ORI(3)
  IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
    GOTO 50
  ELSE
    GOTO 100
  ENDIF
ENDIF

5 CALL COUNT(R1,R2,L1,L2,IFL,IFL1,LOCA,ORI,COLOR,BOOL)
6 GOTO 200

7 CALL FGTST(CNU)
IF (CNU(1:1) .GT. '0' .AND. CNU(1:1) .LE. '9') THEN
  READ(CNU,*) INU
  IF (INU .GT. 1 .AND. INU .LT. 20) NU = INU
  GOTO 50
ELSE
  GOTO 100
ENDIF

8 CALL FGTST(CNM)
IF (CNM(1:1) .GT. '0' .AND. CNM(1:1) .LE. '9') THEN
  READ(CNM,*) INM
  IF (INM .GT. 1 .AND. INM .LT. 20) NM = INM
  GOTO 50
ELSE
  GOTO 100
ENDIF

9 CALL FGTST(CR1)
IF (CR1(1:1) .GT. '0' .AND. CR2(1:1) .LE. '9') THEN
  READ(CR1,*) CR1
  GOTO 50
ELSE
  GOTO 100
ENDIF

10 CALL FGTST(CR2)
IF (CR2(1:1) .GT. '0' .AND. CR2(1:1) .LE. '9') THEN
  READ(CR2,*) CR2
  GOTO 50
ELSE
  GOTO 100
ENDIF

11 CALL FGTST(CLL1)
SUBROUTINE MCYNDR

******************************************************************************
** SUBROUTINE MCYNDR(RS, LEN, LOCA, ORI, COLOR, BOOL) **
******************************************************************************
** PROGRAM DESCRIPTION **
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING **
** A CYLINDER IN ITS CORRECT LOCATION AND ORIENTATION **
** BY: ASHIT R. GANDHI **
** DATE: 11/12/88 **
** PARAMETERS USED: **
** RS = RADIUS OF THE CYLINDER **
** LOCA = LOCATION OF THE CYLINDER **
** ORI = ORIENTATION OF THE CYLINDER **
** COLOR = COLOR TO BE GIVEN TO THE CYLINDER **
** BOOL = BOOLEAN OPERATOR FOR THE CYLINDER **
******************************************************************************
*DEFINE GENERIC MODEL

RS = 1.0
LEN = 5.0
NJ = 4
NW = 4
CTYPE = 'NO CAP'
IFL = 1
LOCAl = 0.0
LOCAL2 = 0.0
LOCAL3 = 0.0
ORI1 = 0.0
ORI2 = 0.0
ORI3 = 0.0.
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1, ERRIND, UNITS, CSIZE, ASIZE)

* PREPARE FOR PICK INPUT *
*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1, 1, 1, 2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005 * CSIZE(1)
PAREA(2) = 0.845 * CSIZE(1)
PAREA(3) = 0.005 * CSIZE(2)
PAREA(4) = 0.145 * CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1, 1, 0, PPATH, 1, PAREA, 0, DATA, 1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1, 1, 3, 2)

*OPEN STRUCTURE
50 CALL GPEST(7)
CALL GPOPST(7)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*CONVERT NUMBERS TO TEXT
WRITE(CRS, '(F7.2)') RS
WRITE(LEN, '(F7.2)') LEN
WRITE(CNU, '(F7.2)') NJ
WRITE(CNW, '(F7.2)') NW
WRITE(COLOR, '(I1)') COLOR
WRITE(LOCAl, '(F7.2)') LOCAl1
WRITE(LOCAL2, '(F7.2)') LOCAL2
WRITE(LOCAL3, '(F7.2)') LOCAL3
WRITE(ORI1, '(F7.2)') ORI1
WRITE(ORI2, '(F7.2)') ORI2
WRITE(ORI3, '(F7.2)') ORI3

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'CYNDR = ' / BOOL
CALL GPPKID(1)
CALL GPANZ(POS, 9, STRG)
POS(1) = 0.62
POS(2) = 0.87
STRG = 'COLOR = '//'COLOR
CALL GPPKID(2)
CALL GPNZ2(PO$•29,STRG)
POS(1) = 0.62
POS(2) = 0.54

STRG = 'LOCATION = '//'CLOCA1//'CLOCA2//'CLOCA3
CALL GPPKID(3)
CALL GPNZ2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.41

STRG = 'ORIENT = '//'CORI1//'CORI2//'CORI3
CALL GPPKID(4)
CALL GPNZ2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.28

STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPNZ2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.15

STRG = 'ABORT'
CALL GPPKID(6)
CALL GPNZ2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80

STRG = 'NU = '//'CNU
CALL GPPKID(7)
CALL GPNZ2(POS,16,STRG)
POS(1) = 0.32
POS(2) = 0.67

STRG = 'NN = '//'CNN
CALL GPPKID(8)
CALL GPNZ2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80

STRG = 'RADIUS = '//'CRS
CALL GPPKID(9)
CALL GPNZ2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67

STRG = 'LENGTH = '//'CLEN
CALL GPPKID(10)
CALL GPNZ2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.94

STRG = 'TYPE = '//'CTYPE
CALL GPPKID(11)
CALL GPNZ2(POS,16,STRG)

CALL GPCLST

#DISSOSSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)

#ASSOCIATE MENU TO VIEW 5
CALL GPDRV(1,5,7,0)

#UPDATE WORKSTATION
CALL FPRMP(14)
ICLASS = 2
CALL GPPKID(1,1,ICLASS,1,1)

#WAIT FOR A PICK
100 CALL GPAHEV(1,.1,ICLASS,IDEV)
IF (ICLASS .NE. 5)GOTO 100
CALL GPGTPK(1,1,PPATH)
PKID = PPATH(2)
GOTO (1,2,3,4,5,6,7,8,9,10,11)IPKID
CALL FGTST(BOOL)
IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. 'X')THEN
  GOTO 100
ENDIF
GOTO 50

CALL FGTST(CCOLOR)
IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
  READ(CCOLOR,*)COLOR
  IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
    GOTO 50
  ELSE
    GOTO 100
  ENDIF
ELSE
  GOTO 100
ENDIF

CALL FGTST(CLOCA)
IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') .OR. (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-' )) THEN
  READ(CLOCA,*)LOCA(1);LOCA(2);LOCA(3)
  GOTO 50
ELSE
  GOTO 100
ENDIF

CALL FGTST(CORI)
IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9') .OR. (CORI(1:1) .EQ. '+' .OR. CORI(1:1) .EQ. '-' )) THEN
  READ(CORI,*)ORI(1);ORI(2);ORI(3)
  GOTO 50
ELSE
  GOTO 100
ENDIF

ICHK = 0
CALL CYNDRI(IFL;RS;LEN;LOCA,ORI,COLOR,BOOL,ICHK,0.)
GOTO 200

CALL FGTST(CNU)
IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
  READ(CNU,*)INU
  IF (INU .GT. 1 .AND. INU .LT. 20) NU = INU
  GOTO 50
ELSE
  GOTO 100
ENDIF

CALL FGTST(CNM)
IF (CMN(1:1) .GE. '0' .AND. CMN(1:1) .LE. '9') THEN
  READ(CMN,*)INM
  IF (INM .GT. 1 .AND. INM .LT. 20) NW = INM
  GOTO 50
ELSE
  GOTO 100
ENDIF

CALL FGTST(CRS)
IF (CRS(1:1) .GE. '0' .AND. CRS(1:1) .LE. '9') THEN
  READ(CRS,*)RS
  GOTO 50
ELSE
  GOTO 100
ENDIF

CALL FGTST(CLEN)
IF (CLEN(1:1) .GE. '0' .AND. CLEN(1:1) .LE. '9') THEN
  READ(CLEN,*)LEN
  GOTO 50
ELSE
  GOTO 100
ENDIF

IF (CTYPE .EQ. 'NO CAP') THEN
  CTYPE = 'CAPPEO'
  IFL = 2
  GOTO 50
ELSE
  CTYPE = 'NO CAP'
  IFL = 1
  GOTO 50
ENDIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.995*CSIZE(1)
PAREA(3)=0.1*CSIZE(2)
PAREA(4)=0.94*CSIZE(2)
PAREA(5)=0.0
SUBROUTINE MELPSOD

** PROGRAM DESCRIPTION **

THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING A ELLIPSOID IN ITS LOCATION AND ORIENTATION

** BY: ** ASHIT R. GANDHI

** DATE: ** 11/12/88

** PARAMETERS USED: **

- RX = RADIUS OF THE ELLIPSOID IN THE X-DIRECTION
- RY = RADIUS OF THE ELLIPSOID IN THE Y-DIRECTION
- LOCA = LOCATION OF THE ELLIPSOID
- ORI = ORIENTATION OF THE ELLIPSOID
- COLOR = COLOR FOR THE ELLIPSOID
- BOOL = BOOLEAN TYPE FOR ELLIPSOID

** COMMON/PATCH/NJ 

*DEFINE GENERIC MODEL *

RX = 1.0
RY = 1.0
NU = 6

LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0

BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE *

CALL GPGADS(1,ERRIND,UNITS,CSIZE,ASIZE)

SUBROUTINE MELPSOD(RX,Ry,LOCA,ORI,COLOR,BOOL)
PAREA(4) = 0.145 * CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPESTI(7)
CALL GPPOSTI(7)
CALL GPPACNI(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPASHCI(0.80)

WRITE(CRX,'(F7.2)') RX
WRITE(CRY,'(F7.2)') RY
WRITE(CMU,'(12)') MU
WRITE(CNN,'(12)') NW
WRITE(COLOR,'(F7.2)') COLOR
WRITE(CLOCA1,'(F7.2)') LOCA(1)
WRITE(CLOCA2,'(F7.2)') LOCA(2)
WRITE(CLOCA3,'(F7.2)') LOCA(3)
WRITE(CORI1,'(F7.2)') ORI(1)
WRITE(CORI2,'(F7.2)') ORI(2)
WRITE(CORI3,'(F7.2)') ORI(3)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'ELPSD = ' 'BOOL
CALL GPPKID(1)
CALL GPAN2(POS,10,STRG)

POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = ' 'COLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)

POS(1) = 0.62
POS(2) = 0.54
STRG = 'LOCATION = ' 'LOCATION
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = ' 'ORIENT
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)

POS(1) = 0.62
POS(2) = 0.15
STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)

POS(1) = 0.32
POS(2) = 0.80
STRG = 'MU = ' 'NU
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.32
POS(2) = 0.67
STRG = 'NM = ' 'NM
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.02
POS(2) = 0.80
```fortran
STRG = 'RAD X = '/CRX
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67
STRG = 'RAD Y = '/CRY
CALL GPPKID(IO)
CALL GPAN2(POS,16,STRG)
CALL GPCLST

*DISSOCIATE SCROLL MESSSAGES FROM VIEW 5
CALL GPRV(1,5,5)

*ASSOCIATE MENU TO VIEW 5
CALL GPRV(1,5,7,0)

*UPDATE WORKSTATION
CALL FPRMP(14)
ICLASS = 2
CALL GPKF(1,1,1,ICLASS,1,1)

*WAIT FOR A PICK
100 CALL GPANEV(1,1,ICLASS,IDEV)
   IF (ICLASS .NE. 5) GOTO 100
   CALL GPGTPK(1,1,PPATH)
   IPKID = PPATH(2)
   GOTO (1,2,3,4,5,6,7,8,9,10)IPKID

1 CALL FGTST(BOOL)
   IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
      GOTO 100
   ENDIF
   GOTO 50

2 CALL FGTST(CCOLOR)
   IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
      READ(CCOLOR,*) COLOR
      IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
         GOTO 50
      ELSE
         GOTO 100
      ENDIF
   ELSE
      GOTO 100
   ENDIF

3 CALL FGTST(CLOCA)
   IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') .OR. (CLOCA(1:1) .GE. '+') .OR. (CLOCA(1:1) .GE. '-')) THEN
      READ(CLOCA,*) LOCA(1), LOCA(2), LOCA(3)
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

4 CALL FGTST(CORI)
   IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9') .OR. (CORI(1:1) .GE. '+') .OR. (CORI(1:1) .GE. '-')) THEN
      READ(CORI,*) ORI(1), ORI(2), ORI(3)
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

5 CALL ELPSOR2(RX,RY,LOCA,ORI,COLOR,BOOL)
   GOTO 200

6 CALL ELPSOR2(RX,RY,LOCA,ORI,COLOR,BOOL)
   GOTO 200

7 CALL FGTST(CNU)
   IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
      READ(CNU,*) NU
      IF (NU .GT. 1 .AND. NU .LT. 20) NU = NU
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

8 CALL FGTST(CNM)
   IF (CNM(1:1) .GE. '0' .AND. CNM(1:1) .LE. '9') THEN
      READ(CNM,*) NM
      IF (NM .GT. 1 .AND. NM .LT. 20) NM = NM
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

Appendix B - FeatureMod Program Listing 266
ENIF
9  CALL FGTST(CRX)
   IF(CRX(1:1).GE.'0' .AND. CRX(1:1) .LE. '9') THEN
      READ(CRX,*10X)
      GOTO 50
   ELSE
      GOTO 100
   ENIF
10 CALL FGTST(CRY)
   IF(CRY(1:1).GE.'0' .AND. CRY(1:1) .LE. '9') THEN
      READ(CRY,*10X)
      GOTO 50
   ELSE
      GOTO 100
   ENIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200 CALL GPKPMD(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
   PAREA(1)=0.85*CSIZE(1)
   PAREA(2)=0.995*CSIZE(1)
   PAREA(3)=0.15*CSIZE(2)
   PAREA(4)=0.94*CSIZE(2)
   PAREA(5)=0.0
   PAREA(6)=CSIZE(3)
   CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATAl)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
200 CALL GPKPMD(1,1,1,2)

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
   CALL GPDVR(1,5,7)

*ASSOCIATE MENU TO VIEW 5
   CALL GPARVD(1,5,5,0)
   RETURN

END

:: SUBROUTINE MHEMIS(RX,LOCA,ORI,COLOR,BXL) ::
:: PROGRAM DESCRIPTION ::
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING 
** A SPHERE AND ITS LOCATION 
** BY: ASHIT R. GANDHI 
** DATE: 11/12/88 
:: PARAMETERS USED: 
** RS = RADIUS OF THE HEMISPHERE 
** LOCA = LOCATION OF THE HEMISPHERE 
** ORI = ORIENTATION OF THE HEMISPHERE 
** COLOR = COLOR FOR THE HEMISPHERE 
** BOOL = BOOLEAN TYPE FOR HEMISPHERE 
::

******************************************************************************************
** SUBROUTINE MHEMIS(RX,LOCA,ORI,COLOR,BXL) **
** PROGRAM DESCRIPTION **
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING 
** A SPHERE AND ITS LOCATION **
** BY: ASHIT R. GANDHI **
** DATE: 11/12/88 **
** PARAMETERS USED: **
** RS = RADIUS OF THE HEMISPHERE **
** LOCA = LOCATION OF THE HEMISPHERE **
** ORI = ORIENTATION OF THE HEMISPHERE **
** COLOR = COLOR FOR THE HEMISPHERE **
** BOOL = BOOLEAN TYPE FOR HEMISPHERE **

******************************************************************************************
** SUBROUTINE MHEMIS(RX,LOCA,ORI,COLOR,BXL) **
REAL*4 LOCA(3), ORI(3), POS(2)
REAL*4 ASIZE(6), CSIZE(3), DATA(12)
REAL*4 PAREA(6), SAREA(6), VAREA(6)
REAL*4 RX, RY
INTEGER*4 COLOR, PPATH(3)
CHARACTER*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA4*7
CHARACTER*2, CCOLOR*2, CORI1*7, CORI2*7, CORI3*7, CORI4*7
CHARACTER*1, CNUM*7, CRX*7, CRY*7, CNU*7

*DEFINE GENERIC MODEL
RX = 1.0
RY = 1.0
NU = 6
**Appendix B · FeatureMod Program Listing**

```
  INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
  CALL GPGADS(1,ERRIND,UNITS,CSIZE,ASIZE)

  CALL GPPKM(1,1,1,2)
  *DEFINE PICK AREAS AND INITIALIZE PICKS
  PAREA(1)=0.005*CSIZE(1)
  PAREA(2)=0.845*CSIZE(1)
  PAREA(3)=0.005*CSIZE(2)
  PAREA(4)=0.145*CSIZE(2)
  PAREA(5)=0.0
  PAREA(6)=CSIZE(3)
  CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

  CALL GPPKM(1,1,3,2)
  *OPEN STRUCTURE
  CALL GPEST(7)
  CALL GPOPST(7)
  CALL GPADCN(1,2)

  CALL GPTXCI(5)
  *SET ANNOTATION TEXT SIZE SCALE FACTOR
  CALL GPAHSC(0.80)
  WRITE(CRX,'(F7.2)') 1RX
  WRITE(CMN,'(I2)') NMN
  WRITE(CCOLOR,'(I2)')  COLOR
  WRITE(CLOCA1,'(F7.2)') 1oca(1)
  WRITE(CLOCAL2,'(F7.2)') 1oca(2)
  WRITE(CLOCAL3,'(F7.2)') 1oca(3)
  WRITE(CORI1,'(F7.2)') ORI(1)
  WRITE(CORI2,'(F7.2)') ORI(2)
  WRITE(CORI3,'(F7.2)') ORI(3)

  *DRAW ALL TITLE TEXT
  POS(1) = 0.62
  POS(2) = 0.80
  STRG = 'HSPHERE = '//BOOL
  CALL GPKID(1)
  CALL GPAN2(POS,10,STRG)
  POS(1) = 0.62
  POS(2) = 0.67
  STRG = 'COLOR = '//'COLOR
  CALL GPKID(2)
  CALL GPAN2(POS,29,STRG)
  POS(1) = 0.62
  POS(2) = 0.54
  STRG = 'LOCATION = '//'Loca1//'Loca2//'Loca3
  CALL GPKID(3)
  CALL GPAN2(POS,32,STRG)
  POS(1) = 0.62
  POS(2) = 0.41
  STRG = 'ORIENT = '//'ORI1//'ORI2//'ORI3
  CALL GPKID(4)
  CALL GPAN2(POS,32,STRG)
  POS(1) = 0.62
  POS(2) = 0.28
```
STRG = 'EXECUTE'
CALL GPPKID(6)
CALL GPANZ(POS, 7, STRG)
POS(1) = 0.62
POS(2) = 0.15

STRG = 'ABORT'
CALL GPPKID(6)
CALL GPANZ(POS, 5, STRG)
POS(1) = 0.32
POS(2) = 0.80

STRG = 'NJ = '//CNJ
CALL GPPKID(7)
CALL GPANZ(POS, 16, STRG)
POS(1) = 0.32
POS(2) = 0.67

STRG = 'NM = '//CNM
CALL GPPKID(8)
CALL GPANZ(POS, 16, STRG)
POS(1) = 0.02
POS(2) = 0.80

STRG = 'RADIUS = '//CRX
CALL GPPKID(9)
CALL GPANZ(POS, 16, STRG)
POS(1) = 0.02
POS(2) = 0.67

STRG = 'TYPE = '//CRY
CALL GPPKID(10)
CALL GPANZ(POS, 16, STRG)

CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1, 5, 5)

*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1, 5, 7, 0)

*UPDATE WORKSTATION
CALL FPRMP(14)
ICLASS = 2
CALL GPPKF(1, 1, ICLASS, 1, 1)

*AWAIT FOR A PICK
100 CALL GPANEV(1, 1, ICLASS, IDENV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPGTPK(1, 1, PPATH)
IPKID = PPATH(2)
GOTO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) IPKID
1 CALL FGSTST(BOOL)
IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
GOTO 100
ENDIF
GOTO 50
2 CALL FGSTST(CCOLOR)
IF ([CCOLOR(1:1)] .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
READ(CCOLOR, *) COLOR
IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
GOTO 50
ELSE
GOTO 100
ENDIF
ELSE
GOTO 100
ENDIF
3 CALL FGSTST(LOCVA)
IF ([LOCVA(1:1)] .GE. '0' .AND. LOCVA(1:1) .LE. '9')
& OR. ([LOCVA(1:1)] .EQ. '+' .OR. LOCVA(1:1) .EQ. '-') THEN
READ(LOCVA, *) LOCA(1), LOCA(2), LOCA(3)
GOTO 50
ELSE
GOTO 100
ENDIF
4 CALL FGSTST(CORI)
IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9')) .OR. (CORI(1:1) .EQ. '+' .0R. CORI(1:1) .EQ. '-') THEN
READ(CORI,*);ORI(1),ORI(2),ORI(3)
GOTO 50
ELSE
GOTO 100
ENDIF
5 CALL HEMISIRX(IFL,LOCA,ORI,COLOR,BOOL)
6 GOTO 200
7 CALL FGTSTICNU)
IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
READ(CNU,*)INU
IF (INU .GT. 1 .AND. INU .LT. 20) INU = INU
GOTO 50
ELSE
GOTO 100
ENDIF
8 CALL FGTSTICNW)
IF (CW(1:1) .GE. '0' .AND. CW(1:1) .LE. '9') THEN
READ(CW,*1NW
IF (INW .GT. 1 .AND. INW .LT. 20) INW = INW
GOTO 50
ELSE
GOTO 100
ENDIF
9 CALL FGTSTTCRX)
IF (CRX(1:1) .GE. '0' .AND. CRX(1:1) .LE. '9') THEN
READ(CRX,*1RX
GOTO 50
ELSE
GOTO 100
ENDIF
10 IF (CRY .EQ. 'NO CAP') THEN
CRY = 'CAPPED'
IFL = 2
GOTO 50
ELSE
CRY = 'NO CAP'
IFL = 1
GOTO 50
ENDIF
#PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200 CALL GPPKMO(1,1,1,2)
#DEFINE PICK AREAS AND INITIALIZE PICKS
 PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.995*CSIZE(1)
PAREA(3)=0.15*CSIZE(2)
PAREA(4)=0.94*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0;DATA,1)
#PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)
#DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDHV(1,5,7)
#ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,5,0)
RETURN
END

SUBROUTINE MPYRMID

******************************************************
******************************************************
** SUBROUTINE MPYRMID(A,B,C,AR,IFL,LOCA,ORI,COLOR,BOOL) **
** PROGRAM DESCRIPTION                                  **
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING **
** A PYRAMID IN ITS CORRECT LOCATION AND ORIENTATION     **
** BY: ASHIT R. GANDHI                                    **
******************************************************
******************************************************
Appendix B - FeatureMod Program Listing 270
** DATE: 11/12/88 **

** PARAMETERS USED: **

** A = LENGTH OF THE PYRAMID IN THE X-DIRECTION **
** B = LENGTH OF THE PYRAMID IN THE Y-DIRECTION **
** C = LENGTH OF THE PYRAMID IN THE Z-DIRECTION **
** LOCA = LOCATION OF THE PYRAMID **
** ORI = ORIENTATION OF THE PYRAMID **
** COLOR = COLOR TO BE GIVEN TO THE PYRAMID **
** BOOL = BOOLEAN OPERATOR FOR THE PYRAMID **

*************************************************************************************************************************************************************************************

SUBROUTINE MPYRMIDIA,B,C,AR,IFL,LOCA,ORI,COLOR,BOOL)
REAL*4 LOCA(3), POS(2), ORI(3), A, B, C, AR
REAL*4 ASIZE(6),CSIZE(3),DATA(12)
REAL*4 PAREA(6),SAREA(6),VAREA(6)
INTEGER*4 COLOR, PPATH(3)
CHARACTER STRG*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA 26
CHARACTER CORI1*7, CORI2*7, CORI3*7, CORI*26
CHARACTER BOOLx1, CCOLORx2
CHARACTER CA*7, CB*7; CC*7; CAR*7• CTYPE*7• CNU*7• CNW*7
CQMON/PATCH/MLM!

*DEFINE GENERIC MODEL
A = 1.0
B = 1.0
C = 1.0
AR = 0.0
NU = 2
NW = 2
CTYPE = 'NO CAPS'
IFL = 1
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPQADS(1,ERRIND,UUNITS,CSIZE,ASIZE)
*************************************************************************************************************************************************************************************

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1)=0.005*CSIZE(1)
PAREA(2)=0.845*CSIZE(1)
PAREA(3)=0.005*CSIZE(2)
PAREA(4)=0.145*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINPK(1,1,1,2,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPES(7)
CALL GPOPST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXC(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'PRISM = '/'BOOL
CALL GPPKID(1)
CALL GPAN2(POS•9,STRG)
POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = ' // COLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)
POS(1) = 0.62
POS(2) = 0.94
STRG = 'LOCATION = ' // LOCAL1 // LOCAL2 // LOCAL3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = ' // ORI1 // ORI2 // ORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.88
STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.18
STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80
STRG = 'NU = ' // CNU
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.32
POS(2) = 0.67
STRG = 'NM = ' // CNM
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'LENGTH = ' // CA
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67
STRG = 'BREATH = ' // CB
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'HEIGHT = ' // CC
CALL GPPKID(11)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.41
STRG = 'A RATIO = ' // CAR
CALL GPPKID(12)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.28
STRG = 'TYPE = ' // CTYP
CALL GPPKID(13)
CALL GPPAN2(POS,16,STRG)

CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)
*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,7,0)

*UPDATE WORKSTATION
CALL FPRMP(14)
ICLASS = 2
CALL GPPKF(1,1,1,ICLASS,1,1)

*AHAIT FOR A PICK
100 CALL GPAHEV(1,1,ICLASS,IDEV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPSTPK(1,1,PPATH)
IPKID = PPATH(2)
GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13)IPKID

1 CALL FGSTST(BOOL)
IF (BOOL .NE. ' ' .AND. BOOL .NE. ' ' .AND. BOOL .NE. ' ' ' ) THEN
END IF
GOTO 50

2 CALL FGSTST(COLOR)
IF (COLOR(1:1) .GE. '0' .AND. COLOR(1:1) .LE. '9') THEN
READ(COLOR) COLOR
IF (COLOR .GT. 1 .AND. COLOR .LE. 6) THEN
END IF
ELSE GOTO 100
ENDIF
GOTO 100

3 CALL FGSTST(CLOCA)
IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') .OR. (CLOCA(1:1) .GE. '+' .OR. CLOCA(1:1) .GE. '-') ) THEN
READ(CLOCA) LOCA(1), LOCA(2), LOCA(3)
GOTO 50
ELSE GOTO 100
ENDIF
GOTO 100

4 CALL FGSTST(CORI)
IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9') .OR. (CORI(1:1) .GE. '+' .OR. CORI(1:1) .GE. '-' ) ) THEN
READ(CORI) ORI(1), ORI(2), ORI(3)
GOTO 50
ELSE GOTO 100
ENDIF
GOTO 100

5 CALL PYRAMID(A,B,C,AR,IFL,LOCA,ORI,COLOR,BOOL)
GOTO 200

6 CALL FGSTST(CNU)
IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
READ(CNU) INU
IF (INU .GT. 1 .AND. INU .LT. 20) INU = INU
GOTO 50
ELSE GOTO 100
ENDIF
GOTO 100

7 CALL FGSTST(CNW)
IF (CNW(1:1) .GE. '0' .AND. CNW(1:1) .LE. '9') THEN
READ(CNW) INW
IF (INW .GT. 1 .AND. INW .LT. 20) INW = INW
GOTO 50
ELSE GOTO 100
ENDIF
GOTO 100

8 CALL FGSTST(CA)
IF (CA(1:1) .GE. '0' .AND. CA(1:1) .LE. '9') THEN
READ(CA) CA
GOTO 50
ELSE GOTO 100
ENDIF

Appendix B - FeatureMod Program Listing

273
CALL FGTST(CB)
IF (CB(1:1) .GE. '0' .AND. CB(1:1) .LE. '9') THEN
READ(CB,*)
GOTO 50
ELSE
GOTO 100
ENDIF

CALL FGTST(CC)
IF (CC(1:1) .GE. '0' .AND. CC(1:1) .LE. '9') THEN
READ(CC,*)
GOTO 50
ELSE
GOTO 100
ENDIF

CALL FGTST(CAR)
IF (CAR(1:1) .GE. '0' .AND. CAR(1:1) .LE. '9') THEN
READ(CAR,*)
GOTO 50
ELSE
GOTO 100
ENDIF

IF(TYPE .EQ. 'NO CAPS') THEN
IFL = 2
CTYPE = 'CAPPED'
GOTO 50
ELSE
IFL = 1
CTYPE = 'NO CAPS'
GOTO 50
ENDIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKM0(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.85*CSIZE(1)
PAREA(2) = 0.995*CSIZE(1)
PAREA(3) = 0.15*CSIZE(2)
PAREA(4) = 0.94*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,10,PATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKM0(1,1,1,2)

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDW(1,5,7)

*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,5,0)

RETURN
END

SUBROUTINE MRING

:: SUBROUTINE MRING(R1, R2, LOCA, ORI, COLOR, BOOL) ::
:: PROGRAM DESCRIPTION ::
:: THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING A RING IN ITS CORRECT LOCATION AND ORIENTATION ::
:: BY: ASHIT R. GANDHI ::
:: DATE: 11/12/88 ::
:: PARAMETERS USED: ::
:: R1 = INNER RADIUS FOR RING ::
:: R2 = OUTER RADIUS FOR RING ::
:: LOCA = LOCATION OF RING ::
:: ORI = ORIENTATION OF RING ::
:: COLOR = COLOR TO BE GIVEN TO RING ::
:: BOOL = BOOLEAN OPERATOR FOR RING ::

Appendix B · FeatureMod Program Listing 274
SUBROUTINE MRING(R1, R2, LOCA, ORI, COLOR, BOOL)

REAL*4 LOCA(3), POS(2), ORI(3)
REAL*4 ASIZE(6), CSIZE(6), DATA(12)
REAL*4 PAREA(6), SAREA(6), VAREA(6)
REAL*4 R1, R2
INTEGER*4 COLOR, PPATH(3)

CHARACTER STRG*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA*26
CHARACTER CORI1*7, CORI2*7, CORI3*7, CORI*26
CHARACTER BOOL*1, CCOLOR*7, CNU*7, CNM*7
CHARACTER CR1*7, CR2*7

COFMON/PATCH/NNU

*DEFINE GENERIC MODEL
R1 = 1.0
R2 = 2.0
NU = 4
NN = 4
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1, ERRIND, UNITS, CSIZE, ASIZE)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1, 1, 0, PPATH, 1, PAREA, 0, DATA, 1)

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPROMO(1, 1, 1, 2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

WRITE(CR1,'(F7.2)') R1
WRITE(CR2,'(F7.2)') R2
WRITE(CCOLOR,'(I2)') COLOR
WRITE(CNU,'(I2)') NU
WRITE(CNM,'(I2)') NN
WRITE(CLOCAL,'(F7.2)') LOCA(1)
WRITE(CLOCAL,'(F7.2)') LOCA(2)
WRITE(CLOCAL,'(F7.2)') LOCA(3)
WRITE(CORI1,'(F7.2)') ORI(1)
WRITE(CORI2,'(F7.2)') ORI(2)
WRITE(CORI3,'(F7.2)') ORI(3)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'RING = ' /BOOL
CALL GPPKID(1)
CALL GPAN2(POS, 9, STRG)
POS(1) = 0.62
POS(2) = 0.87
STRG = 'COLOR = ' /CCOLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)
POS(1) = 0.62
POS(2) = 0.94
STRG = 'LOCATION = '//CLOCA1//CLOCA2//CLOCA3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = '//CORI1//CORI2//CORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.15
STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80
STRG = 'NUM = '//CNU
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.32
POS(2) = 0.67
STRG = 'NUM = '//CNU
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'R1 = '//CR1
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67
STRG = 'R2 = '//CR2
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)
CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDHV(1, 5, 5)

*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1, 5, 7, 0)

*UPDATE WORKSTATION
CALL FPRLP(14)
ICLASS = 2
CALL GPPKF(1,1,1,ICLASS,1,1)

*WAIT FOR A PICK
100 CALL GPANEV(1,1,1,ICLASS,1,DEV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPGTPK(1,1,PPATH)
IPKID = PPATR(2)
GOTO (1,2,3,4,5,6,7,8,9,10)IPKID
1 CALL FGTSI(BOOL)
IF (BOOL .NE. '4' .AND. BOOL .NE. '1' .AND. BOOL .NE. '0' .AND. BOOL .NE. '*') THEN
GOTO 100
ENDIF
GOTO 50
2 CALL FGTSI(CCOLOR)
IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
READ(CCOLOR,*)COLOR

Appendix B · FeatureMod Program Listing 276
IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
  GOTO 50
ELSE
  GOTO 100
ENDIF
ELSE
  GOTO 100
ENDIF

3 CALL FGTST(CLOCA)
IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9')
& OR. (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-')) THEN
  READ(CLOCA,*),LOCA(1),LOCA(2),LOCA(3)
  GOTO 50
ELSE
  GOTO 100
ENDIF

4 CALL FGTST(CORI)
IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9')
& OR. (CORI(1:1) .EQ. '+' .OR. CORI(1:1) .EQ. '-')) THEN
  READ(CORI,*),ORI(1),ORI(2),ORI(3)
  GOTO 50
ELSE
  GOTO 100
ENDIF

5 ICHK = 0
CALL RING(R1,R2,LOCA,ORI,COLOR,BOOL,ICHK,0.)
GOTO 200

6 CALL FGTST(CNU)
IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
  READ(CNU,*),INU
  IF (INU .GT. 1 .AND. INU .LT. 20) INU = INU
  GOTO 50
ELSE
  GOTO 100
ENDIF

7 CALL FGTST(CNW)
IF (CNW(1:1) .GE. '0' .AND. CNW(1:1) .LE. '9') THEN
  READ(CNW,*),INW
  IF (INW .GT. 1 .AND. INW .LT. 20) INW = INW
  GOTO 50
ELSE
  GOTO 100
ENDIF

8 CALL FGTST(CR1)
IF (CR1(1:1) .GE. '0' .AND. CR1(1:1) .LE. '9') THEN
  READ(CR1,*),R1
  GOTO 50
ELSE
  GOTO 100
ENDIF

9 CALL FGTST(CR2)
IF (CR2(1:1) .GE. '0' .AND. CR2(1:1) .LE. '9') THEN
  READ(CR2,*),R2
  GOTO 50
ELSE
  GOTO 100
ENDIF

PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,1,2)

DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.85*CSIZE(1)
PAREA(2) = 0.95*CSIZE(1)
PAREA(3) = 0.15*CSIZE(2)
PAREA(4) = 0.95*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPKPM(1,1,3,2)

DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,7)

ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,1,2)
RETURN
END

Appendix B - FeatureMod Program Listing 277
SUBROUTINE MSLAB

PROGRAM DESCRIPTION

THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING A SLAB IN ITS CORRECT LOCATION AND ORIENTATION.

BY: ASHIT R. GANDHI

DATE: 11/12/88

PARAMETERS USED:

A = LENGTH OF THE SLAB
B = BREADTH OF THE SLAB
C = HEIGHT OF THE SLAB
LOCA = LOCATION OF THE SLAB
ORI = ORIENTATION OF THE SLAB
COLOR = COLOR TO BE GIVEN TO THE SLAB
BOOL = BOOLEAN OPERATOR FOR THE SLAB

******************************************************************************
SUBROUTINE MSLAB(A,B,C,LOCA,ORI,COLOR,BOOL)
******************************************************************************

REAL*4 LOCA(3), ORI(3), A, B, C
REAL*4 ASIZE(6), CSIZE(3), DATA(12)
REAL*4 PAREA(6), SAREA(6)
INTEGER*4 COLOR, PPATH(3)
CHARACTER*32 STRG, CLOCA1, CLOCA2, CLOCA3, CLOCA
CHARACTER*7 CORI1, CORI2, CORI3, CORI
CHARACTER*2 BOOL, COLOR
CHARACTER*7 CA, CB, CC, CNU, CNW
COMMON/PATCH/NU,NM

*DEFINE GENERIC MODEL
A = 1.0
B = 1.0
C = 1.0
NU = 2
NH = 2
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPQADS(1, ERRIND, UNITS, CSIZE, ASIZE)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPINPK(1,1,0, PPATH,1, PAREA,0, DATA,1)

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPKMO(1,1,2)

*OPEN STRUCTURE
50 CALL GPEST(7)
CALL GPUPST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW

Appendix B - FeatureMod Program Listing
CALL GPTXCII(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)
WRITE(CA,'(F7.2)')IA
WRITE(CC,'(F7.2)')IC
WRITE(CNU,'(I2)')IMU
WRITE(CNM,'(I2)')INM
WRITE(CCOLOR,'(I2)')ICOLOR
WRITE(CLOCA1,'(F7.2)')LOCAl
WRITE(CLOCA2,'(F7.2)')LOCAl2
WRITE(CLOCA3,'(F7.2)')LOCAl3
WRITE(CORI1,'(F7.2)')ORI1
WRITE(CORI2,'(F7.2)')ORI2
WRITE(CORI3,'(F7.2)')ORI3

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80
STRG = 'SLAB = '//BOOL
CALL GPPIKID(1)
CALL GPAN2(POS,9,STRG)
POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = '//CCOLOR
CALL GPPIKID(2)
CALL GPAN2(POS,29,STRG)
POS(1) = 0.62
POS(2) = 0.54
STRG = 'LOCATION = '//CLOCA1//CLOCA2//CLOCA3
CALL GPPIKID(3)
CALL GPAN2(POS,52,STRG)
POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = '//CORI1//CORI2//CORI3
CALL GPPIKID(4)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPPIKID(5)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.15
STRG = 'ABORT'
CALL GPPIKID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80
STRG = 'NU = '//CNU
CALL GPPIKID(7)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.32
POS(2) = 0.67
STRG = 'NM = '//CNM
CALL GPPIKID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'LENGTH = '//CA
CALL GPPIKID(9)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.67
STRG = 'BREATH = '//CB
CALL GPPIKID(10)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.54
STRG = 'HEIGHT = '//CC
CALL GPPIKID(11)

Appendix B - FeatureMod Program Listing 279
CALL GPAN2(POS,16,STRG)
CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)

*ASSOCIATE MENU TO VIEW 5
CALL GPDRV(1,5,7,0)

*UPDATE WORKSTATION
CALL FPMP(14)
  ICCLASS = 2
CALL GPDRF(1,1,1,ICCLASS,1,1)

*WAIT FOR A PICK
100  CALL GPMV(1,1,ICCLASS,IDEV)
    IF (ICCLASS .NE. 5) GOTO 100
    CALL GPGTPK(1,1,PPATH)
    IPKID = PPATH(2)
    GOTO (1,2,3,4,5,6,7,8,9,10,11)IPKID
1  CALL FGSTST(BOOL)
    IF (BOOL .NE. ' ' .AND. BOOL .NE. '-' .AND. BOOL .NE. 'W') THEN
        GOTO 100
    ENDIF
    GOTO 50
2  CALL FGSTST(CCOLOR)
    IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
        READ(CCOLOR,*)COLOR
        IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
            GOTO 50
        ELSE
            GOTO 100
        ENDS
        ELSE
            GOTO 100
    ENDS
3  CALL FGSTST(CLOCA)
    IF (CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9'
        OR. (CLOCA(1:1) .EQ. '+' OR. CLOCA(1:1) .EQ. '-')) THEN
        READ(CLOCA,*)LOCAl4
        GOTO 50
    ELSE
        GOTO 100
    ENDS
4  CALL FGSTST(CORI)
    IF (CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9'
        OR. (CORI(1:1) .EQ. '+' OR. CORI(1:1) .EQ. '-')) THEN
        READ(CORI,*)ORl(1:ORI(2):ORI(5)
        GOTO 50
    ELSE
        GOTO 100
    ENDS
5  CALL SLOB(A,B,C,LOCAl4,ORI, COLOR, BOOL)
7  GOTO 200
7  CALL FGSTST(CNU)
    IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
        READ(CNU,*)INU
        IF (INU .GT. 1 .AND. INU .LT. 20) NU = INU
        ELSE
            GOTO 100
        ENDS
8  CALL FGSTST(INM)
    IF (INM(1:1) .GE. '0' .AND. INM(1:1) .LE. '9') THEN
        READ(INM,*)IM
        IF (IM .GT. 1 .AND. IM .LT. 20) IM = INM
        ELSE
            GOTO 100
        ENDS
9  CALL FGSTST(ICA)
    IF (ICA(1:1) .GE. '0' .AND. ICA(1:1) .LE. '9') THEN
        READ(ICA,*)A
        ELSE
            GOTO 100
        ENDS

Appendix B - FeatureMod Program Listing 280
CALL FGTST(CB)
IF (CB(1:1) .GE. '0' .AND. CB(1:1) .LE. '9') THEN
    READ(CB,*)B
    GOTO 50
ELSE
    GOTO 100
ENDIF

CALL FGTST(CC)
IF (CC(1:1) .GE. '0' .AND. CC(1:1) .LE. '9') THEN
    READ(CC,*)C
    GOTO 50
ELSE
    GOTO 100
ENDIF

PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON

CALL GPPKMO(1,1,1,2)

DEFINE PICK AREAS AND INITIALIZE PICKS

PAREA(1)=0.85*CSIZE(1)
PAREA(2)=0.95*CSIZE(1)
PAREA(3)=0.15*CSIZE(2)
PAREA(4)=0.04*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINKP(1,1,0,PPATH,1,PAREA,0,DATA,1)

PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON

CALL GPPKMO(1,1,3,2)

DISSOCIATE SCROLL MESSAGES FROM VIEW 5

CALL GPDV(1,5,7)

ASSOCIATE MENU TO VIEW 5

CALL GPAR(1,5,5,0)

RETURN

END

SUBROUTINE MSPHERE(R$;LOCA;COLOR;BOOL)

PROGRAM DESCRIPTION

THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING
A SPHERE AND ITS LOCATION

BY: ASHIT R. GANDHI
DATE: 11/12/88

PARAMETERS USED:
RS = RADIUS OF THE SPHERE
LOCA = LOCATION OF THE SPHERE
COLOR = COLOR TO BE ASSIGNED TO SPHERE
BOOL = BOOLEAN OPERATOR FOR SPHERE

SUBROUTINE MSPHERE(RS,LOCA,COLOR,BOOL)

REAL*4 LOCA(3), POS(2), RS
REAL*4 ASIZE(6),CSIZE(3),DATA(12)
INTEGER*4 PAREA(6),SAREA(6),VAREA(6)
CHARACTER STRG*32, CRS*7, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA2*6
CHARACTER BOOL*1, CCOLOR*2, CNN*7, CNW*7
COMMON/PATCH/NUM

DEFINE GENERIC MODEL

RS = 1.0
NUM = 6
LOCA1 = 0.0
LOCA2 = 0.0
LOCA3 = 0.0
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE
CALL GPGADS(1,ERRIND,UNITS,CSIZE,ASIZE)

* PREPARE FOR PICK INPUT

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
CALL GPPOKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)
CALL GPPNPKI(1,1,0,PPTH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
CALL GPPOKMO(1,1,3,2)

*OPEN STRUCTURE
50 CALL GPEST1)
CALL GPUPST(7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW
CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR
CALL GPAHSC(0.80)

*CONVERT NUMBERS TO TEXT
WRITE(CRS,'(F7.2)')RS
WRITE(CNU,'(I2)')NU
WRITE(CNW,'(I2)')NW
WRITE(CLOCA1,'(F7.2)')LOCA(1)
WRITE(CLOCA2,'(F7.2)')LOCA(2)
WRITE(CLOCA3,'(F7.2)')LOCA(3)

*DRAW ALL TITLE TEXT
POS(1) = 0.62
POS(2) = 0.80

STRG = 'SPHERE = '//BOOL
CALL GPPKID(1)
CALL GPAN2(POS,10,STRG)
POS(1) = 0.62
POS(2) = 0.67
WRITE(CCOLOR,'(I2)')COLOR
STRG = 'COLOR = '//CCOLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)
POS(1) = 0.62
POS(2) = 0.54

STRG = 'LOCATION = '//CLOCA1//CLOCA2//CLOCA3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)
POS(1) = 0.62
POS(2) = 0.41

STRG = 'EXECUTE'
CALL GPPKID(4)
CALL GPAN2(POS,7,STRG)
POS(1) = 0.62
POS(2) = 0.26

STRG = 'ABORT'
CALL GPPKID(5)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.80

STRG = 'NU = '//CNJ
CALL GPPKID(6)
CALL GPAN2(POS,8,STRG)
APPENDIX B - FeatureMod Program Listing

POS(1) = 0.32
POS(2) = 0.67

STRG = 'NW = ' //CNM
CALL GPKPID(7)
CALL GPAN2(POS,8,STRG)

POS(1) = 0.02
POS(2) = 0.80

STRG = 'RADIUS = ' //CRS
CALL GPKPID(8)
CALL GPAN2(POS,16,STRG)

CALL GPCLST

DISASSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV(1,5,5)

ASSOCIATE MENU TO VIEW 5
CALL GPDRV(1,5,7,0)

UPDATE WORKSTATION
CALL FPRMP(14)

ICLASS = 2
CALL GPKF(1,1,ICLASS,1,1)

WAIT FOR A PICK

100 CALL GPANV(1,1,ICLASS,IDEV)
IF (ICLASS .NE. 5) GOTO 100
CALL GPGTPI(1,1;PPATH)
IPKID = PPATH(2)
GOTO (1,2,3,4,5,6,7,8)IPKID

1 CALL FGTST(BOOL)
IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
GOTO 100
ENDIF
GOTO 50

2 CALL FGTST(CCOLOR)
IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') THEN
READ(CCOLOR,*) COLOR
IF (COLOR .GE. 1 .AND. COLOR .LE. 6) THEN
GOTO 50
ELSE GOTO 100
ENDIF
ELSE GOTO 100
ENDIF

3 CALL FGTST(CLOCA)
IF (CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') THEN
& OR (CLOCA(1:1) .EQ. '2' .OR. CLOCA(1:1) .EQ. '-') THEN
READ(CLOCA,*) LOCA(1), LOCA(2), LOCA(3)
GOTO 50
ELSE GOTO 100
ENDIF

4 CALL SPHERE(RS,LOCA,COLOR,BOOL)
GOTO 200

5 CALL FGTST(CNU)
IF (CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') THEN
READ(CNU,*) INU
IF (INU .LE. 20) INU = INU
ELSE GOTO 100
ENDIF

6 CALL FGTST(CNM)
IF (CNM(1:1) .GE. '0' .AND. CNM(1:1) .LE. '9') THEN
READ(CNM,*) INM
IF (INM .LE. 20) INM = INM
ELSE GOTO 100
ENDIF

7 CALL FGTST(CRS)
IF (CRS(1:1) .GE. '0' .AND. CRS(1:1) .LE. '9') THEN
READ(CRS,*) RS
GOTO 50

ELSE
    GOTO 100
ENDIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200    CALL GPPKM0(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS
    PAREA(1)=0.85*CSIZE(1)
    PAREA(2)=0.995*CSIZE(2)
    PAREA(3)=0.15*CSIZE(2)
    PAREA(4)=0.94*CSIZE(2)
    PAREA(5)=0.0
    PAREA(6)=CSIZE(3)
    CALL GPINPK(1,1.0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
    CALL GPPKM0(1,1,3,2)

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
    CALL GPDRV(1,5,7)

*ASSOCIATE MENU TO VIEW 5
    CALL GPARV(1,5,5,0)

RETURN
END

SUBROUTINE MTUNNEL

******************************************************************************
** SUBROUTINE MTUNNEL(R1,R2,THK,LOCA,ORI,COLOR,BOOL)  
** PROGRAM DESCRIPTION 
** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING 
** A TUNNEL IN ITS CORRECT LOCATION AND ORIENTATION 
** 
** BY:   ASHIT R. GANDHI 
** DATE:  11/12/88 
** 
** PARAMETERS USED: 
** R1  = INNER RADIUS OF THE TUNNEL 
** R2  = OUTER RADIUS OF THE TUNNEL 
** THK = THICKNESS OF THE TUNNEL 
** LOCA = LOCATION OF THE TUNNEL 
** ORI = ORIENTATION OF THE TUNNEL 
** COLOR = COLOR TO BE GIVEN TO THE TUNNEL 
** BOOL = BOOLEAN OPERATOR FOR THE TUNNEL 
** 
******************************************************************************

SUBROUTINE MTUNNEL(R1,R2,THK,LOCA,ORI,COLOR,BOOL)

REAL*4 LOCA(3), POS(2), ORI(3)
REAL*4 ASIZE(6), CSIZE(5), DATA(12)
REAL*4 PAREA(6), SAREA(6), VAREA(6)
REAL*4 R1, R2
INTEGER*4 COLOR, PPATH(3)

CHARACTER STRG*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA4*7
CHARACTER COR1*7, COR2*7, COR3*7, COR4*7
CHARACTER BOOL*1, CCOLOR*2
CHARACTER CR1*7, CR2*7, CTHK*7, CNU*7, CNW*7
CHARACTER COMMON/PATCH/NJ,NW

*DEFINE GENERIC MODEL
    R1 = 1.0
    R2 = 2.0
    THK = 1.0
    NU = 6
    NW = 6
    LOCA(1) = 0.0
    LOCA(2) = 0.0
    LOCA(3) = 0.0
    ORI(1) = 0.0
    ORI(2) = 0.0
    ORI(3) = 0.0
    BOOL = 'x'
    COLOR = 2

Appendix B - FeatureMod Program Listing 284
*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE*

CALL GPQADS(1,ERRIND,UNITS,CSIZE,ASIZE)

*****************************************************************************
* PREPARE FOR PICK INPUT
*****************************************************************************

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON*

CALL GPPKMO(1,1,1,2)

*DEFINE PICK AREAS AND INITIALIZE PICKS*

PAREA(1)=0.005*CSIZE(1)
PAREA(2)=0.045*CSIZE(1)
PAREA(3)=0.005*CSIZE(2)
PAREA(4)=0.145*CSIZE(2)
PAREA(5)=0.0
PAREA(6)=CSIZE(3)
CALL GPINKP(1,1,0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON*

CALL GPPKMO(1,1,3,2)

*OPEN STRUCTURE*

50 CALL GPESTI7)
CALL GPPOSFI7)
CALL GPADCN(1,2)

*SET TEXT COLOR TO YELLOW*

CALL GPTXCI(5)

*SET ANNOTATION TEXT SIZE SCALE FACTOR*

CALL GPAHSC(0.80)

*CONVERT NUMBERS TO TEXT*

WRITE(CR1,'(F7.2)')R1
WRITE(CR2,'(F7.2)')R2
WRITE(CTHK,'(F7.2)')THK
WRITE(CN1,'(12)')NU
WRITE(CN2,'(12)')NM
WRITE(CCOLOR,'(12)')COLOR
WRITE(CLOCAL1,'(F7.2)')CLOCA1
WRITE(CLOCAL2,'(F7.2)')CLOCA2
WRITE(CLOCAL3,'(F7.2)')CLOCA3
WRITE(CORI1,'(F7.2)')ORI(1)
WRITE(CORI2,'(F7.2)')ORI(2)
WRITE(CORI3,'(F7.2)')ORI(3)

*DRAH ALL TITLE TEXT*

POS(1) = 0.62
POS(2) = 0.80
STRG = 'BUSH = '//BOOL
CALL GPKPID(1)
CALL GPANZ(POS,9,STRG)

POS(1) = 0.62
POS(2) = 0.67
STRG = 'COLOR = '//CCOLOR
CALL GPKPID(2)
CALL GPANZ(POS,29,STRG)

POS(1) = 0.62
POS(2) = 0.64
STRG = 'LOCATION = '//CLOCAL1//CLOCAL2//CLOCAL3
CALL GPKPID(3)
CALL GPANZ(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.41
STRG = 'ORIENT = '//CORI1//CORI2//CORI3
CALL GPKPID(4)
CALL GPANZ(POS,52,STRG)

POS(1) = 0.62
POS(2) = 0.28
STRG = 'EXECUTE'
CALL GPKPID(5)
CALL GPANZ(POS,7,STRG)

POS(1) = 0.62

Appendix B - FeatureMod Program Listing
POS(2) = 0.15
STRG = 'ABORT'
CALL GPKRID(6)
CALL GPAN2(POS,5,STRG)
POS(1) = 0.32
POS(2) = 0.60
STRG = 'NH = '//CN
CALL GPKRID(8)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.80
STRG = 'R2 = '//CR2
CALL GPKRID(10)
CALL GPAN2(POS,16,STRG)
POS(1) = 0.02
POS(2) = 0.54
STRG = 'THICK = '//CTH
CALL GPKRID(11)
CALL GPAN2(POS,16,STRG)

CALL GPCLST
*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDRV1(1,5,5)
*ASSOCIATE MENU TO VIEW 5
CALL GPARV(1,5,7,0)
*UPDATE WORKSTATION
CALL FPRHP(14)
ICLASS = 2
CALL GPKRIF(1,1,1,ICLASS,1,1)
*WAIT FOR A PICK
100 CALL GPANEV(1,1,ICLASS,IDEV)
   IF (ICLASS .NE. 5)GOTO 100
   CALL GPSTPK(1,1,PPATH)
   IPKID = PPATI(2)
   GOTO (1,2,3,4,5,6,7,8,9,10,11)IPKID
1   CALL FGSTI(BOOL)
      IF (BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*' )THEN
         GOTO 100
      ENDIF
      GOTO 50

2   CALL FGSTI(CCOLOR)
      IF (CCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9')THEN
         READ(CCOLOR,*)COLOR
         IF ICOLOR .GE. 1 .AND. COLOR .LE. 6 THEN
            GOTO 50
         ELSE
            GOTO 100
         ENDIF
      ELSE
         GOTO 100
      ENDIF

3   CALL FGSTI(CLOCA)
      IF (CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9')
         OR (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-' )THEN
         READ(CLOCA,*)LOCAL(1,LOCAL2,LOCAL3)
         GOTO 50
      ELSE
         GOTO 100
      ENDIF
4 CALL FGTST(CORI)
   IF (CORI(1:1).GE. '0' .AND. CORI(1:1).LE. '9')
      & OR. (CORI(1:1).EQ. '+' OR. CORI(1:1).EQ. ' - ')
      THEN
         READ(CORI,*),CORI(1),CORI(2),CORI(3)
         GOTO 50
      ELSE
         GOTO 100
      ENDIF

5 ICHK = 1
   CALL FGRP4(R1,R2,THK,LOCA,ORI,COLOR,BOOL,ICHK,0.)
   GOTO 200

7 CALL FGTST(CNU)
   IF (CNU(1:1).GE. '0' .AND. CNU(1:1).LE. '9')THEN
      READ(CNU,*),CNU(1),CNU(2)
      IF(CNU.GT.1 .AND. CNU.LT.20)CNU = CNU
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

8 CALL FGTST(CNH)
   IF (CNH(1:1).GE. '0' .AND. CNH(1:1).LE. '9')THEN
      READ(CNH,*),CNH(1),CNH(2)
      IF(CNH.GT.1 .AND. CNH.LT.20)CNH = CNH
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

9 CALL FGTST(CRI)
   IF (CRI(1:1).GE. '0' .AND. CRI(1:1).LE. '9')THEN
      READ(CRI,*),CRI(1),CRI(2)
      GOTO 50
   ELSE
      GOTO 100
   ENDIF

10 CALL FGTST(CR2)
    IF (CR2(1:1).GE. '0' .AND. CR2(1:1).LE. '9')THEN
       READ(CR2,*),CR2(1),CR2(2)
       GOTO 50
    ELSE
       GOTO 100
    ENDIF

11 CALL FGTST(CTHK)
    IF (CTHK(1:1).GE. '0' .AND. CTHK(1:1).LE. '9')THEN
       READ(CTHK,*),CTHK(1),CTHK(2)
       GOTO 50
    ELSE
       GOTO 100
    ENDIF

*PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON
200 CALL GPPKMO(1,1,1,2)
*DEFINE PICK AREAS AND INITIALIZE PICKS
   PAREA(1)=0.85*CSIZE(1)
   PAREA(2)=0.95*CSIZE(1)
   PAREA(3)=0.15*CSIZE(2)
   PAREA(4)=0.0
   PAREA(5)=0.0
   CALL GPINPK(1,1,0,PPATH,1,PAREA,0,DATA,1)

*PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON
   CALL GPPKMO(1,1,3,2)
*DISSOCIATE SCROLL MESSSAGES FROM VIEW 5
   CALL GPDRV(1,5,7)
*ASSOCIATE MENU TO VIEW 5
   CALL GPARV(1,5,5,0)
RETURN

END

SUBROUTINE MWEDGE
** PROGRAM DESCRIPTION **

** THIS ROUTINE WILL DISPLAY MENU FOR GETTING PARAMETERS DEFINING A WEDGE IN ITS CORRECT LOCATION AND ORIENTATION **

** BY: ASHIT R. GANDHI **

** DATE: 11/12/88 **

** PARAMETERS USED: **

** A = LENGTH OF THE WEDGE **

** B = BREADTH OF THE WEDGE **

** C = HEIGHT OF THE WEDGE **

** LOCA = LOCATION OF THE WEDGE **

** ORI = ORIENTATION OF THE WEDGE **

** COLOR = COLOR TO BE GIVEN TO THE WEDGE **

** BOOL = BOOLEAN OPERATOR FOR THE WEDGE **

********************************************************************************

** SUBROUTINE WEDGE(A,B,C,LOCA,ORI,COLOR,BOOL) **

REAL*4 LOCA(3), POS(2), ORI(3), A, B, C
REAL*4 ASIZE(6), CSIZE(3), DATA(12)
REAL*4 PAREA(6), SAREA(6), VAREA(6)
INTEGER*4 COLOR, PPATH, ISS
CHARACTER STRG*32, CLOCA1*7, CLOCA2*7, CLOCA3*7, CLOCA*26
CHARACTER CORI1*7, CORI2*7, CORI3*7, CORI*26
CHARACTER BOOL*1, CCOLOR*2
CHARACTER CA*7, 08*7, CC*7, CNU*7, CNU+7
CGIKN/PATCH/NJ>M|

*DEFINE GENERIC MODEL *

A = 1.0
B = 1.0
C = 1.0
NM = 2
NW = 2
LOCA(1) = 0.0
LOCA(2) = 0.0
LOCA(3) = 0.0
ORI(1) = 0.0
ORI(2) = 0.0
ORI(3) = 0.0
BOOL = '+'
COLOR = 2

*INQUIRE ACTUAL MAXIMUM DISPLAY SURFACE SIZE *

CALL GPQADS(1, ERRIND, UNITS, CSIZE, ASIZE)

******************************************************************************

** PREPARE FOR PICK INPUT **

******************************************************************************

** PLACE PICK IN THE REQUEST MODE AND TURN ECHO SWITCH ON **

CALL GPPKMO(1,1,1,2)

** DEFINE PICK AREAS AND INITIALIZE PICKS **

PAREA(1) = 0.005*CSIZE(1)
PAREA(2) = 0.845*CSIZE(1)
PAREA(3) = 0.005*CSIZE(2)
PAREA(4) = 0.145*CSIZE(2)
PAREA(5) = 0.0
PAREA(6) = CSIZE(3)

CALL GPINPIU(1,1, PPATH, 1, PAREA, 0, DATA, 1)

** PLACE PICK IN THE EVENT MODE AND TURN ECHO SWITCH ON **

CALL GPPKMO(1, 1, 3, 2)

** OPEN STRUCTURE **

50 CALL GPES(7)
CALL GPOPST(7)
CALL GPADCN(1,2)

** SET TEXT COLOR TO YELLOW **

CALL GPTXCI(5)

** SET ANNOTATION TEXT SIZE SCALE FACTOR **

CALL GPAHSC(0.80)

** DRAW ALL TITLE TEXT **

POS(1) = 0.62
POS(2) = 0.80

STRG = 'HEDGE = '//BOOL
CALL GPPKID(1)
CALL GPAN2(POS,9,STRG)

WRITE('A',*(F7.2))A
WRITE('B',*(F7.2))B
WRITE('C',*(F7.2))C
WRITE('D',*(F7.2))D
WRITE('E',*(F7.2))E
WRITE('F',*(F7.2))F
WRITE('G',*(F7.2))G
WRITE('H',*(F7.2))H
WRITE('I',*(F7.2))I
WRITE('J',*(F7.2))J
WRITE('K',*(F7.2))K
WRITE('L',*(F7.2))L
WRITE('M',*(F7.2))M
WRITE('N',*(F7.2))N
WRITE('O',*(F7.2))O
WRITE('P',*(F7.2))P
WRITE('Q',*(F7.2))Q
WRITE('R',*(F7.2))R
WRITE('S',*(F7.2))S
WRITE('T',*(F7.2))T
WRITE('U',*(F7.2))U
WRITE('V',*(F7.2))V
WRITE('W',*(F7.2))W
WRITE('X',*(F7.2))X
WRITE('Y',*(F7.2))Y
WRITE('Z',*(F7.2))Z
WRITE('0',*(F7.2))0
WRITE('1',*(F7.2))1
WRITE('2',*(F7.2))2
WRITE('3',*(F7.2))3
WRITE('4',*(F7.2))4
WRITE('5',*(F7.2))5
WRITE('6',*(F7.2))6
WRITE('7',*(F7.2))7
WRITE('8',*(F7.2))8
WRITE('9',*(F7.2))9

POS(1) = 0.62
POS(2) = 0.67

STRG = 'COLOR = '//CCOLOR
CALL GPPKID(2)
CALL GPAN2(POS,29,STRG)

POS(1) = 0.62
POS(2) = 0.54

STRG = 'LOCATION = '//CLOCA1//CLOCA2//CLOCA3
CALL GPPKID(3)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.41

STRG = 'ORIENT = '//CORI1//CORI2//CORI3
CALL GPPKID(4)
CALL GPAN2(POS,32,STRG)

POS(1) = 0.62
POS(2) = 0.26

STRG = 'EXECUTE'
CALL GPPKID(5)
CALL GPAN2(POS,7,STRG)

POS(1) = 0.62
POS(2) = 0.15

STRG = 'ABORT'
CALL GPPKID(6)
CALL GPAN2(POS,6,STRG)

POS(1) = 0.32
POS(2) = 0.80

STRG = 'NU = '//CW
CALL GPPKID(7)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.32
POS(2) = 0.67

STRG = 'NN = '//CW
CALL GPPKID(8)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.02
POS(2) = 0.80

STRG = 'LENGTH = '//CA
CALL GPPKID(9)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.02
POS(2) = 0.67

STRG = 'BREATH = '//CB
CALL GPPKID(10)
CALL GPAN2(POS,16,STRG)

POS(1) = 0.02
POS(2) = 0.54

STRG = 'HEIGHT = '//CC
CALL GPPKID(11)
CALL GPAN2(POS,16,STRG)

CALL GPCLST

*DISSOCIATE SCROLL MESSAGES FROM VIEW 5
CALL GPDV(1,5,5)

*ASSOCIATE MENU TO VIEW 5

Appendix B - FeatureMod Program Listing
CALL GPARV(1,5,7,0)
*UPDATE WORKSTATION
CALL FPRMP(I4)
ICLASS = 2
CALL GPVPPF(1,1,1,ICLASS,1,1)
*WAIT FOR A PICK
100 CALL GPAHEV(1,1,1,ICLASS,IDEV)
   IF (ICLASS .NE. 5) GOTO 100
   CALL GPVTPK(1,1,PPATH)
   IPKID = PPATH(2)
   GOTO (1,2,3,4,5,6,7,8,9,10,11)IPKID
1 CALL FGSTST(BOOL)
   IF [BOOL .NE. '+' .AND. BOOL .NE. '-' .AND. BOOL .NE. '*') THEN
      GOTO 100
   ENDIF
   GOTO 50
2 CALL FGSTST(CCOLOR)
   IF ((CCCOLOR(1:1) .GE. '0' .AND. CCOLOR(1:1) .LE. '9') .OR. (CCCOLOR(1:1) .EQ. '+' .OR. CCOLOR(1:1) .EQ. '-')) THEN
      READ(CCOLOR;*) COLOR
      IF (COLOR .GT. 1 .AND. COLOR .LT. 6) THEN
         GOTO 50
      ELSE
         GOTO 100
      ENDIF
   ELSE
      GOTO 100
   ENDIF
3 CALL FGSTST(CLOCA)
   IF ((CLOCA(1:1) .GE. '0' .AND. CLOCA(1:1) .LE. '9') .OR. (CLOCA(1:1) .EQ. '+' .OR. CLOCA(1:1) .EQ. '-')) THEN
      READ(CLOCA;*) LOCA(1:3)
      GOTO 50
   ELSE
      GOTO 100
   ENDIF
4 CALL FGSTST(CORI)
   IF ((CORI(1:1) .GE. '0' .AND. CORI(1:1) .LE. '9') .OR. (CORI(1:1) .EQ. '+' .OR. CORI(1:1) .EQ. '-')) THEN
      READ(CORI;*) ORI(1:3)
      GOTO 50
   ELSE
      GOTO 100
   ENDIF
5 CALL NGDEGE(A,B,C,LOCA,ORI,COLOR,BOOL)
6 GOTO 200
7 CALL FGSTST(CNU)
   IF ((CNU(1:1) .GE. '0' .AND. CNU(1:1) .LE. '9') .OR. (CNU(1:1) .EQ. '+' .OR. CNU(1:1) .EQ. '-')) THEN
      READ(CNU;*) INU
      IF (INU .GT. 1 .AND. INU .LT. 20) INU = NU
      GOTO 50
   ELSE
      GOTO 100
   ENDIF
8 CALL FGSTST(CNM)
   IF ((CNM(1:1) .GE. '0' .AND. CNM(1:1) .LE. '9') .OR. (CNM(1:1) .EQ. '+' .OR. CNM(1:1) .EQ. '-')) THEN
      READ(CNM;*) INM
      IF (INM .GT. 1 .AND. INM .LT. 20) INM = NM
      GOTO 50
   ELSE
      GOTO 100
   ENDIF
9 CALL FGSTST(CA)
   IF ((CA(1:1) .GE. '0' .AND. CA(1:1) .LE. '9') .OR. (CA(1:1) .EQ. '+' .OR. CA(1:1) .EQ. '-')) THEN
      READ(CA;*) A
      GOTO 50
   ELSE
      GOTO 100
   ENDIF
10 CALL FGSTST(CB)
    IF ((CB(1:1) .GE. '0' .AND. CB(1:1) .LE. '9') .OR. (CB(1:1) .EQ. '+' .OR. CB(1:1) .EQ. '-')) THEN
      READ(CB;*) B
      GOTO 50
    ELSE
      GOTO 100
    ENDIF
11 CALL FGSTST(CC)

Appendix B - FeatureMod Program Listing
SUBROUTINE PYRAMID

PROGRAM DESCRIPTION

THIS ROUTINE WILL COMPUTE THE CONTROL POINTS REQUIRED FOR
CREATING A PYRAMID USING B-SPLINE SURFACES

BY: ASHIT R. GANDHI

DATE: 11/12/88

PARAMETERS USED:

L = LENGTH OF THE PYRAMID IN THE X-DIRECTION
B = LENGTH OF THE PYRAMID IN THE Y-DIRECTION
H = LENGTH OF THE PYRAMID IN THE Z-DIRECTION
AR = ASPECT RATIO OF THE PYRAMID
LOCA = LOCATION OF THE PYRAMID
ORI = ORIENTATION OF THE PYRAMID
COLOR = COLOR TO BE GIVEN TO THE PYRAMID
BOOL = BOOLEAN OPERATOR FOR THE PYRAMID

INTEGER AXNUM, OBNUM, INNUM, LTYPE, CNUM
REAL F2(2), F4(4), X(4), Y(4), Z(4), PTS(20, 20, 3)
REAL SPTS(6,4,3), U(4), W(4), LOCAL(3), ORI(3)
REAL PLOC(3), PORI(3)

CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8

COMMON/PATCH/NM, NM
COMMON/IDS/AXNUM, OBNUM, INNUM
COMMON/CMP/ASSEM, OBJECT
COMMON/REndo/PLOC, PORI

DATA F2/1.0,-1.0/
DATA F4/-1.0,0.0,1.0,2.0/
DATA HO/400,1.0/
DATA U/0.0, 0.3333333, 0.66666667, 1.0/
DATA W/0.0, 0.3333333, 0.66666667, 1.0/

DO 50 I = 1,3
  PLOC(I) = LOC(I)
50  
CONTINUE
IF (BOOL .EQ. '2') LTYPE = 1
IF (BOOL .EQ. '3') LTYPE = 2
IF (BOOL .EQ. '4') LTYPE = 3

INNUM = INNUM + 1
IF (FIL .EQ. 1) THEN
  NSURF = 4
ELSE IF (AR .EQ. 0.0) THEN
  NSURF = 5
ELSE
  NSURF = 6
ENDIF
ENDIF
CALL FDINS(INNUM,LTYPE,'PYRAMID ',NSURF,CNUM,
  OBJECTICMJM-10001,LOC,ORI)

CREATE THE PATCHES FOR Z = CONSTANT

X(1) = 0.0
X(2) = L/3.0
X(3) = 2.0*L/3.0
X(4) = L

ZO = H
DZO = -H/3.0
ZL = H + AR)/2.0
DZL = H*AR/3.0

Y0 = 0.0
DYO = B/3.0
YL = B(1.0 - AR)/2.0
DYL = B*AR/3.0

WRITE(10,100)INNUM,'4 4 0 0',ICOL, NU, NM
DO 200 I = 1,4
  DO 300 K = 1,4
    Z(K) = ZO + (ZL-ZO)*(K-1)/3.0
    X(K) = X(I) + (X(4)-X(I))*(K-1)/3.0
    Y(K) = Y(I) + (Y(4)-Y(I))*(K-1)/3.0
    PTS(1) = X(K)
    PTS(2) = Y(K)
    PTS(3) = Z(K)
    SPTS(I,K,1) = PTS(1)
    SPTS(I,K,2) = PTS(2)
    SPTS(I,K,3) = PTS(3)

100 CONTINUE
Y0 = Y0 + DYO
YL = YL + DYL

200 CONTINUE
DO 250 I = 1,4
  DO 350 K = 1,4
    Z(K) = ZO + (ZL-ZO)*(K-1)/3.0
    X(K) = X(I) + (X(4)-X(I))*(K-1)/3.0
    Y(K) = Y(I) + (Y(4)-Y(I))*(K-1)/3.0
    PTS(1) = X(K)
    PTS(2) = Y(K)
    PTS(3) = Z(K)
    SPTS(I,K,1) = PTS(1)
    SPTS(I,K,2) = PTS(2)
    SPTS(I,K,3) = PTS(3)

250 CONTINUE
Y0 = Y0 + DYO
YL = YL + DYL

DO 400 I = 1,4
  DO 500 K = 1,4
    Z(K) = ZO + (ZL-ZO)*(K-1)/3.0
    X(K) = X(I) + (X(4)-X(I))*(K-1)/3.0
    Y(K) = Y(I) + (Y(4)-Y(I))*(K-1)/3.0
    PTS(1) = X(K)
    PTS(2) = Y(K)
    PTS(3) = Z(K)
    SPTS(I,K,1) = PTS(1)
    SPTS(I,K,2) = PTS(2)
    SPTS(I,K,3) = PTS(3)

300 CONTINUE
Y0 = Y0 + DYO
YL = YL + DYL

400 CONTINUE
CALL FSICP(U,M,SPTS,PNTS)
DO 450 I = 1,4
  WRITE(10,*)PNTS(I,KL,1),KL=1,4
  WRITE(10,*)PNTS(I,KL,2),KL=1,4
  WRITE(10,*)PNTS(I,KL,3),KL=1,4

450 CONTINUE
CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
DO 500 K = 1,4
  Y(K) = Y0 + (YL-Y0)*(K-1)/3.0
  Z(K) = Z0 + (ZL-Z0)*(K-1)/3.0
  PTS(1) = X(K)
  PTS(2) = Y(K)
  PTS(3) = Z(K)
  SPTS(I,K,1) = PTS(1)
  SPTS(I,K,2) = PTS(2)
  SPTS(I,K,3) = PTS(3)
500 CONTINUE
Z0 = Z0 + DZ0
ZL = ZL + DZL
600 CONTINUE
CALL FSTCP(U,N,$PTS,PNTS)
DO 850 I = 1,4
  WRITE(10,'10)INNM,'4 4 0 0',ICOL, NU, NM
  WRITE(10,*)(*PNTS(I,KL,1),KL=1,4)
  WRITE(10,*)(*PNTS(I,KL,2),KL=1,4)
  WRITE(10,*)(*PNTS(I,KL,3) KL=1,4)
  CALL FBPCH4,4,0,0,PNTS,H0,ICOL,LTYP5)
  Z0 = Z0 + DZ0
  ZL = ZL + DZL
800 CONTINUE
CALL FSTCP(U,N,$PTS,PNTS)
DO 1000 I = 1,4
  WRITE(10,'10)INNM,'4 4 0 0',ICOL, NU, NM
  WRITE(10,*)(*PNTS(I,KL,1),KL=1,4)
  WRITE(10,*)(*PNTS(I,KL,2),KL=1,4)
  WRITE(10,*)(*PNTS(I,KL,3) KL=1,4)
  CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYP5)
  IF (IFL .LT. 1)RETURN
*CREATE THE PATCHES FOR X = CONSTANT
Y(1) = -B
Y(2) = B
Y(3) = B
Y(4) = 2.0*B
WRITE(10,'10)INNM,'4 4 0 0',ICOL, NU, NM
DO 1000 K = 1,4
  Z(K) = F4(I)*H
  PTS(1) = X(K)
  PTS(2) = Y(K)
  PTS(3) = Z(K)
  PNTS(I,K,1) = PTS(1)
  PNTS(I,K,2) = PTS(2)
  PNTS(I,K,3) = PTS(3)
1000 CONTINUE
*WRITE(10,*)(*PNTS(I,KL,1),KL=1,4)
*WRITE(10,*)(*PNTS(I,KL,2),KL=1,4)
*WRITE(10,*)(*PNTS(I,KL,3) KL=1,4)
1000 CONTINUE
CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYP5)
IF (AR .LT. 0.0)RETURN
Y(1) = -B*AR
SUBROUTINE RING

*******************************************************************************
** SUBROUTINE RING(R1,R2,LOC,ORI,ICOL,BOOL,ICHK,PLACE)                **
*******************************************************************************
** PROGRAM DESCRIPTION: **
** THIS ROUTINE WILL COMPUTE THE CONTROL POINTS REQUIRED TO **
** CREATE A RING USING B-SPLINE SURFACES **
** BY: ASHIT R. GANDHI **
** DATE: 11/21/88 **
** PARAMETERS USED: **
** R1 = INNER RADIUS OF RING (REAL,I/P) **
** R2 = OUTER RADIUS OF RING (REAL,I/P) **
** LOC = LOCATION OF THE RING (REAL,I/P) **
** ORI = EULER ORIENTATION FOR THE RING (REAL,I/P) **
** ICOL = COLOR FOR FEATURE **
** BOOL = BOOLEAN TYPE FOR FEATURE **
** ICHK = FLAG FOR NEW FEATURE **
** PLACE= LOCATION OF FEATURE IN LOCAL AXIS **
*******************************************************************************

SUBROUTINE RING(R1,R2,LOC,ORI,ICOL,BOOL,ICHK,PLACE)

INTEGER*4 ASJNH; OBMM, INMM, LTYPE; CMM
REAL*4 R1, R2, LOC(3), ORI(3), FZ(4), FY(4), FX(4)
REAL*4 U(4), N(4), H0(20, 20), PT(3), PNTS(20, 20, 3)
REAL*4 PLOC(3), PORI(3)
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8
COMMON/PATCH/NJ,NM
COMMON/IDS/ASJNH,OBNUM,INNUM
COMMON/COMP/ASSEM,OBJECT
COMMON/PRTOF/CNUM
COMMON/REND/PLOC,PORI

DATA FX/-1.0, 0.0, 1.0, 2.0/
DATA FY/-1.0, 0.0, 1.0, 2.0/
DATA FU/ 1.0, 1.0, 1.0, 1.0/
DATA HU/ 0.0, 0.33333333, 0.66666667, 1.0/
DATA HU/ 0.0, 0.33333333, 0.66666667, 1.0/
DATA HG/ 0.0, 1.0/

DO 50 I = 1,3
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)
50 CONTINUE

*MAKE SURE THAT OUTER RADIUS IS GREATER THAN INNER RADIUS
IF (R1 .GT. R2) THEN

Appendix B - FeatureMod Program Listing 294
WRITE(6,*)'RING ====> INNER RADIUS IS GREATER THAN'
WRITE(6,*)'RING ====> OUTER RADIUS
BOOL = .NOT.
R1 = R2
R2 = R3
ENDIF
IF (BOOL .EQ. '='+)LTYPE = 1
IF (BOOL .EQ. '=')LTYPE = 2
IF (BOOL .EQ. '-')LTYPE = 3
IF (ICHK .EQ. 0) THEN
CALL FDINS(INNUM,LTYPE,'RING ',4,CNUM,
   OBJECT(CNUM-1000),LOC,ORI)
ENDIF
PI = 3.14159
DEL = PI/6.0
pered the ring
DO 600 K = 1,4
WRITE(10,10)INNUM,'4 4 0 0',ICOL,NU,NM
THETA = 0.0
DO 500 I = 1,4
DO 400 J = 1,4
PT(1) = 0.0 + PLACE
PT(2) = FY(K)*U(I)*R1 + U(J)*(R2-R1))xSIN(THETA)
PT(3) = FZ(K)*U(I)*R1 + U(J)*(R2-R1))xCOS(THETA)
PTS(I,J,1) = PT(1)
PTS(I,J,2) = PT(2)
PTS(I,J,3) = PT(3)
400 CONTINUE
THETA = THETA + DEL
500 CONTINUE
CALL FSTCP(U,N,PTS,PNTS)
DO 550 I = 1,4
WRITE(10,10)PNTS(I,J,1),J=1,4)
WRITE(10,10)PNTS(I,J,2),J=1,4)
WRITE(10,10)PNTS(I,J,3),J=1,4)
550 CONTINUE
CALL FBPCH4,4,0,0,PNTS,H0,ICOL,LTYPE)
600 CONTINUE
RETURN
10 FORMAT(1X,I5,1X,A7,3(1X,I2))
END

SUBROUTINE SLAB

**********************************************************************
SUBROUTINE SLAB(L,B,H,LOC,ORI,COLOR,BOOL)
*************************************************************************
PARAMETER USED:
L = LENGTH OF THE SLAB
B = BREADTH OF THE SLAB
H = HEIGHT OF THE SLAB
LOC = LOCATION OF THE SLAB
ORI = ORIENTATION OF THE SLAB
COLOR = COLOR TO BE GIVEN TO THE SLAB
BOOL = BOOLEAN OPERATOR FOR THE SLAB
*************************************************************************
INTEGER ASNUM, OBNUM, INNUM, LTYPE, CNUM
REAL L, B, H, LOC(3), ORI(3), H0(20,20)
REAL F2(2), F4(4), X(4), Y(4), Z(4), PTS(3), PNTS(20,20,3)
REAL PLOC(5), P0(3)
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(900)*8
COMMON/PATCH/ASSEM

Appendix B - FeatureMod Program Listing 295
COMMON/IDS/ASNUM,OBNUM,INNUM
COMMON/CMP/ASSEM,OBJECT
COMMON/PRT/PFN,LOC,POR
DATA F2/-0.0,1.0/
DATA F4/-1.0,0.0,1.0,2.0/
DATA H0/400*-1.0/
INNUM = INNUM + 1
IF (BOOL .LT. '+') LTYPE = 1
IF (BOOL .LT. '-' I) LTYPE = 2
IF (BOOL .LT. '*') LTYPE = 3
DO 50 I = 1,3
   LOC(I) = LOC(I)
   POR(I) = ORI(I)
50 CONTINUE
CALL FDINS(INNUM,LTYPE,'SLAB',CMM,
            OBJECT,CNUM,1,LOC,ORI)
*CREATE THE PATCHES FOR Z = CONSTANT
X(1) = -L
X(2) = 0.0
X(3) = L
X(4) = 2.0*M
DO 500 J = 1,2
   WRITE(10,10)INNUM,'4 4 0 0',ICOL,NJ,NM
   DO 400 I = 1,4
      400 Y(I) = F4(I)*M
      Z(I) = F2(I)*H
      PTS(1) = X(I)
      PTS(2) = Y(I)
      PTS(3) = Z(I)
      PNTS(I,K,1) = PTS(1)
      PNTS(I,K,2) = PTS(2)
      PNTS(I,K,3) = PTS(3)
   400 CONTINUE
   WRITE(10,*)(PNTS(I,K),K=1,4)
   WRITE(10,*)(H0(I,K),K=1,4)
   CONTINUE
   CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
500 CONTINUE
*CREATE THE PATCHES FOR Y = CONSTANT
X(1) = -L
X(2) = 0.0
X(3) = L
X(4) = 2.0*M
DO 500 J = 1,2
   WRITE(10,10)INNUM,'4 4 0 0',ICOL,NJ,NM
   DO 400 I = 1,4
      400 Y(I) = F4(I)*M
      Z(I) = F2(I)*H
      PTS(1) = X(I)
      PTS(2) = Y(I)
      PTS(3) = Z(I)
      PNTS(I,K,1) = PTS(1)
      PNTS(I,K,2) = PTS(2)
      PNTS(I,K,3) = PTS(3)
   400 CONTINUE
   WRITE(10,*)(PNTS(I,K),K=1,4)
   WRITE(10,*)(H0(I,K),K=1,4)
   CONTINUE
   CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
500 CONTINUE
*CREATE THE PATCHES FOR X = CONSTANT
Y(1) = -B
Y(2) = 0.0
Y(3) = B
Y(4) = 2.0*M
DO 900 J = 1,2
   WRITE(10,10)INNUM,'4 4 0 0',ICOL,NJ,NM
   DO 800 I = 1,4
      800 X(I) = F2(I)*M
      Z(I) = F4(I)*H
      PTS(1) = X(I)
      PTS(2) = Y(I)
      PTS(3) = Z(I)
      PNTS(I,K,1) = PTS(1)
   800 CONTINUE
   WRITE(10,*)(PNTS(I,K),K=1,4)
   WRITE(10,*)(H0(I,K),K=1,4)
   CONTINUE
   CALL FBPCH(4,4,0,0,PNTS,H0,ICOL,LTYPE)
900 CONTINUE

Appendix B - FeatureMod Program Listing 296
SUBROUTINE SPHERE

***************
** SUBROUTINE SPHERE(RAD,LOC,ICOL,BOOL) 
**
** PROGRAM DESCRIPTION 
**
** THIS ROUTINE COMPUTES THE CONTROL POINTS THAT WOULD BE 
** REQUIRED TO CREATE A SPHERE USING B-SPLINE SURFACES 
**
** BY: ASHIT R. GANDHI 
** DATE: 11/21/88 
**
** PARAMETERS USED: 
** RAD = RADIUS OF THE SPHERE (REAL I/P) 
** LOC = LOCATION OF THE SPHERE (REAL I/P) 
** ICOL = COLOR FOR SPHERE (INTEGER I/P) 
** BOOL = BOOLEAN TYPE FOR SPHERE (CHARACTER I/P) 
**

INTEGER*4 ASNJM, OBNUM, INNJM, LTYPE, CMM 
REAL*4 RAD, LOC(3), PT(3), PTS(4,4,3), ORI(3), LOCB(3) 
REAL*4 U(4), M(4), PNTS(20,20,3), H0(20,20), YS(2) 
REAL*4 PLOC(3), PORI(3) 
CHARACTER BOOL*1, ASSEM(900)*8, OBJECT(9000)*8 
COMMON/PATCH/NU, NW 
COMMON/IDS/ASNJM, OBNUM, INNJM 
COMMON/COMP/ASSEM, OBJECT 
COMMON/PRTOF/CNUM 
COMMON/REND/PLOC, PORI 
DATA U/0.0,.33333333,.66666667,1.0/ 
DATA M/0.0,.33333333,.66666667,1.0/ 
DATA H/0.0,.33333333,.66666667,1.0/ 
DATA YS/1.0,-1.0,0.0/ 
DATA LOCB/3*0.0/ 
PI = 3.14159 
INNUM = INNUM + 1 
ORI(1) = 0.0 
ORI(2) = 0.0 
ORI(3) = 0.0 
DO 50 I = 1,3 
PLOC(I) = LOC(I) 
PORI(I) = ORI(I) 
50 CONTINUE 
CALL FDINS(INNUM, LTYPE, 'SPHERE ', 8, CNUM, 
OBJECT(INNUM-1000), LOC, ORI) 
IF (BOOL .EQ. '+')LTYPE = 1 
IF (BOOL .EQ. '*')LTYPE = 2 
IF (BOOL .EQ. '-')LTYPE = 3 
ORI(1) = 0.0 
ORI(2) = 0.0 
DELTA = PI/6.0 
DO 600 IK = 1,2 
THETA = 0.0 
ORI(I) = 0.0 
DO 200 I = 1,6 
ALPHA = 0.0 
200 CONTINUE
DO 100 J = 1,4
  PT(1) = RAD*COS(THETA)*COS(ALPHA)
  PT(2) = YSL(I,K)*RAD*SIN(THETA)
  PT(3) = RAD*COS(THETA)*SIN(ALPHA)
  CALL FTRNSILOCB ORI,PT
  PTS(I,J,1) = PT(1)
  PTS(I,J,2) = PT(2)
  PTS(I,J,3) = PT(3)
  ALPHA = ALPHA + DELTA
100 CONTINUE
THETA = THETA + DELTA
200 CONTINUE
CALL FSTCP(U,N,PTS,PNTS)
CALL FBPCAN(4,9,4,0,PNTS,H0,ICOL,LTYPE)
WRITE(10,10)INNUM,'4 4 0 0',ICOL,NU,NM
DO 250 I = 1,4
  WRITE(10,4)(PNTS(I,J,1),J=1,4)
  WRITE(10,4)(PNTS(I,J,2),J=1,4)
  WRITE(10,4)(PNTS(I,J,3),J=1,4)
WRITE(10,4)(H0(I,J),J=1,4)
250 CONTINUE
300 CONTINUE
CALL FSTCP(U,N,PTS,PNTS)
CALL FBPCAN(4,9,4,0,PNTS,H0,ICOL,LTYPE)
WRITE(10,10)INNUM,'4 4 0 0',ICOL,NU,NM
DO 450 I = 1,4
  WRITE(10,4)(PNTS(I,J,1),J=1,4)
  WRITE(10,4)(PNTS(I,J,2),J=1,4)
  WRITE(10,4)(PNTS(I,J,3),J=1,4)
WRITE(10,4)(H0(I,J),J=1,4)
450 CONTINUE
500 CONTINUE
CONTINUE
600 CONTINUE
RETURN
10 EGSMAT(1X,I5,1X,A7,3(1X,I2))

SUBROUTINE TUNNEL

*** SUBROUTINE TUNNEL(R1,R2,THK,LOCA,ORI,COLOR,BOOL,ICHK,PLACE) ***
*** PROGRAM DESCRIPTION ***
***\ This routine will compute the control points for \***
*** creating a tunnel using B-spline surfaces. ***
***\ BY: \ ASHIT R. GANDHI ***
*** DATE: 11/12/88 ***
***\ PARAMETERS USED: ***
***\ R1 = inner radius of the tunnel ***
*** R2 = outer radius of the tunnel ***
*** THK = thickness of the tunnel ***
*** LOCA = location of the tunnel ***
*** ORI = orientation of the tunnel ***
*** COLOR = color to be given to the tunnel ***
*** BOOL = boolean operator for the tunnel ***
*** ICHK = flag for new feature ***
*** PLACE = location of feature in local axis ***
*** SUBROUTINE TUNNEL(R1,R2,THK,LOCA,ORI,COLOR,BOOL,ICHK,PLACE) ***
INTEGER*4 ICOL
INTEGER*4 ASINUM, CBNUM, INNUM, LTYPE, CNUM
REAL*4 LOC(3), ORI(3), PT(3), PTS(4,4,3)
REAL*4 R1, R2, LEN
REAL*4 PLOC(3), PORI(3), PLACE
REAL*4 PNT$20;20;3); H0(20;20); FX(4); FY(4); FZ(4); U(4); N(4)
CHARACTER BOOL*1; A$SEM(900); OBJECT(9000)*8
COMMON/PATCH/NUM
COMMON/IDS/ASSEM;CBNUM,INNUM
COMMON/COMPS/ASSEM,OBJECT
COMMON/REND/PLOC,PORI
DATA FX/1.0, 0.0, 1.0, -1.0/
DATA FY/1.0, 1.0, -1.0, -1.0/
DATA FZ/1.0, -1.0, -1.0, 1.0/
DATA U/0.0, 0.33333333, 0.66666667, 1.0/
DATA N/0.0, 0.33333333, 0.66666667, 1.0/
DATA H0/400*1.0/
DO 50 I = 1,3
   PLOC(I) = LOC(I)
   PORI(I) = ORI(I)
50 CONTINUE
*MAKE SURE THAT OUTER RADIUS IS GREATER THAN INNER RADIUS
IF (R1 > R2)
   WRITE(6,**,*)'RING ===> INNER RADIUS IS GREATER THAN'
   WRITE(6,**,*)'RING ===> OUTER RADIUS'
   R3 = R1
   R1 = R2
   R2 = R3
ENDIF
IF (BOOL .EQ. '+')LTYPE = 1
IF (BOOL .EQ. '-')LTYPE = 2
IF (BOOL .EQ. '*')LTYPE = 3
INNUM = INNUM + 1
CALL FDINS(INNUM,LTYPE,'TUNNEL ',ICOL;NUM;
100 OBJECT(INUM-1000),LOC,ORI)
PI = 3.14159
DEL = PI/6.0
*COMPUTE THE HOLLOW CYLINDER
DO 400 II = 1,2
   RAD = R1
   IF (II .EQ. 2)RAD = R2
   DO 300 K = 1,2
      WRITE(10,10)INNUM,' 4 4 0 0',ICOL,NU,NH
      THETA = 0.0
      DO 200 I = 1,4
         PT(1) = U(J)*LEN + PLACE
         PT(2) = FY(K)*RAD*SIN(THETA)
         PT(3) = FZ(K)*RAD*COS(THETA)
         PTS(I,J,1) = PT(1)
         PTS(I,J,2) = PT(2)
         PTS(I,J,3) = PT(3)
      200 CONTINUE
      THETA = THETA + DEL
   300 CONTINUE
   CALL FSICP(U,N,PTS,PNT$)
   CALL FBPCCH(4,4,0,0,PNT$,H0,ICOL,LTYPE)
   DO 250 I = 1,4
      WRITE(10,**,*)('PNTS(I,J,J),J=1,4)
      WRITE(10,**,*)('H0(I,J,J),J=1,4')
   250 CONTINUE
400 CONTINUE
*COMPUTE THE RING
DO 900 II = 1,2
   PLACE1 = 0.
   IF (II .EQ. 2)PLACE1 = LEN
   DO 800 K = 1,2
      WRITE(10,10)INNUM,' 4 4 0 0',ICOL,NU,THETA = 0.0
      DO 500 J = 1,4
         PT(1) = 0.0 + PLACE1
         PT(2) = FY(K)*RAD + U(J)*(R2-R1)*SIN(THETA)
         PT(3) = FZ(K)*RAD + U(J)*(R2-R1)*COS(THETA)
         PTS(I,J,1) = PT(1)
         PTS(I,J,2) = PT(2)
         PTS(I,J,3) = PT(3)
      500 CONTINUE
900 CONTINUE
Appendix B - FeatureMod Program Listing 299
**SUBROUTINE WEDGE**

*******************************************************************************
** SUBROUTINE WEDGE(L,B,H,LOC,ORI,COLOR,BOOL) **
*******************************************************************************

** PROGRAM DESCRIPTION **

** THIS ROUTINE WILL COMPUTE THE CONTROL POINTS REQUIRED FOR **
** CREATING A WEDGE USING B- SPLINE SURFACES. **

** BY: ASHIT R. GANDHI **
** DATE: 11/12/88 **

** PARAMETERS USED: **

** L = LENGTH OF THE WEDGE **
** B = BREADTH OF THE WEDGE **
** H = HEIGHT OF THE WEDGE **
** LOC = LOCATION OF THE WEDGE **
** ORI = ORIENTATION OF THE WEDGE **
** COLOR = COLOR TO BE GIVEN TO THE WEDGE **
** BOOL = BOOLEAN OPERATOR FOR THE WEDGE **

*******************************************************************************

SUBROUTINE NEDGEIL,B>H,LOC,ORI,ICOL,BOOL)

INTEGER*4 ASNM,OBNM,INNM,LTYPE,CNUM
REAL*4 L,B,H,LOC(3),ORI(3),HO(20,20),FU(4),U(4),W(4)
REAL*4 FZ(2),F4(4),X(4),Y(4),Z(4),PTS(3),PNTS(20,20,3)
REAL*4 PLOC(3),PORI(3)
CHARACTER BOOL*1,ASSEM(900)*8,OBJECT(9000)*8
COMMON/PATCH/NU,NN
COMMON/TDS/ASNM,OBNM,INNM
COMMON/COMP/ASSEM,OBJECT
COMMON/PRTDF/CNUM
COMMON/REND/PLOC,PORI
DATA FZ/0.0,1.0/
DATA F4/-1.0,0.0,1.0,2.0/
DATA H0/400/1.0/
DATA FU/1.0,0.6666667,0.3333333,0.0/
DATA W/0.0,0.5555556,0.8333333,1.0/
DATA N/0.0,0.3333333,0.6666667,1.0/

DO 50 I = 1,3
   PO(I) = LOC(I)
   PG(I) = ORI(I)
50 CONTINUE

IF (BOOL .EQ. '+') LTYPE = 1
IF (BOOL .EQ. '-') LTYPE = 2
IF (BOOL .EQ. '*') LTYPE = 3
INNUM = INNUM + 1
CALL FDININUM LTYPE, 'HEDGE', INNUM, I

CREATE THE PATCHES FOR Z = CONSTANT
X(1) = 0.0
X(2) = L/3.0
X(3) = 2.0*L/3.0
X(4) = L

DO 300 J = 1,2
   WRITE(10,10) INNUM, '4 4 0 0', ICOL, NU, NW
   DO 200 I = 1,4
      Y(I) = F1(I)*FU(I)
      Z(I) = H
      PTS(1) = X(I)
      PTS(2) = Y(I)
      PTS(3) = Z(I)
      SPTS(I,K,1) = PTS(1)
      SPTS(I,K,2) = PTS(2)
      SPTS(I,K,3) = PTS(3)
   200 CONTINUE
   CALL FSTCP(U,W,SPTS,PNTS)

DO 250 J = 1,4
   WRITE(10,*)(PNTS(I,K,1), K=1,4)
   WRITE(10,*)(PNTS(I,K,2), K=1,4)
   WRITE(10,*)(PNTS(I,K,3), K=1,4)

250 NRITE(10) = 4
   CALL FBCHI4,4,0,0,PNTS,H0,ICOL,LTYPE
300 CONTINUE

CREATE THE PATCHES FOR Y = CONSTANT
X(1) = 0.0
X(2) = L/3.0
X(3) = 2.0*L/3.0
X(4) = L

DO 500 J = 1,2
   WRITE(10,10) INNUM, '4 4 0 0', ICOL, NU, NW
   DO 400 K = 1,4
      Y(K) = F1(J)*FU(I)
      Z(K) = U(I)*H
      PTS(1) = X(I)
      PTS(2) = Y(K)
      PTS(3) = Z(K)
      SPTS(I,K,1) = PTS(1)
      SPTS(I,K,2) = PTS(2)
      SPTS(I,K,3) = PTS(3)
   400 CONTINUE
   CALL FSTCP(U,W,SPTS,PNTS)

DO 550 J = 1,4
   WRITE(10,*)(PNTS(I,K,1), K=1,4)
   WRITE(10,*)(PNTS(I,K,2), K=1,4)
   WRITE(10,*)(PNTS(I,K,3), K=1,4)

550 NRITE(10) = 4
   CALL FBCHI4,4,0,0,PNTS,H0,ICOL,LTYPE
600 CONTINUE

CREATE THE PATCHES FOR X = CONSTANT
Y(1) = 0.0
Y(2) = B/3.0
Y(3) = 2.0*B/3.0
Y(4) = B

DO 900 J = 1,1
   WRITE(10,10) INNUM, '4 4 0 0', ICOL, NU, NW
   DO 800 K = 1,4
      X(K) = 0.0
      Z(K) = F1(I)*H
      PTS(1) = X(K)
      PTS(2) = Y(I)
      PTS(3) = Z(K)
   800 CONTINUE
   CALL FBCHI4,4,0,0,PNTS,H0,ICOL,LTYPE
900 CONTINUE
SPTS(I,K,1) = PTS(1)
SPTS(I,K,2) = PTS(2)
SPTS(I,K,3) = PTS(3)

CONTINUE

CALL FSTCP(U,W,SPTS,PNTS)
DO 850 I = 1, 4
  WRITE(10,*) (PNTS(I,KL,1), KL = 1, 4)
  WRITE(10,*) (PNTS(I,KL,2), KL = 1, 4)
  WRITE(10,*) (PNTS(I,KL,3), KL = 1, 4)
  CALL FBPC(U,4,0,PNTS,H0,ICOL,LTYPE)
850 CONTINUE

RETURN

FORMAT(1X,I5,1X,A7,3(1X,I2))
END
Appendix C - FeatureMod User’s Guide

Overview

This guide is designed to document the use of FeatureMod by answering the following questions:

• How is FeatureMod software accessed?
• What requirements must be met to run FeatureMod?
• What functions can FeatureMod perform and how are they performed?

The information contained in this guide is organized by functional topic such as how to run FeatureMod, how to create and retrieve model assemblies, how to generate shaded images, etc. The reader is assumed to be familiar with the logon procedure for VPI&SU’s VM3 system and have access to a VM3 userid.
**Functional Capabilities of FeatureMod**

In the current version of FeatureMod a user can perform the following functions:

- interactive creation of a geometric model using features
- display model geometry as a three dimensional surface image
- rotate and scale the three dimensional image using valuator dials
- display multiple views of the model
- produce hidden surface image of the model assembly
- produce a shaded image of the model assembly
- change the red, blue and green fractions of the first displayable workstation color
- interactively store and retrieve models

All of the above functions are discussed in the "FeatureMod Functions" section of this user's guide.
FeatureMod Run-time Requirements

The following are the requirements to run FeatureMod on VPI&SU’s VM3 system.

Userid Requirements

To run FeatureMod on a VPI&SU VM3 userid, the userid must have:

• access to at least 12 megabytes of virtual storage (memory)
• 25 loader tables defined

File Requirements

The following files are need to successfully run FeatureMod:

- Feature Exec
  is the command language program used to run FeatureMod
- Feature Text
  is the text file containing all modules that form FeatureMod

Hardware Requirement

FeatureMod currently runs only on the IBM 5080 workstation. Thus an IBM 5080 must be logically attached to your userid.
Running FeatureMod

If you have met all the requirements listed in the previous section you can run FeatureMod by typing the following command:

FEATURE

FEATURE is a command language program which performs all the actions necessary to link, load and run the FeatureMod software. The FeatureMod main menu screen should appear on the IBM 5080 screen. Depending on the current load on the system it may take up to one minute for this screen to appear. When the menu does appear the following messages will be printed in the grey message area at the bottom of the screen:

<table>
<thead>
<tr>
<th>Select Start or Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Welcome to "FeatureMod"
A Feature Based Geometric Modeling System
Version 1.0
Developed at
Virginia Polytechnic Institute And State University

The first step is to start a new model file or to recall a previously created model file.
FeatureMod Functions

Overview

The following pages contain details of how to use the functions currently available in FeatureMod. Under each function the following format is used to explain the function's use:

Purpose
explains the purpose of the function.

Menu Path
shows the nesting of the menu items which lead to the desired function.

Description
describes the function's use including an example of the FeatureMod message, prompt and input areas as shown below:

<table>
<thead>
<tr>
<th>Prompt Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Area</td>
</tr>
<tr>
<td>message line 9</td>
</tr>
<tr>
<td>message line 8</td>
</tr>
<tr>
<td>message line 7</td>
</tr>
<tr>
<td>message line 6</td>
</tr>
<tr>
<td>message line 5</td>
</tr>
<tr>
<td>message line 4</td>
</tr>
<tr>
<td>message line 3</td>
</tr>
<tr>
<td>message line 2</td>
</tr>
<tr>
<td>MESSAGE LINE 1 (CURRENT MESSAGE)</td>
</tr>
</tbody>
</table>

Example
provides an example of the function's use.
Starting A New Model

Purpose
To activate a new file for interactive model creation

Menu Path
FILES (START)

Description
When you select the “START” menu item, you will be asked to enter the name for the new model as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Model To Be Started</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
</tr>
</tbody>
</table>

You must type in a file name with maximum four characters. If less than four characters are used zeros will be post-concatenated to the name. The file name is actually the CMS filetype. The CMS filename and filemode are “FILE” and “A1” respectively.

If the model name you entered already exists on your userid, you will receive a “CANNOT START EXISTING MODEL” message and will be asked to reenter a new filename. If the file is not found then you will receive the message “STARTED MODEL filename”.

Example
Enter “TRY” to start a new model file. You should get the “STARTED MODEL TRY0” prompt.
Reading Existing Model Files

Purpose
To load a previously defined model file.

Menu Path
FILES (RECALL)

Description
When you select the "RECALL" menu item, you will be asked to enter the name of a previously created model file as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Model To Be Recalled</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
</tr>
</tbody>
</table>

Recalling Model filename
Model Recalled

You must type in a file name. The file name is actually the CMS file type. The CMS filename and filemode are "FILE" and "A1" respectively.

If the model name you entered does not exist on your userid, you will receive a "CANNOT RECALL NON-EXISTENT MODEL" message and will be asked to reenter a new filename. If the file is found then you will receive the message "RECALLING MODEL filename".

Example
Enter "PAWN" to recall the model that contains the geometry for a chess pawn. After recallsing the model you should get the "MODEL RECALLED" prompt.
Defining An Object

Purpose
To allow the definition of an object name and its creation by using feature instances

Menu Path
COMMANDS (MODELING (DEFINE (SLAB / CYLINDER / ... )))

Description
When you select the "DEFINE" menu item, you will be asked to enter the name for the new object. The process for object definition is as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Object To Be Defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectname</td>
</tr>
</tbody>
</table>

Defined Object "objectname"

Once the name is input, it is checked for previously existing assemblies or objects. If the component already exists then you will receive a "OBJECT ALREADY EXISTS" message and will be asked to reenter a new object name. The command can be cancelled by entering "CANCEL". Once the object is defined, a new menu page will allow the selection and creation of different features that will form part of the defined object.

Example
Enter "MOUNT" to define a new object. You should get the "DEFINED OBJECT MOUNT" prompt.
Defining An Assembly

Purpose
To allow the definition of an assembly name and its creation by using existing objects.

Menu Path
COMMANDS (MODELING (ASSEMBLE))

Description
When you select the “ASSEMBLE” menu item, you will be asked to enter the name for the new assembly. The process for assembly definition is as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Assembly To Be Defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>assemname</td>
</tr>
</tbody>
</table>

Defined Assembly “assemname”

Once the name is input, it is checked for previously existing assemblies or objects. If the component already exists then you will receive a “ASSEMBLY ALREADY EXISTS” message and will be asked to reenter a new assembly name. The command can be cancelled by entering “CANCEL”. Once the assembly is defined, a template will allow the selection of different objects that will form part of the defined assembly.

Example
Enter “SEAT” to define a new assembly. You should get the “DEFINED ASSEMBLY SEAT” prompt.
Adding To an Object or Assembly

Purpose
To allow addition to existing objects or assemblies.

Menu Path
COMMANDS (MODELING (ADDTO))

Description
When you select the “ADDTO” menu item, you will be asked to enter the name for the component that must be added to. The process for adding to a component is as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Component To Be Added To</th>
</tr>
</thead>
<tbody>
<tr>
<td>compname</td>
</tr>
</tbody>
</table>

Adding To “compname”

Once the name is input, it is checked for existing assemblies or objects. If the component does not exist then you will receive a “COMPONENT DOES NOT EXIST” message and will be asked to reenter a new component name. The command can be cancelled by entering “CANCEL”. Once the required name is obtained the appropriate menu page or template is displayed.

Example
Enter “SEAT” to delete the previously defined assembly. You should get the “DELETED ASSEMBLY SEAT” prompt.
Deleting an Object or Assembly

Purpose
To allow deletion of objects or assemblies.

Menu Path
COMMANDS (MODELING (DELETE))

Description
When you select the “DELETE” menu item, you will be asked to enter the name for the component that must be deleted. The process for component deletion is as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Component To Be Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>compname</td>
</tr>
</tbody>
</table>

Deleted “compname”

Once the name is input, it is checked for existing assemblies or objects. If the component does not exist then you will receive a “COMPONENT DOES NOT EXIST” message and will be asked to reenter a new component name. The command can be cancelled by entering “CANCEL”. Once the required name is obtained the appropriate component is removed from the object hierarchy and deleted from the screen.

Example
Enter “SEAT” to add to the previously defined assembly. You should get the “ADDING TO ASSEMBLY SEAT” prompt.
Displaying an Object or Assembly

Purpose
To allow the display of a single object or assembly.

Menu Path
COMMANDS (DISPLAY)

Description
When you select the "DISPLAY" menu item, you will be asked to enter the name for the component that must be displayed. The process for assembly definition is as shown below:

<table>
<thead>
<tr>
<th>Enter Name Of Component To Displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>compname</td>
</tr>
</tbody>
</table>

Displayed "compname"

Once the name is input, it is checked for existing assemblies or objects. If the component does not exist then you will receive a "COMPONENT DOES NOT EXIST" message and will be asked to reenter a new component name. The command can be cancelled by entering "CANCEL". Once the required name is obtained the appropriate component is displayed.

Example
Enter "SEAT" to add to the previously defined assembly. You should get the "DISPLAYED ASSEMBLY SEAT" prompt.
Model Inquiry

Purpose
To allow the inquiry of total number of objects and assemblies in the model and their names

Menu Path
COMMANDS (INQUIRE)

Description
When you select the “INQUIRE” menu item, you will get the total number of objects and assemblies in the message area. Their names and structure ids are displayed on the text terminal. The message area is as shown below:

<table>
<thead>
<tr>
<th>Select Option</th>
</tr>
</thead>
</table>

| Number Of Assemblies: 2 |
| Number Of Objects: 4 |

Information of the objects that form a particular assembly is also displayed on the text terminal.

Example
The above message indicates that the model has 2 assemblies and 4 objects.
Displaying Multiple Views

Purpose
To allow the display of four primary views of a component

Menu Path
COMMANDS (WINDOWS (MULTIPLE))

Description
When you select the "MULTIPLE" menu item, you will get the four primary views of the component that is currently being displayed.

Displaying Single Views

Purpose
To allow the display of primary views of a component

Menu Path
COMMANDS (WINDOWS (SINGLE (FRONT / TOP / .... )))

Description
When you select the menu item, you will get the appropriate primary view of the component that is currently being displayed. Once the required view has been selected the viewing matrix is reset to the viewing parameters specified for that view.
Displaying A Hidden Surface Image

Purpose
To allow the display of a hidden surface image of the entire model.

Menu Path
COMMANDS (RENDERING (HIDDEN SURFACE))

Description
When you select the "HIDDEN SURFACE" menu item, processing for image generation will begin. The message screen will be as shown below:

<table>
<thead>
<tr>
<th>Select Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Creating Hidden Surface Image

Once the image has been generated you will receive a "HIDDEN SURFACE IMAGE CREATED". To reset the image use menu item "RESET".
Displaying A Shaded Image

Purpose
To allow the display of a shaded image of the entire model.

Menu Path
COMMANDS (RENDERING (SHADING (CONSTANT)))

Description
When you select the "CONSTANT" menu item, processing for image generation will begin. The message screen will be as shown below:

<table>
<thead>
<tr>
<th>Select Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Creating Shaded Image

Once the image has been generated you will receive a "SHADE IMAGE CREATED". To reset the image use menu item "RESET".
Performing Three Dimensional Viewing Transformations

Purpose
To rotate and zoom the three dimensional image of the model

Menu Path
Any menu that is not requesting numeric or string input

Description
The workstation valuator dials are used to perform viewing transformations and are mapped as follows:

• dial 1 - global x-axis rotation
• dial 2 - global y-axis rotation
• dial 3 - global z-axis rotation
• dial 4 - view scale

If multiple views of the model are displayed then the viewing transformations are applied to the current view.
Appendix D - Feature Inquiry Program

Purpose
To allow a user to query the feature dictionary for information about individual features.

Description
This program requires three files for successful execution:

- **INFO EXEC** is the command language program
- **INFO TEXT** is the module containing commands that can be processed
- **INFO DATA** contains information about individual features

Running the Program
You can run the program by typing **INFO**. Once execution starts you will get a prompt asking for the feature you wish to inquire about. Once you have entered the name of the feature, you will be prompted for the kind of information that you need. The following words may be use to get the required information:

- **FEATURE** provides the name of the feature.
- **ENGLISH** provides the definition of the feature.
- **GROUP** provides the classification of the feature according to topology and shape-form
- **SYNONYM** provides a list of synonym features.
- **RELATIONS** provides a list of mathematical relations that govern the feature parameter ratios.
- **DEFINING** provides a list of parameters that define the feature.
- **END** to exit from the program.

The program requires only the first three characters of the keywords.
The vita has been removed from the scanned document.