Interactive 3-D Computer-Aided Design of External Spur Gears Cut by a Hob

by

Gary M. Irwin

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. A. Myklebust

Dr. H. H. Mabie

Dr. C. F. Reinholtz

April, 1986

Blacksburg, Virginia

Interactive 3-D Computer-Aided Design of External Spur Gears Cut by a Hob

by

Gary M. Irwin

Dr. A. Myklebust

Mechanical Engineering

(ABSTRACT)

An interactive program is presented which enhances the design of external spur gears cut by a hob. The program code calculates the geometry of an involute spur gear with trochoidal fillets and then uses the Graphical Kernel System (GKS), CADAM, and MOVIE.BYU to represent and display the gear. GKS, an international standard, is used to represent the gear in two dimensions; while the CAD/CAM system CADAM and the software package MOVIE.BYU accurately create wireframe geometric design models in three dimensions. Examples of the input parameters needed and each of the software packages in use are shown and explained.

ACKNOWLEDGEMENTS

The author would like to extend his deepest gratitude to his advisor, Dr. Arvid Myklebust, for his guidance, support, dedication, and training throughout the course of this study. Also, Dr. Hamilton H. Mabie and Dr. Charles F. Reinholtz are thanked for their guidance, service on the graduate committee, and expertise in the field of mechanisms. And a sincere appreciation is expressed to Dr. Reginald G. Mitchiner for his help and earlier work on gear drawing programs.

Gratitude is also expressed to Mitch Keil, Sandi Pennington, and Ashit Gandhi for their expertise and help throughout this thesis.

A special thanks goes to Regina Dugan Rieves, Tony Spagnuolo, Mike Hersh, Bob Williams, and Nestor Sanchez for their support and timely discussions.

Finally, the author is indebted to his parents and family for their continuous love and support in everything he chooses to do.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1 INTRODUCTION

Computer-Aided Design/Computer-Aided Manufacturing, CAD/CAM, continues to infiltrate every phase of the engineering design and production process. This thesis emphasizes the use of interactive two-dimensional and three-dimensional graphics as a design tool. An interactive program which calls on major CAD/CAM software packages using CAD interfaces will be used to create, modify, analyze, and optimize the design of external spur gears cut by a hob.

The most important design step when using a CAD system is the geometric modeling. Geometric modeling, deals with the designer making an accurate mathematical description of the geometry of a part. Once this geometry has been established, a whole host of other computer functions can be added to enhance the software package and help the designer. Examples which can be employed fall into the categories of engineering analysis, design review and evaluation, drafting, and numerical control (NC) programming. This is why two chapters of this thesis, Chapters 3 and 4, have been devoted to the development of the spur gear geometry.

The geometry and many of the resultant characteristics of a gear are dependent upon its manufacturing technique. When producing an external spur gear with a hob (rack) cutter, (as long as the pitch line of the rack represents the hob pitch line), the resultant tooth geometry is an involute profile. The involute profile is generated outward from the base

circle of the gear. Inside the base circle, the profile is defined to be a straight radial line. However, when hobbing, a trochoid fillet is formed. The trochoid fillet will intersect or be tangent to the involute curve at some point if the parameters chosen make it geometrically possible.

The interactive computer program was written in FORTRAN 77 using an IBM 4341 running the VM/CMS system. It mathematically represents the involute curve, the trochoid fillet, and the rest of the gear geometry and writes this data to a CAD data base. The flow chart in Figure 1 demonstrates the overall control structure for the program.

Design features of interest include geometry generation for normal and undercut gear teeth, standard and nonstandard gears, and a plain gear blank design. Also of interest is a parameter for the hob tip radius. This allows the user to specify any hob (rack) cutter from a sharp-tipped edge to a fully-rounded edge. This is explained more fully in Chapter 3 along with the rest of the gear geometry.

The program next calls on the major software packages GKS, CADAM-CADCD, and MOVIE.BYU to take the numerical data and graphically display it on the computer terminal screens. Chapter 5 discusses each of these packages, their interfaces, and how they are used by the interactive program.

It should be noted that only the initial wire frame geometric model of the gear will be created in this investigation. Surfaces, solids,

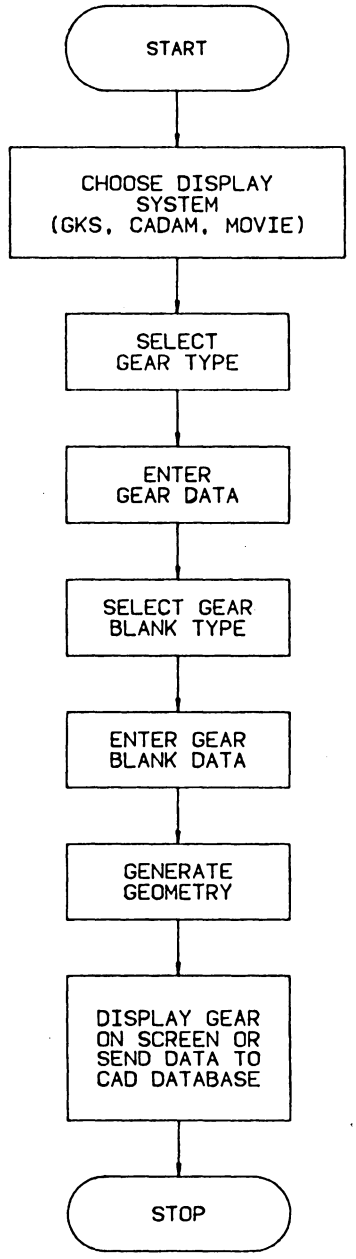


Figure 1. Overall Control Structure for Gear Program

hidden-line elimination, and other functions must be done manually with the software packages mentioned. But, with the geometry established, the model can now be called from the data files for review, analysis, and changes. The program can also be expanded to encompass different gear types, blank types, or answer other design questions if compatible subroutines are added. Expandibility of the program will be discussed throughout the thesis with a detailed coverage in Chapter 6.

CHAPTER 2 LITERATURE REVIEW

Many authors have dealt with the design of spur gears [1-21], however, only a few have used the computer as a graphical design tool [15-21]. And there were no papers found that used a three-dimensional (3-D) CAD system to help in the design of gears. Gears shown in three dimensions and graphically produced have been pictured in magazines, but they are produced for visual impressiveness. A closer look at these gears will reveal a simplified geometry; either the fillet is ignored, the teeth are square, or some other simplification has been made. Such examples can be found in the ASME NEWS [25] and the gear models produced by Baer [26].

One of the earliest investigations that used the computer to draw a hobbed involute gear tooth (spur or helical) was Kochanek [15] in 1968. Kochanek used a digital computer to plot enlarged profiles of a gear tooth cut by a hob. Figure 2 shows an example of his work, a computer drawn profile of a segment of a finished spur gear tooth. The computer program was devised to plot a tooth profile using either set of input data shown as follows:

A. Gear Tooth Profile Design (Input)

1. Hob tip radius, A, in.
2. Gear tooth addendum factor, k_a

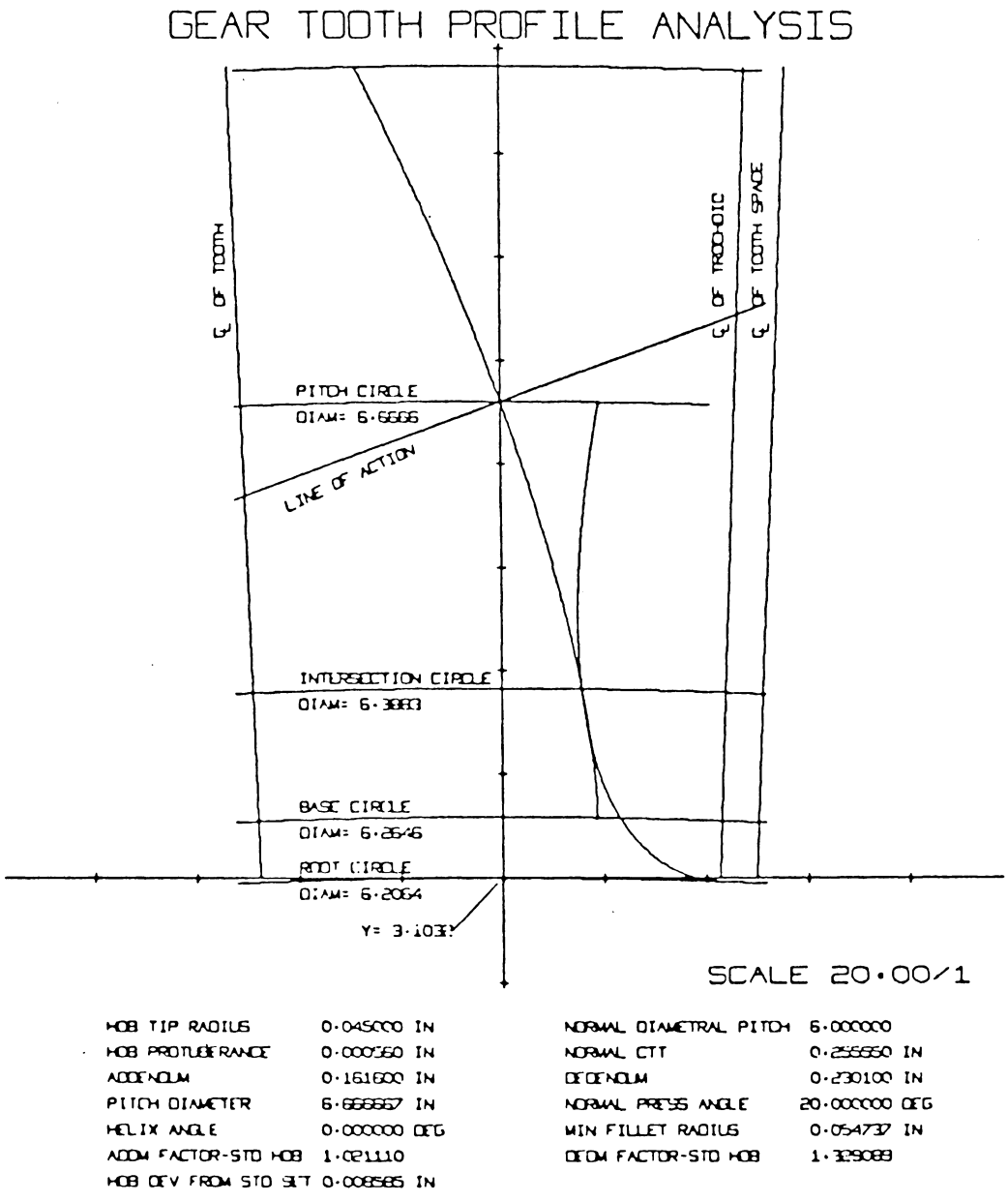


Figure 2. Computer-Drawn Profile of a Spur Gear Tooth [15]

3. Gear tooth dedendum factor, k_b
4. Normal diametral pitch, P_n
5. Deviation of hob setting from standard, Q, in.
6. Hob protuberance, S, in.
7. Pitch diameter, D, in.
8. Normal pressure angle, ϕ_n, deg.
9. Helix angle, ψ, deg
10. Magnification factor, M

B. Gear Tooth Profile Analysis (Input)

1. Hob tip radius, A, in.
2. Normal diametral pitch, P_n
3. Hob protuberance, S, in.
4. Normal circular tooth thickness, t_n, in.
5. Gear tooth addendum, a, in.
6. Gear tooth dedendum, b, in.
7. Pitch diameter, D, in.
8. Normal pressure angle, ϕ_n, deg.
9. Helix angle, ψ, deg.
10. Magnification factor, M

Two forms were developed because the data available for designing a new gear is usually different from the data used for analyzing an existing

one. The computer program was also based on the following equations that give rectangular coordinates of points on the tooth profile:

$$x = R_n\cos\phi_n\{[\cos\omega+\omega\sin\omega]\sin\Omega-[\sin\omega-\omega\cos\omega]\cos\Omega\}$$

$$y = R_n\cos\phi_n\{[\cos\omega+\omega\sin\omega]\cos\Omega+[\sin\omega-\omega\cos\omega]\sin\Omega\}-R_n+b$$

for the involute rectangular coordinates and

$$x = \{[1+\frac{A}{\sqrt{(R_n\theta)^2+(b-A)^2}}][R_n\theta\sin\theta-(b-A)\cos\theta]+R_n\cos\theta\}\sin\gamma$$
$$- \{[1+\frac{A}{\sqrt{(R_n\theta)^2+(b-A)^2}}][R_n\theta\cos\theta+(b-A)\sin\theta]-R_n\sin\theta\}\cos\gamma$$

$$y = \{[1+\frac{A}{\sqrt{(R_n\theta)^2+(b-A)^2}}][R_n\theta\sin\theta-(b-A)\cos\theta]+R_n\cos\theta\}\cos\gamma$$
$$+ \{[1+\frac{A}{\sqrt{(R_n\theta)^2+(b-A)^2}}][R_n\theta\cos\theta+(b-A)\sin\theta]-R_n\sin\theta\}\sin\gamma-R_n+b$$

for the trochoid rectangular coordinates,

where

$$b = \frac{k_b}{P_n} + Q \quad \text{(Gear dedendum), in.}$$

R = pitch circle radius of gear, in.

$$R_n = \frac{R}{\cos^2\psi} \text{ , in.}$$

$$\gamma = \frac{(b-A)\tan\phi_n + \frac{A-S}{\cos\phi_n}}{R_n} \text{ , radians}$$

and

θ = independent variable in trochoid equations, radians

ω = independent variable in involute equations, radians

$\Omega = \tan\phi_n - \phi_n$, radians

Kochanek used the root circle for computing the tooth profile points, where most authors use the base circle as their reference circle.

Another method to generate gear teeth using a computer was developed by Cooley [16]. His paper describes computer programs that were written to define the geometry of a rack using 2-D interactive graphics. Then, he stored this shape and used it to generate the gear tooth by stepping the rack in a finite number of positions along a certain line. Figure 3 demonstrates how each movement will remove an amount of material. Profile shifts and the use of nonstandard cutters are both possible with this method.

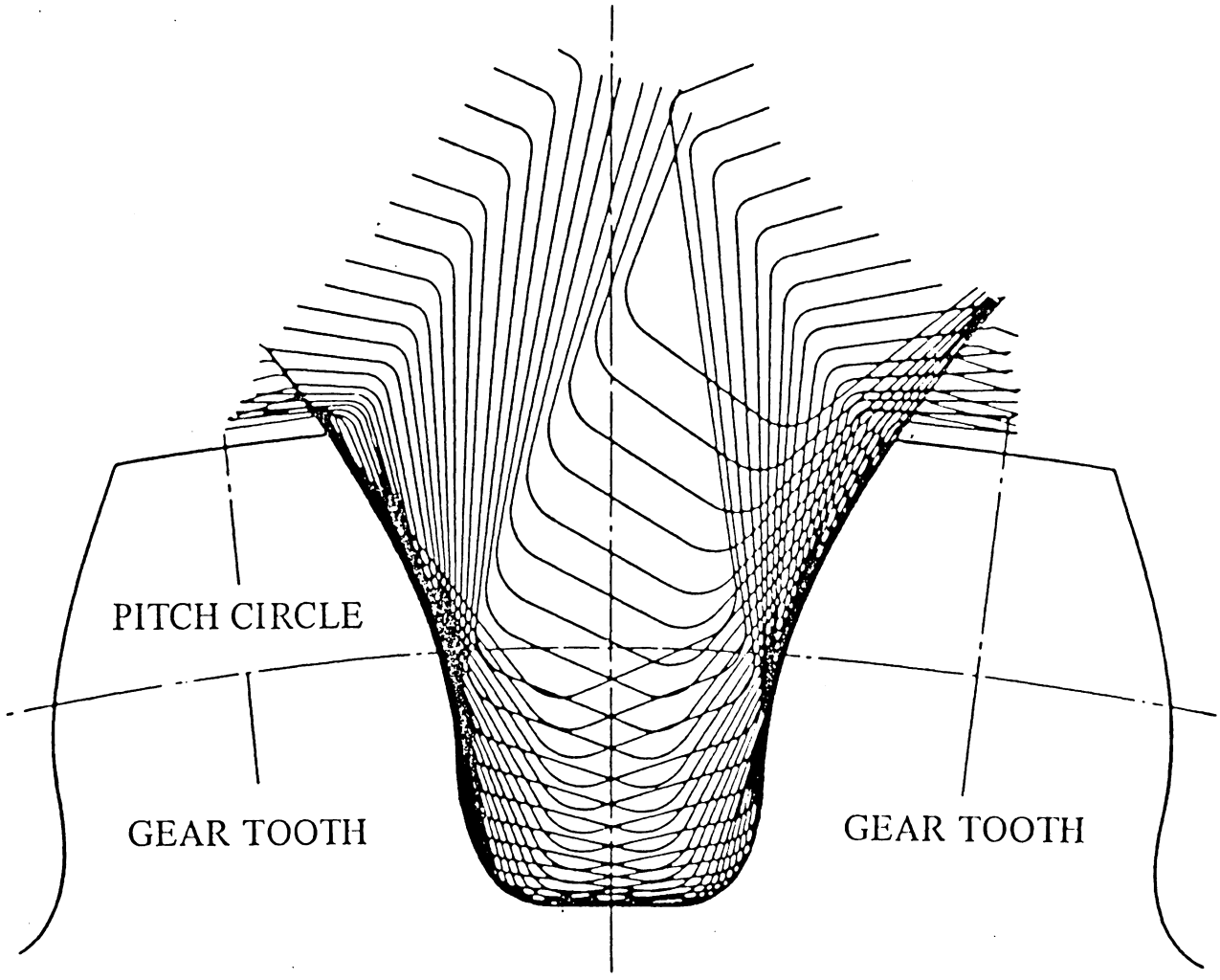


Figure 3. Rack Creating a Gear Tooth [7]

Other authors who give the equations for the involute curve and trochoid fillet and/or discuss a computer program to calculate geometry coordinates are Dudley [5,6], Khiralla [7], and Mitchiner and Mabie [17]. Mitchiner and Mabie's equations and work were the basis for the gear's geometry in this investigation.

Each of the authors, with the exception of Kochanek, developed the 2-D gear-tooth profile for further analysis. They each needed the actual shape of the gear tooth for further investigation, because the more accurate their model, the better the results. Some of the main areas of interest are the determination of the area enclosed by the meshing teeth [16], the Lewis form factor [17], the AGMA geometry factor J [17], undercutting calculations [18,19], and finite element models [20,21].

Although the above papers have each contributed to the advancement of gear design, none of them deal with representing and drawing the gear in three dimensions. An important distinction should be made between computer graphic pictures of gears and design models of gears which can be interactively modified. This program, unlike others, generates design models. The gears can be accurately incorporated into a 3-D design model. A more accurate and complete analysis is possible due to the representation of three dimensions. And if gear sectors are to be cut for a special part, this can be accomplished by using modern three or five axis NC machining centers and CAD/CAM NC part programming when a 3-D model is available to specify the cutter path.

CHAPTER 3 HOB AND GEAR GEOMETRY

3.1 INTRODUCTION

Spur gears are used to transmit power between two parallel shafts. The principal advantages of spur gearing are the elimination of end thrust and axial displacement limits, the general economy in manufacturing and maintanence, and the simplicity of manufacturing.

As mentioned earlier, the manufacturing technique determines many characteristics of a gear, including its geometry and use. Some of these techniques include stamping, casting, milling, and generating. Stamping and casting are usually used when large volume, low cost, less accurate gears are acceptable. Milling with a form cutter is used when a small quantity of gears are needed at a low cost, and usually when accurate tooth spacing is not required. However, the most widely used method is shaping with a generating cutter. This process economically produces quiet running, accurate, high strength teeth for both large and small volume runs.

The two generating methods are gear hobbing and gear shaping. Both are versatile, economical, and accurate methods for producing gears; the difference is that the tool used for the hobbing process is a rack or "worm" type cutter, and the tool used in shaping is a "pinion" type cutter.

The geometry in this study is based upon the rack or hob cutter. The hob and the pinion cutter can both cut an involute profile, but the fillet formed during the cut is different. A hob creates a fillet curve known as a trochoid, the pinion cutter creates an epicycloid curve.

The involute tooth form was chosen for study in this thesis because its the simplest to manufacture and the most common tooth form. Also, it satisfies the continuous contact requirement, and the velocity ratio of two involute gears in mesh will not change if the center distance is altered.

The following sections will (1) establish the inputs to the program, (2) set up some special checks (3) mathematically describe the geometry of the tooth root land, trochoid fillet, involute flank, and the top land profiles of a gear cut with a hob, and (4) discuss how the program finds the rectangular coordinates for each of these curves. The geometry produced in this chapter will be used to generate the entire gear with the interactive program and the CAD systems.

3.2 PROGRAM PARAMETERS

This part of the program, shown in Appendix A and graphically expressed in Figure 4 as a flow chart, is adapted from Moosavi-Rad [19] and Jalilvand [20] and is set up for the designer to use a minimum number of parameters. A detailed example of the Input/Output (I/O) for the program

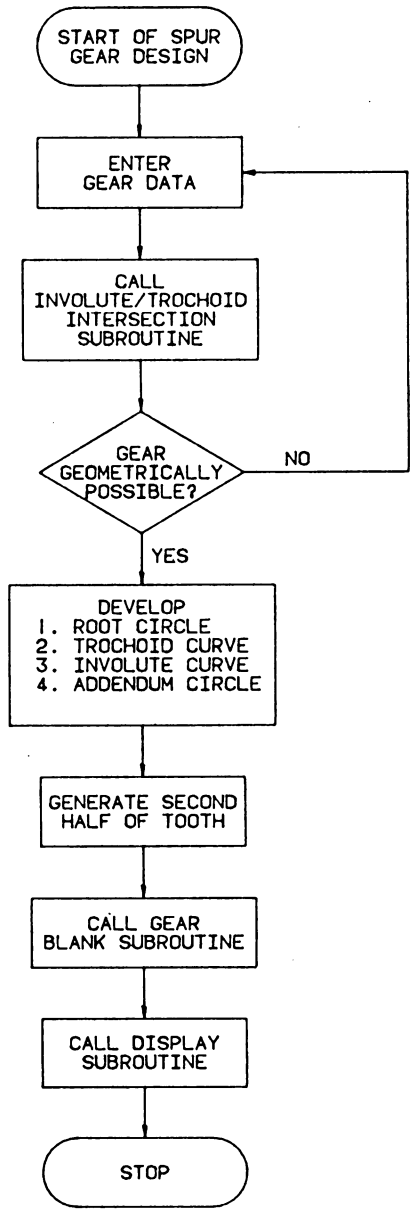


Figure 4. Program Generation of the Tooth Profile Geometry

is shown in Appendix G. The input in this case will generate the model in Figure 30 of this text.

For a standard gear, the parameter inputs are as follows:

1. The diametral pitch
2. An addendum fraction
3. A dedendum fraction
4. A hob tip radius
5. The number of teeth in the gear
6. The cutting pressure angle.

The diametral pitch is the ratio of the number of teeth to the pitch diameter, mathematically expressed as:

$$P = \frac{N}{D} \tag{1}$$

where N = number of teeth

D = pitch diameter

The definitions of the addendum and dedendum by AGMA (see Figure 5) for standard, course pitch, involute spur gears are given by:

$$a = \frac{1.000}{P} \qquad b = \frac{1.250}{P}$$

where a = gear addendum

b = gear dedendum (hob addendum)

P = diametral pitch

The addendum and dedendum fractions are the numerators of the actual addendum and dedendum before they have been divided by the diametral pitch.

The hob tip radius is a special feature in this study. The hob teeth can vary from a sharp-tipped hob, see Figure 6a, to a fully rounded hob, see Figure 6c. Many investigations have only studied the corner point of the sharp-tipped hob. However, in practice, all hobs are not sharp-tipped and actually, because of wear, this corner point is only theoretical. To solve this problem Mitchiner and Mabie [17] and Mitchiner, Mabie, and Moosavi-Rad [18] have developed a number of equations dealing with rounded-tipped hobs, many of which are included in this program. Further explanation of the hob tip radius and useful equations will be discussed in the next sections.

The number of teeth in the gear refers to the number of teeth the designer would like on the gear being designed. And the cutting pressure angle is the inclination of the rack profile.

Nonstandard gears can also be designed by the program by adding two more input parameters. In addition to the six already listed, the cutting pitch radius and the thickness of the gear tooth on the pitch circle are also needed.

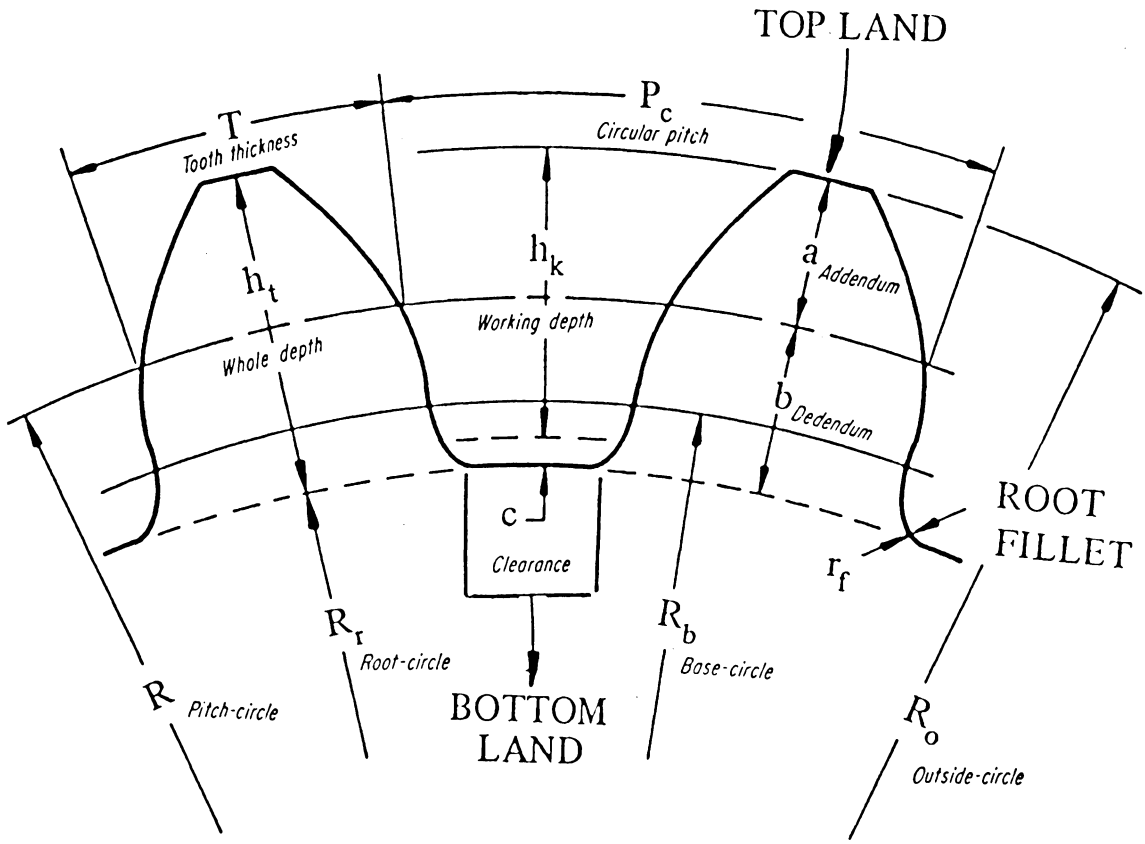


Figure 5. Gear Symbols and Nomenclature [7]

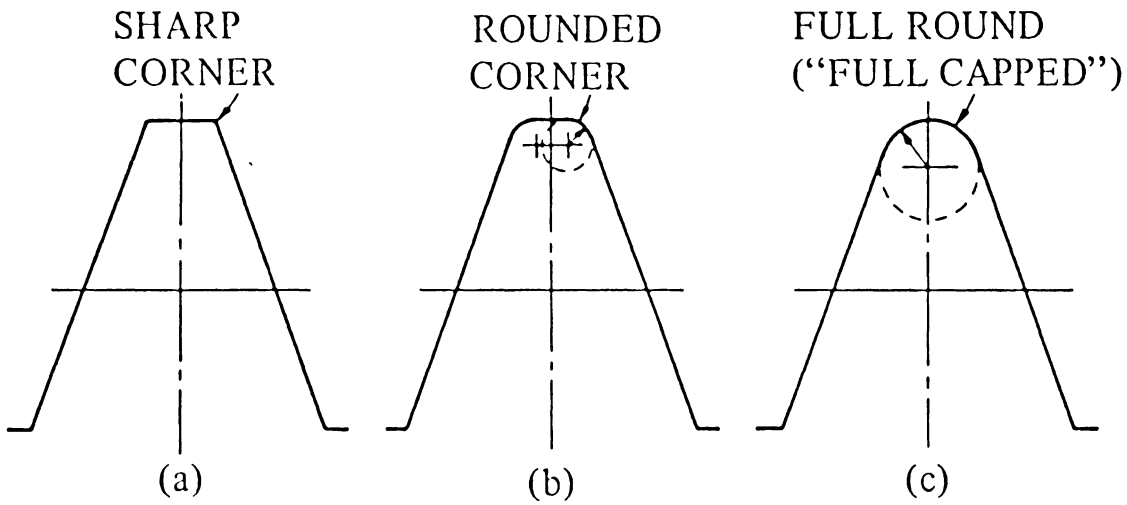


Figure 6. Hob Teeth with Different Types of Corners [7]

3.3 HOB TIP RADIUS

Figure 7 shows an enlarged hob tooth tip with rounded corners. The distance Δ, which is one-half the width of the rack-tip land has been described by Mitchiner and Mabie [17] as:

$$\Delta = \frac{\pi}{4P} - (b-r_f)\tan\phi - \frac{r_f}{\cos\phi} \tag{2}$$

When $\Delta = 0$ or the hob tooth tip is fully rounded, an expression can be found for the maximum hob tip radius geometrically possible. Mitchiner and Mabie have presented this as:

$$r_f = \frac{1}{1-\sin\phi}\left(\frac{\pi}{4P}\cos\phi - b\sin\phi\right) \tag{3}$$

The interactive program uses Equation 3 as a check of the hob tip radius entered by the designer. If the value entered exceeds r_f, the program will show the maximum value allowable and will ask the user to reenter the parameter.

3.4 MINIMUM NUMBER OF TEETH BEFORE UNDERCUTTING BEGINS

Another equation used in the program checks to see if the gear parameters chosen will produce a gear with undercut teeth. Undercutting of the teeth occurs when a portion of the active profile has been removed by the generating process. The tip of the hob, whether it is sharp or rounded, intersects the active involute profile, see Figure 8. If undercutting

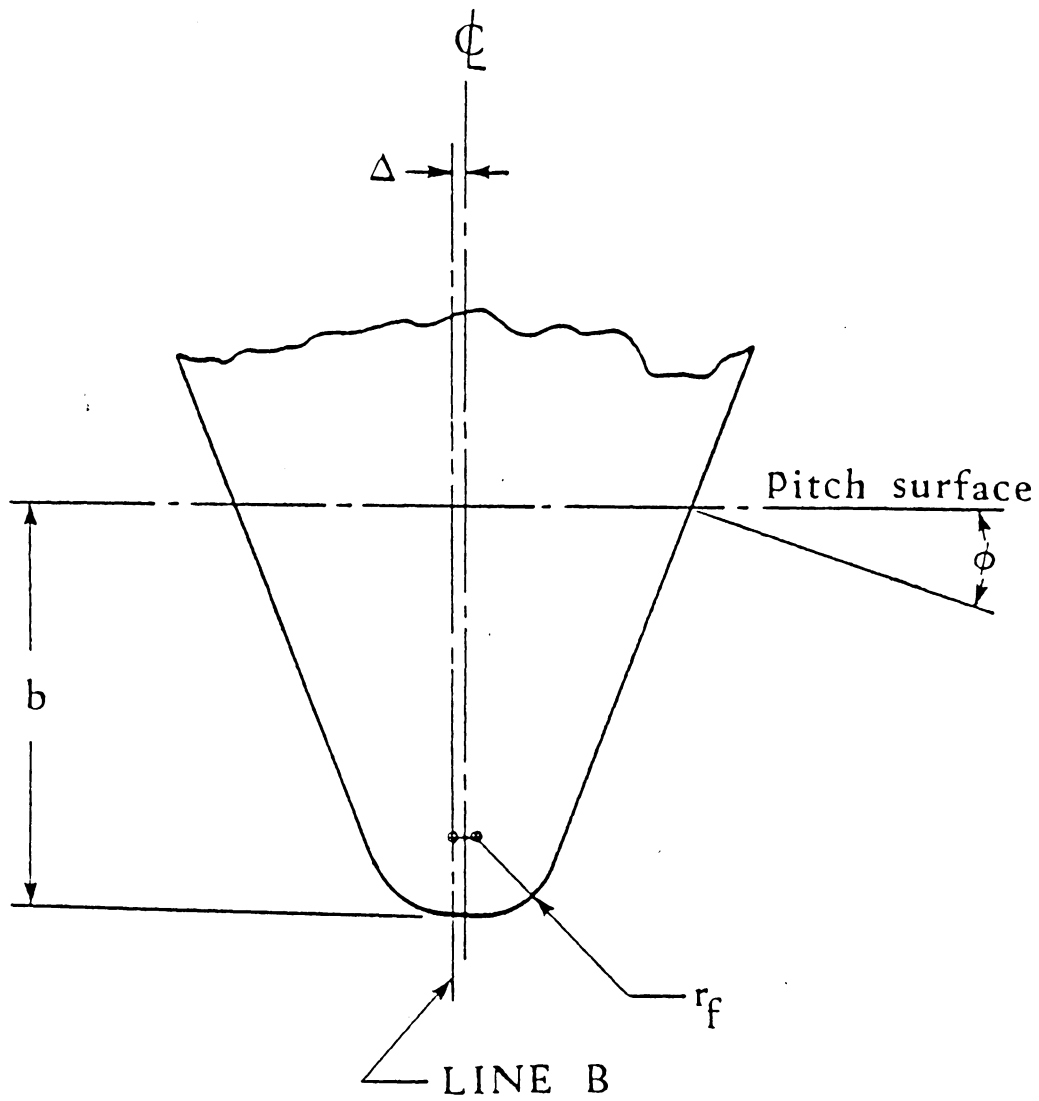


Figure 7. Enlarged Hob Tooth Tip [17]

is avoided, the trochoid fillet is tangent to the involute curve. Undercutting is considered undesirable because it reduces the length of action and weakens the gear teeth.

The equation developed by Mitchiner, Mabie, and Moosavi-Rad [18] finds the minimum number of teeth allowable before undercutting begins and is defined as:

$$N_1 = \frac{2P[b + r_f(\sin\phi - 1)]}{\sin^2\phi} \tag{4}$$

where P = diametral pitch

b = hob addendum (gear dedendum)

r_f = hob tip radius

ϕ = cutting pressure angle

As described by AGMA, b and r_f are functions of the diametral pitch.

Mabie and Ocvirk [8] independently developed an equation for the minimum number of teeth to avoid undercutting using a sharp-tipped hob. The equation is as follows:

$$N_1 = \frac{2b}{\sin^2\phi} \tag{5}$$

where b = hob addendum

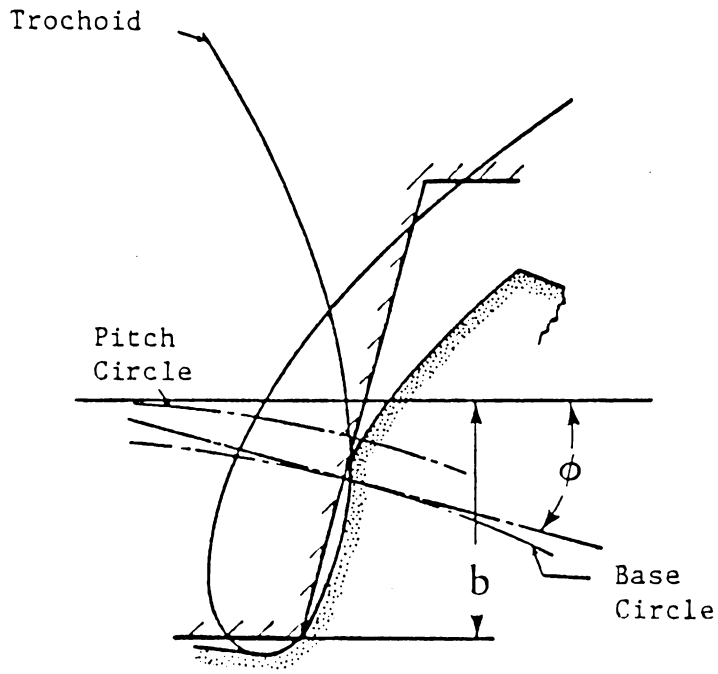


Figure 8. Intersection of the Trochoid and Involute Profile [18]

It can be seen that in the special case where $r_f = 0$, the two equations become identical.

It should be noted that the program only tells the designer that the gear is being undercut, and that the number of teeth in the gear should be increased to a minimum of N_1 to avoid undercutting. There are no calculations made for how much the gear is being undercut. Extensive research on the amount of undercutting and equations to calculate this amount has been done by a number of authors, including Mitchiner, Mabie, and Moosavi-Rad [18], Spotts [12], Martin [9], and Dudley [5].

3.5 GENERATION OF THE GEAR PROFILE GEOMETRY

3.5.1 TROCHOID FILLET

The following explains how a hob creates a trochoidal fillet. Figure 9 will help in the discussion. The center of the circle at the corner of the rack generates a rack trochoid. The points on the rack trochoid each have a circle of radius r_f around it. The trochoidal fillet is the envelope of the family of circles traced by the points on the rack cutter.

Mitchiner and Mabie [17] have parametrically described the coordinates of the trochoid curve cut with a rounded-edged hob (see Figure 10) as:

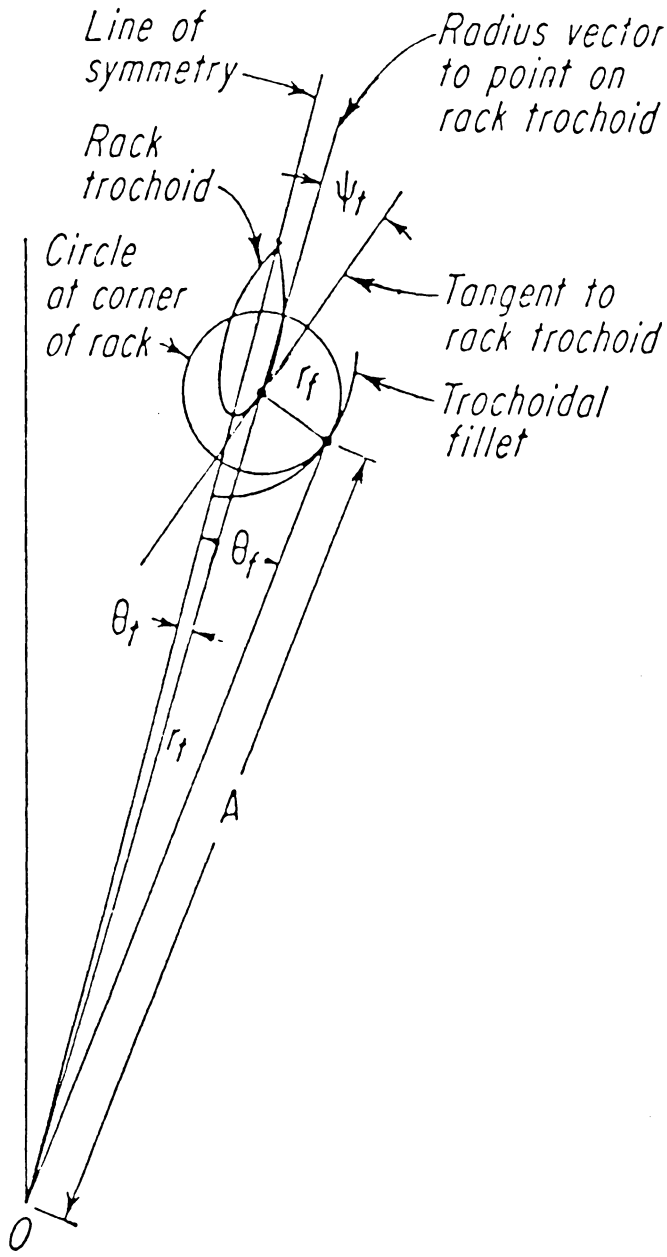


Figure 9. Geometry of the Trochoid Fillet [5]

$$x_H = (R - b + r_f)\sin(\beta+\theta) - R\theta\cos(\beta+\theta) - \frac{r_f}{\sqrt{(R\theta)^2 + (b-r_f)^2}}[R\theta\cos(\beta+\theta) + (b-r_f)\sin(\beta+\theta)] \quad (6)$$

$$y_H = (R - b + r_f)\cos(\beta+\theta) + R\theta\sin(\beta+\theta) + \frac{r_f}{\sqrt{(R\theta)^2 + (b-r_f)^2}}[R\theta\sin(\beta+\theta) - (b-r_f)\cos(\beta+\theta)] \quad (7)$$

where

$$\beta = \frac{\pi}{N} - \eta \quad (8)$$

$$\eta = \frac{\Delta}{R} \quad (9)$$

and Δ is defined as in Equation 2.

3.5.2 INVOLUTE CURVE

Figure 11 shows the generation of one-half an involute tooth profile. Point Q is a point on the involute. Angle AOQ is shown to be:

$$\text{Angle AOQ} = [(\frac{P}{2R} + \text{inv}\phi) - \text{inv}\mu]$$

and

$$R_Q = \frac{R\cos\phi}{\cos\mu}$$

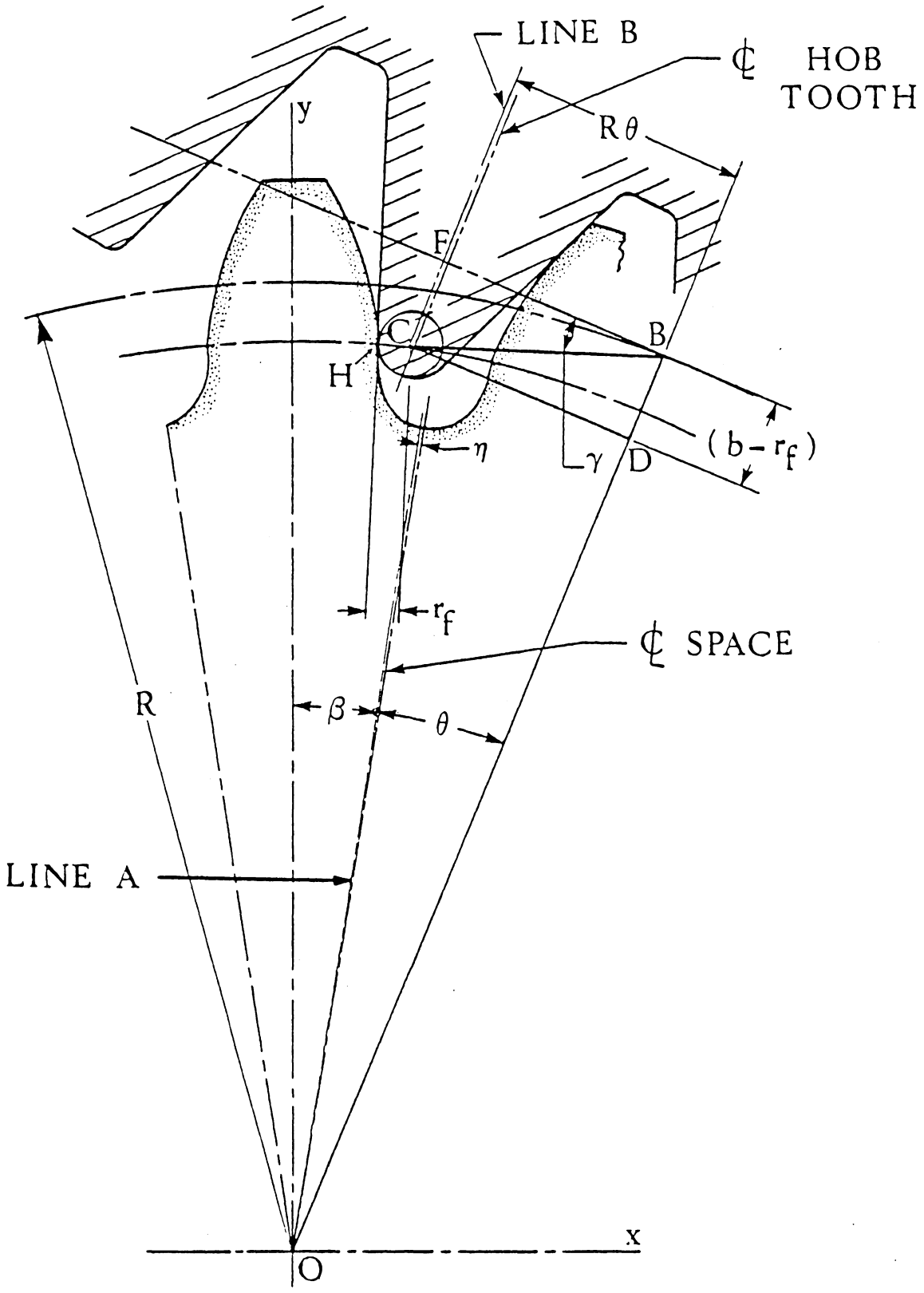


Figure 10. Engagement of the Hob and the Gear Blank [17]

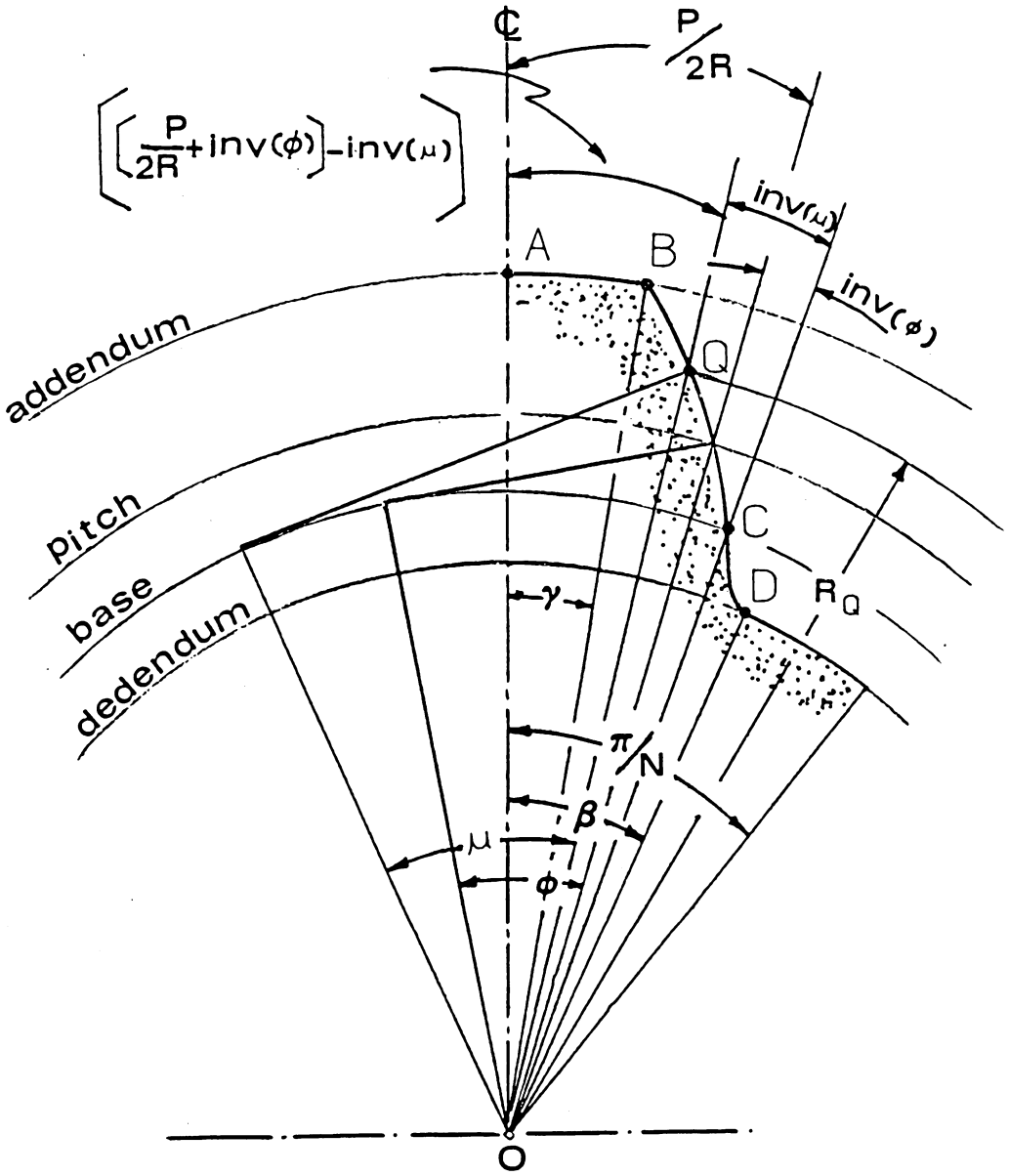


Figure 11. Geometry of the Gear Tooth Profile [20]

Therefore, the coordinates for the involute curve can be defined by:

$$x_Q = R_Q \sin(AOQ)$$

$$x_Q = R \frac{\cos\phi}{\cos\mu} \sin[(\frac{P}{2R} + inv\phi) - inv\mu] \quad (10)$$

$$y_Q = R_Q \cos(AOQ)$$

$$y_Q = R \frac{\cos\phi}{\cos\mu} \cos[(\frac{P}{2R} + inv\phi) - inv\mu] \quad (11)$$

3.5.3 ROOT AND TOP LANDS

Figure 11 can also be used to determine the equations to descibe the coordinates of the tooth root land and the top land of the gear profile. The root land coordinates are calculated using the equations as follows:

$$x_{root} = (R - b)\sin\phi \quad (12)$$

$$y_{root} = (R - b)\cos\phi \quad (13)$$

The top land coordinates are calculated in a similar manner using:

$$x_{add} = (R + a)\sin\phi \tag{14}$$

$$y_{add} = (R + a)\cos\phi \tag{15}$$

3.5.4 COORDINATE CALCULATIONS BY THE COMPUTER PROGRAM

Now that each section of the profile is expressed in mathematical terms, the user is asked how many intervals per curve are needed. This could be very important if the final function of the CAD/CAM system is NC programming. The more points per curve, the smoother and more exact the final part will be up to the limits of the NC machine. Unfortunately, the maximum number of intervals for each curve is usually limited by the major software package itself. Some trial and error may be necessary to find these maximum values.

Once this value is entered, the program initially uses equations 6, 7, 10, and 11 and the Newton-Raphson method to solve for point C in Figure 11. Point C is either the point of tangency or the point of intersection between the involute curve and the trochoidal fillet.

With this data known, the computer program uses equations 12 and 13 to calculate the root land coordinates by varying θ from π/N to β for the

number of intervals specified (see Figure 11). When the intersection between the root circle and the trochoid curve (point D) is located by comparison, calculations of the trochoid curve begins. This uses equations 6 and 7 with θ ranging from zero to the point of tangency or intersection (point C). Using equations 10 and 11, the involute coordinates are then compiled from point C to the intersection of the involute curve and the addendum circle (point B). Finally, equations 14 and 15 determine the top land coordinates for the number of intervals specified. θ starts from

$$[(P/2R + \text{inv}(\phi)) - \text{inv}(\gamma)]$$

where $\gamma = \arccos[R\cos\phi/(R + a)]$

and continues until $\theta = 0$.

One-half of a gear profile now exists. With a short do-loop the coordinates of the other half of the gear are generated as a mirror image about the y-axis in the x-y plane.

CHAPTER 4 GEAR BLANK GEOMETRY

4.1 INTRODUCTION

In as much as the manufacturing technique of the teeth affects the geometry and use of the gear, the original gear blank selection is just as critical to the design of the gear. Selection of the gear blanks also involves complex design decisions. The type of material used, the hardening process, and the way the blank was manufactured are all important questions which need to be answered if the proper gear or gears are to be selected. However, this is not the emphasis of this study, although it will be shown in this chapter that separating the geometry of the blank from the design decisions mentioned above is not easily done.

This chapter will discuss the three standard types of gear blank designs, and then highlight the plain gear blank design used in the program. Limitations and assumptions in the program will be noted as the need arises.

4.2 TYPES OF GEAR BLANKS

Standard gear blank patterns are utilized whenever possible for economic reasons. The three standard gear blank designs are shown in Figure 12: plain, webbed, and spoked. Each is shown with the three types of standard

hub extensions: type A, without the hub; type B, with the hub extension on one side; and type C, with the hub extension on both sides.

Figure 13 shows how the spoked blank will change with increasing face width. Similarly, the webbed blank also needs added support as the face width is increased. The webbed and spoked blanks are more complex since both change their geometries as the width increases. So, for simplicity, only the plain gear blank with the standard hub as an option was used in the program. The geometry subroutine for the gear blank can be found in Appendix B, while a flow chart of its structure is shown in Figure 14.

4.3 COMPUTATION OF THE PLAIN GEAR BLANK

Input parameters entered at this point (see Figure 15) are the face width F, the bore diameter G, and the type of plain gear blank (A, B, or C from Figure 12). If type B or type C is chosen, the hub extension distance S needs to be entered as well.

After a few simple checks to be certain the geometry is reasonable, the coordinates of the bore hole are calculated according to Figure 16 by:

$$x_Q = R_Q \cos\theta \tag{16}$$

$$y_Q = R_Q \sin\theta \tag{17}$$

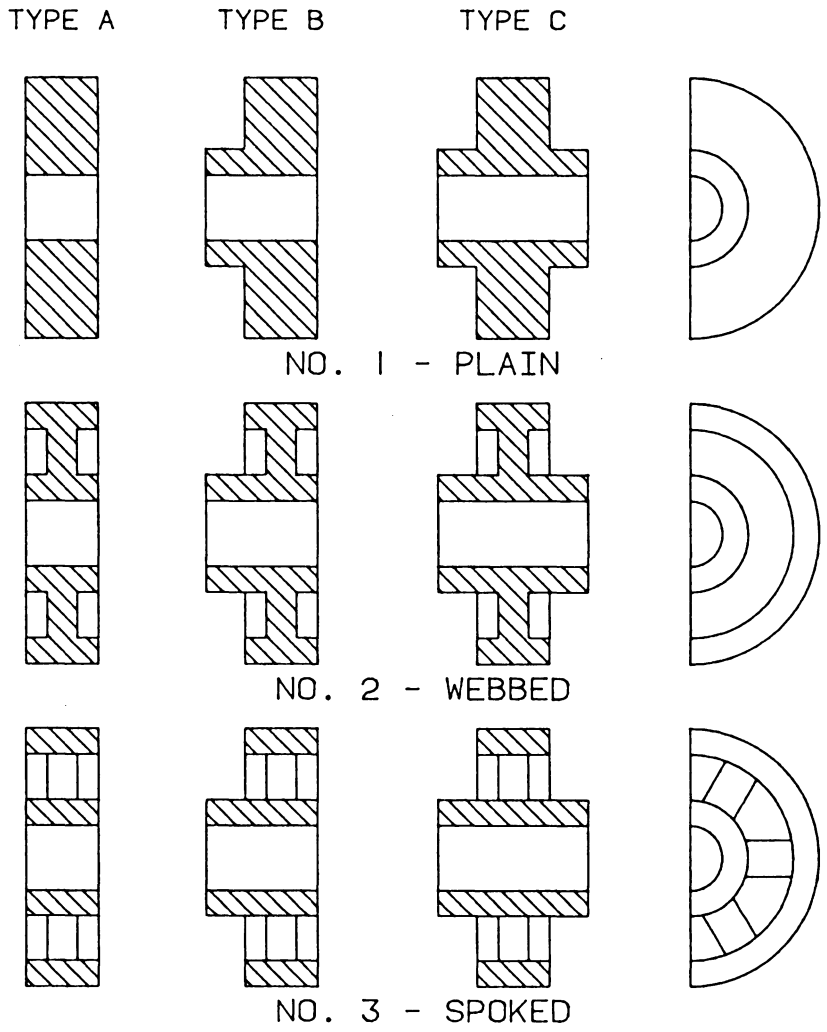


Figure 12. Types of Standard Gear Blanks [4]

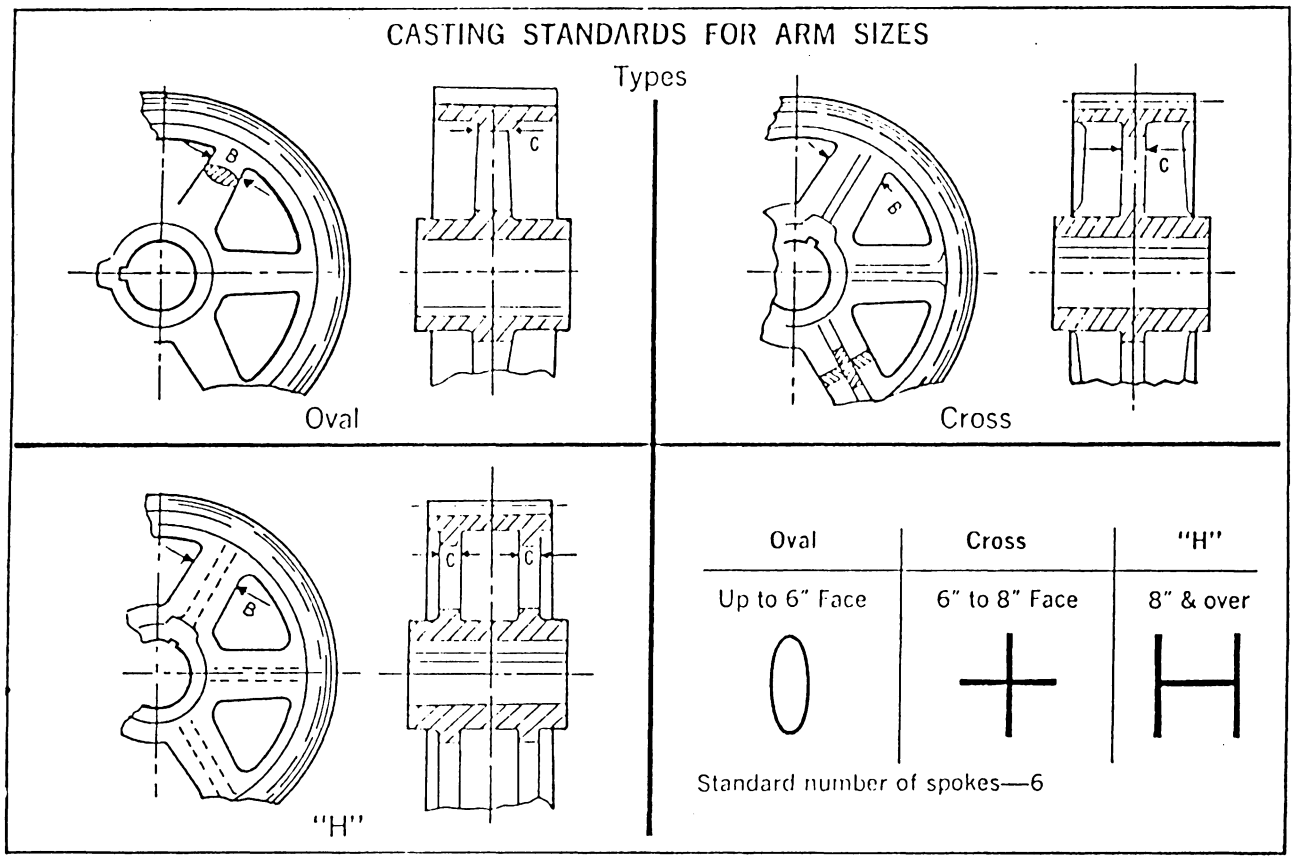


Figure 13. Changes in the Geometry of a Spoked Gear Blank [11]

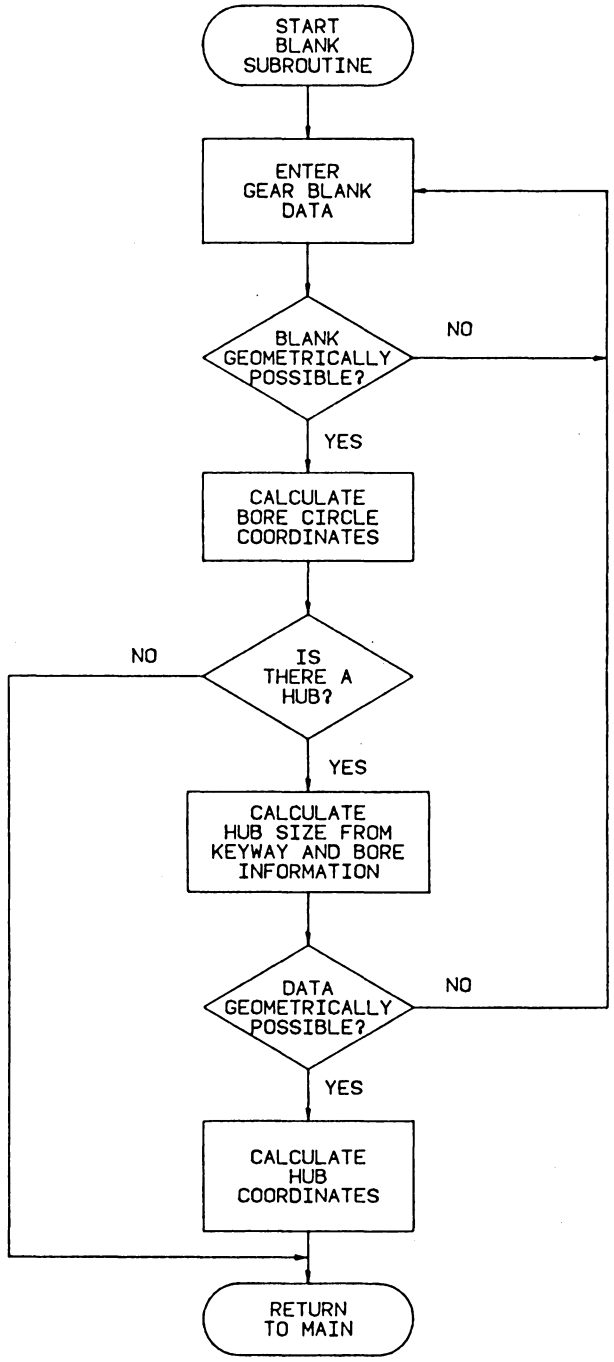


Figure 14. Program Generation of the Gear Blank Geometry

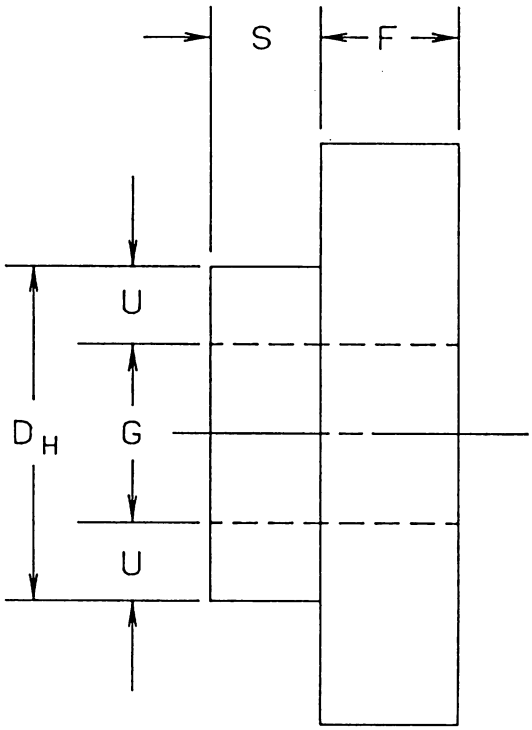


Figure 15. Profile of a Plain Gear Blank

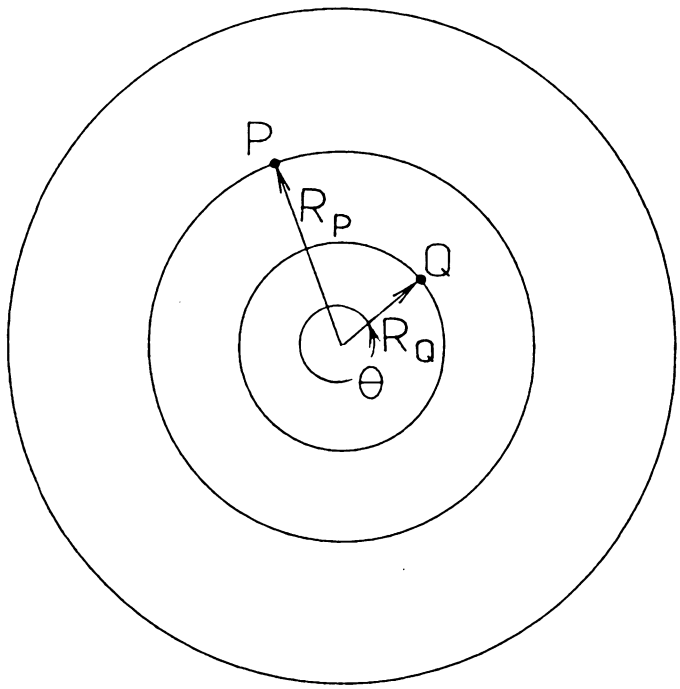


Figure 16. Front View of a Plain Gear Blank

where $0 \leq \theta < 360$.

Next, Table 1, AGMA recommended keyways for bores in gearing, is used to calculate the keyway depth, which in turn is used to calculate the hub wall thickness (U) by the formula:

Hub wall thickness = (2.5)(keyseat depth)

The depth of the standard keyway is one-half of the key height (H) measured at the edge according to Figure 17. Figure 17 and Table 1 are used for plain and gib head taper keys with the standard taper of one-eighth inch per foot where the depth shown is the deep end of the keyway.

Only the hub wall thickness is of interest for the geometry since there are no provisions in the program at this time for a keyway. However, the user is always able to manually insert a keyway. The approximate effect of standard keyways on the torsional strength of a solid shaft is shown in Table 2 as a reference (as long as the width equals one-quarter the shaft diameter and the depth is one-half the width) [11].

AGMA standards for hub design [11] calculate the hub diameter as:

(1.6)(bore diameter) for steel and high tensile bronze,

or

(1.8)(bore diameter) for cast iron and bronze.

TABLE I: KEYWAYS AND KEY STOCK

DIAMETER OF HOLES INCLUSIVE	STANDARD KEYWAYS AND KEYS		
	KEYWAYS		KEY STOCK
	WIDTH	DEPTH	
5/16 TO 7/16	3/32	3/64	3/32 X 3/32
1/2 TO 9/16	1/8	1/16	1/8 X 1/8
5/8 TO 7/8	3/16	3/32	3/16 X 3/16
15/16 TO 1 1/4	1/4	1/8	1/4 X 1/4
1 5/16 TO 1 3/8	5/16	5/32	5/16 X 5/16
1 7/16 TO 1 3/4	3/8	3/16	3/8 X 3/8
1 13/16 TO 2 1/4	1/2	1/4	1/2 X 1/2
2 5/16 TO 2 3/4	5/8	5/16	5/8 X 5/8
2 13/16 TO 3 1/4	3/4	3/8	3/4 X 3/4
3 5/16 TO 3 3/4	7/8	7/16	7/8 X 7/8
3 13/16 TO 4 1/2	1	1/2	1 X 1
4 9/16 TO 5 1/2	1 1/4	7/16	1 1/4 X 7/8
5 9/16 TO 6 1/2	1 1/2	1/2	1 1/2 X 1
6 9/16 TO 7 1/2	1 3/4	5/8	1 3/4 X 1 1/4
7 9/16 TO 8 15/16	2	3/4	2 X 1 1/2
9.0 TO 10 15/16	2 1/2	7/8	2 1/2 X 1 3/4
11.0 TO 12 15/16	3	1	3 X 2
13.0 TO 14 15/16	3 1/2	1 1/4	3 1/2 X 2 1/2
15.0 TO 17 15/16	4	1 1/2	4 X 3
18.0 TO 21	5	1 3/4	5 X 3 1/2

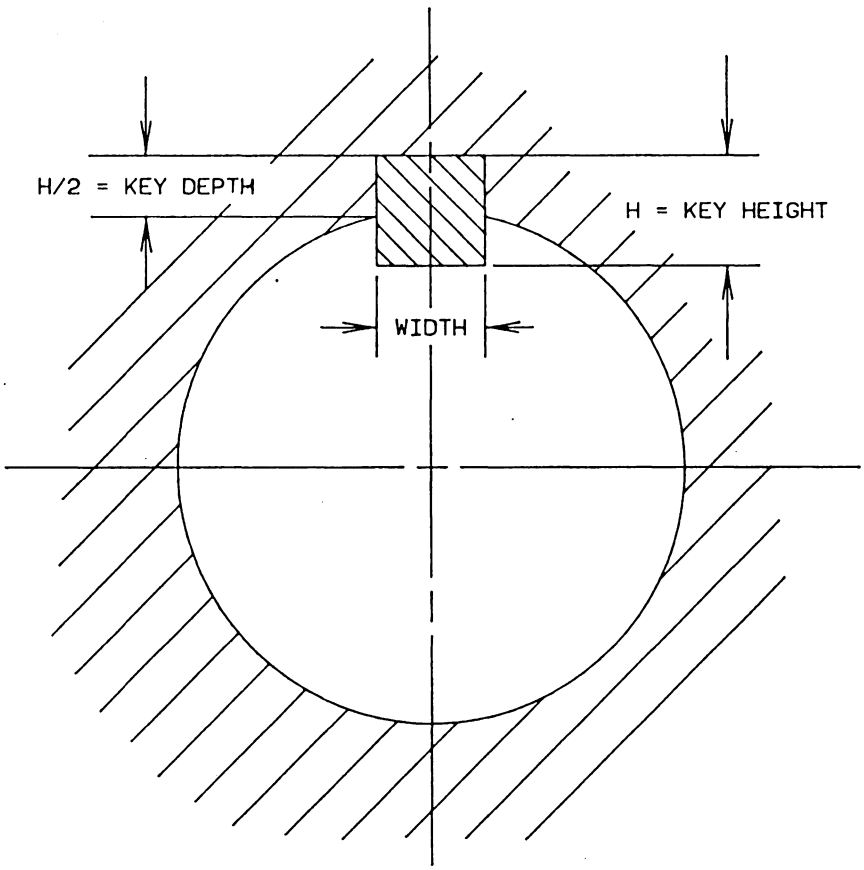


Figure 17. Standard Keyway Dimensions

TABLE 2: EFFECT OF KEYWAYS [11]

NUMBER OF KEYWAYS	COMPARISON WITH STRENGTH OF SHAFT WITHOUT KEYWAYS
1	85%
2	78%
3	73%
4	70%

The program used does not take into account the type of material from which the blank was made. Therefore, the latter method of calculating a diameter was chosen to follow a conservative line of design.

Now, knowing both the hub diameter and the hub wall thickness, a comparison is made between the hub diameter D_H and twice the hub wall thickness U plus the bore diameter G. The larger of these two becomes the new hub diameter.

Also suggested by AGMA [11] was the following:

hub diameters up to 8 inches take
    to the next highest eighth inch,
hub diameters between 8 and 16 inches take
    to the next highest quarter inch,
hub diameters over 16 inches take
    to the next highest half inch.

Once the actual hub diameter is found and a final geometry check against the root diameter is made and allowed, the hub data coordinates (see Figure 16) can be calculated using:

$$x_p = R_p \cos\theta \tag{18}$$

$$y_p = R_p \sin\theta \tag{19}$$

where $0 \leq \theta < 360.$

CHAPTER 5 COMPUTER-AIDED DESIGN SOFTWARE PACKAGES

5.1 INTRODUCTION

At this point, a single, external, spur gear tooth profile exists, along with a face width, and gear blank data. This chapter presents the three major CAD/CAM software packages; GKS, CADAM-CADCD, and MOVIE.BYU, that were used to display the entire spur gear geometry on the screen. The CAD systems provide an interface to high level languages such as FORTRAN. This allows for the applications program to be written for drawing, design, and analysis purposes. Figure 18 pictorially represents the graphics program architecture, while Figure 19 shows a simplified flow chart of how the user written program is used to file the geometry into the CAD systems.

5.2 GRAPHICAL KERNEL SYSTEM (GKS)

5.2.1 INTRODUCTION

Standards in computer graphics have taken a long time to develop, unlike standards in programming languages, like FORTRAN, which were common early in the computer's history. In 1982, however, the International Standards Organization (ISO) accepted the Graphical Kernel System (GKS) as a draft international standard and in 1985 as an international standard.

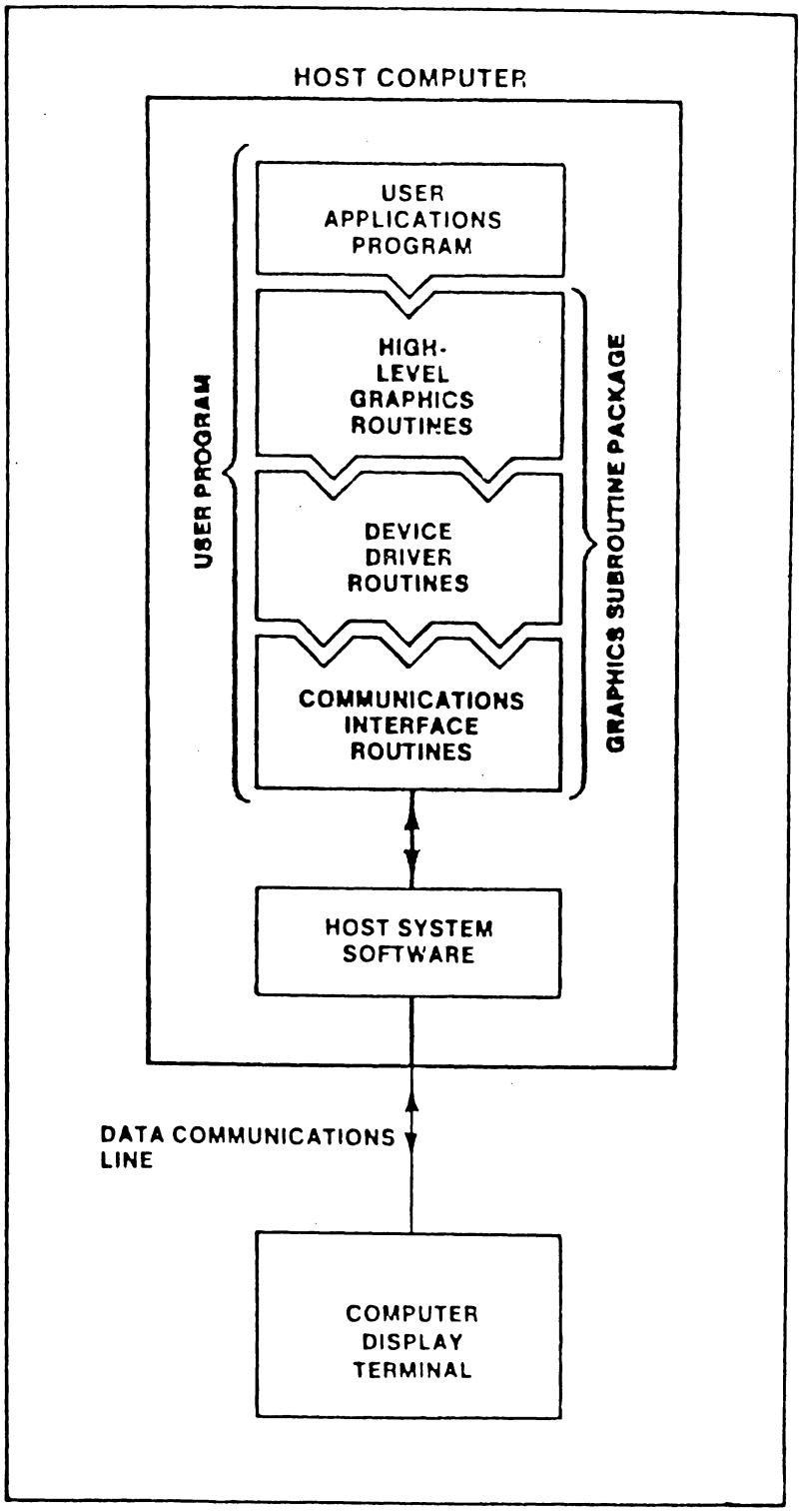


Figure 18. Graphics Program Architecture [35]

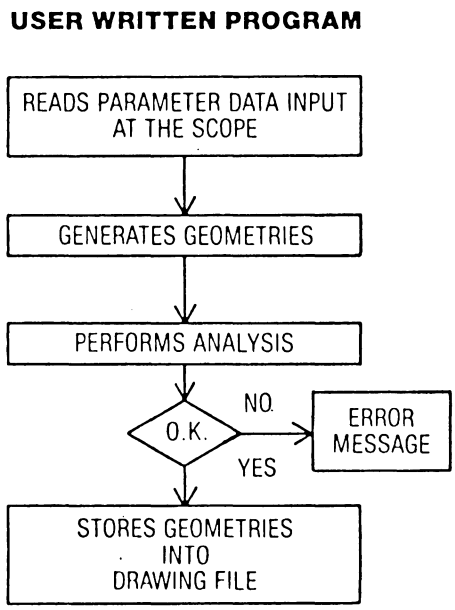


Figure 19. Sample Flowchart of User Written Program [39]

GKS is a 2-D graphics system which is defined independently of programming languages and can be utilized on many different graphics devices (device independent). Its main purpose is the production and manipulation of graphics from simple line graphs to engineering drawings. As long as the part can be mathematically described in the cartesian coordinate system, GKS can draw the part at any window size set by the user.

The functions which can be performed by GKS are divided into nine categories: (1) Control, (2) Primitives, (3) Attributes, (4) Inquiries, (5) Transformations, (6) Segments, (7) Graphic Input, (8) Errors/Utilities, and (9) Metafiles. Only the categories and those functions within those categories which were used by the GKS subroutine in Appendix C will be covered in this chapter. For further instruction and information about GKS, the reader should consult the Hopgood, Duce, Gallop, and Sutcliffe text [33], the Harris text [34], or the PLOT10 GKS TEK Reference Guide [35]. The basic structure of the GKS subroutine is shown in Figure 20.

5.2.2 GKS CONTROL ROUTINES

GKS control routines must be set before any other functions can be used. These routines control GKS and the workstation.

The user initializes GKS at the start of the subroutine by the function:

```
call GOPKS (ERRFIL)
```

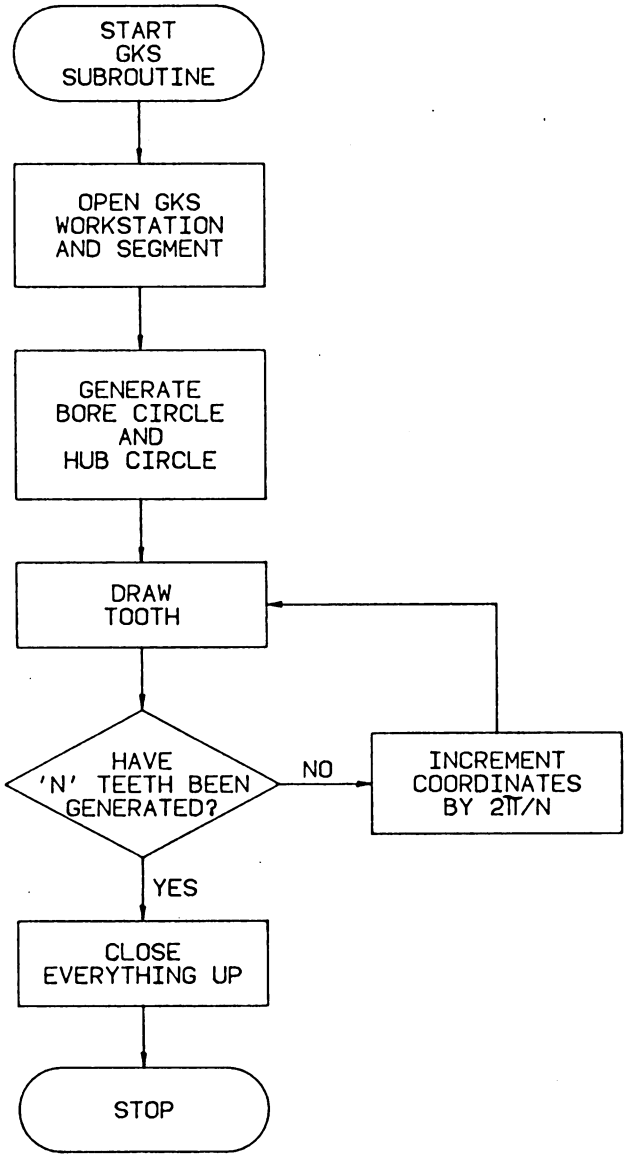


Figure 20. GKS Subroutine Structure to Generate Front View of Gear

where ERRFIL = the name of a file (integer)
used by GKS to return error
information to the application program.

ERRFIL is a parameter which is host dependent and GOPKS must be the first GKS routine called. Opening GKS will set a number of variables to their default values.

And just as GKS must be opened and initialized before any other function can be called, it must also be closed at the end of the GKS session. This is done by the call:

```
call GCLKS
```

There are no parameters with this call and it must be the last GKS routine called. The error file is closed and any other necessary functions will be performed by this routine.

5.2.3 WORKSTATION CONTROL ROUTINES

A workstation provides the interface between the application program and the physical device being used. The categories and graphics device are specified by a workstation type. The workstation control consists of four calls: (1) open workstation, (2) close workstation, (3) activate workstation, and (4) deactivate workstation.

There is a strict order relating to the sequence of these functions along with the opening and closing routines of GKS. GKS precisely defines which GKS function can be called while the program is in a certain operating state.

Below are the four functions and their explanations.

Open Workstation:

```
call GOPWK (WKID,CONID,WTYPE)
     where WKID  = workstation identifier (integer)
           CONID = connection identifier (integer), site dependent
           WTYPE = workstation type (integer), dependent
                   on manufacturer
```

Close Workstation:

```
call GCLWK (WKID)
     where WKID  = workstation identifier (integer)
```

Activate Workstation:

```
call GACWK (WKID)
     where WKID  = workstation identifier (integer)
```

Deactivate Workstation:

```
call GDAWK (WKID)
```

where WKID = workstation identifier (integer)

An example of the sequence is shown below.

```
call GOPKS (ERRFIL)
call GOPWK (WKID,CONID,410700)
call GACWK (WKID)

other GKS functions and FORTRAN programming

call GDAWK (WKID)
call GCLWK (WKID)
call GCLKS
```

5.2.4 PRIMITIVES

Drawings or models in GKS are constructed from four basic primitives. The four primitives are:

(1) polyline, which draws a line segment from point to point in a sequence specified by the user

(2) polymarker, which places a marker at the points specified with a symbol

(3) fill area, which fills an area of specified shape with a certain pattern

(4) text, which draws a string of characters.

Each of the four primitives has associated with it a set of parameters and attributes. The parameters define the form of that primitive, the attributes describe the appearance of the primitives.

Polyline was the only primitive used by the program and, as mentioned, will be the only one discussed.

Polyline is the main line drawing primitive of GKS. A set of connected line segments are used to draw the shape desired. If a curved surface is needed in GKS, the user must realize that the shorter the line segments, or closer the points specified, the more accurate the model will be. To generate a polyline, the call is:

```
call GPL (N,XPTS,YPTS)
     where N          = number of points in the polyline
                        array (integer)  2 < N < 1000
           XPTS, YPTS = array of x and y coordinates (real)
```

The polyline consists of N-1 line segments constructed from N points with their cartesian coordinates listed in the arrays XPTS and YPTS.

5.2.5 ATTRIBUTES

Attributes associated with polyline are linetype, linewidth, and color. These attributes are used to differentiate one polyline from another, if the need occurs. Fortunately, the default values set when opening GKS work very nicely for the display of the gear. A change using the special attribute routines is not necessary.

5.2.6 INQUIRY FUNCTIONS

Inquiry functions in GKS allow a user to access information contained in the various GKS state lists. A state list contains information about the current state of GKS. The inquiry function will help in setting up GKS, debugging errors, or writing library routines.

The inquiry function used in this program helps to set up the GKS workstation. The function has the form:

```
call GQEWK (N,ERRIND,NUMBER,WKTYPE)
     where N      = list element of workstation types (integer)
           ERRIND = error indicator (integer)
                    0  - requested value reported
                    >0 - refer to error message booklet
           NUMBER = number of available workstation types (integer)
           WKTYPE = workstation type of element (integer)
```

GQEWK is used because GKS requires the list element of available workstation types so a validation can be made between its list of possible workstations and the device the designer is using. If a zero is returned, the program will continue; if a number other than zero is returned, the program loops until a valid device is entered.

5.2.7 TRANSFORMATIONS

GKS transformations are used to map graphic output from one coordinate system to another. There are three coordinate systems used in GKS: (1) world coordinate space, (2) normalized device coordinate space, and (3) device coordinate space. And there are two transformations: (1) normalization and (2) workstation.

The world coordinate system and the normalization transformation are used in the program. The term world coordinate system is used in GKS to define a cartesian coordinate system used to present the graphic output to GKS. The normalization transformation is the first GKS transformation applied to the graphic output.

To transform the correct world coordinate system to the screen or window of the graphic device only two functions were needed. The first sets the normalization transformation window size in the x and y direction by specifying the minimum and maximum of each. This is done by:

```
call GSWN (TNR,XMIN,XMAX,YMIN,YMAX)
```

where TNR = transformation identifier (integer)

XMIN, XMAX = x - coordinate extents (real)

YMIN, YMAX = y - coordinate extents (real)

Then, to select the normalization transformation as specified the call:

```
call GSELNT (TNR)
```

where TNR = transformation identifier (integer)

is needed.

The window size used in the program varies with the size of the gear being designed. This allows the gear to always fit on the screen.

5.2.8 SEGMENTS

A segment is a collection of output primitives which can be manipulated as a whole. They are used as storage mechanisms in GKS because of the need to move drawings, transform them, or display them at different places on the screen when building a picture. A segment is created by the function:

```
call GCRSG (SGNA)
```

where SGNA = segment name (integer)

Once the segment is open, any graphics output will be a part of this segment until the close segment function is encountered. A segment is closed by:

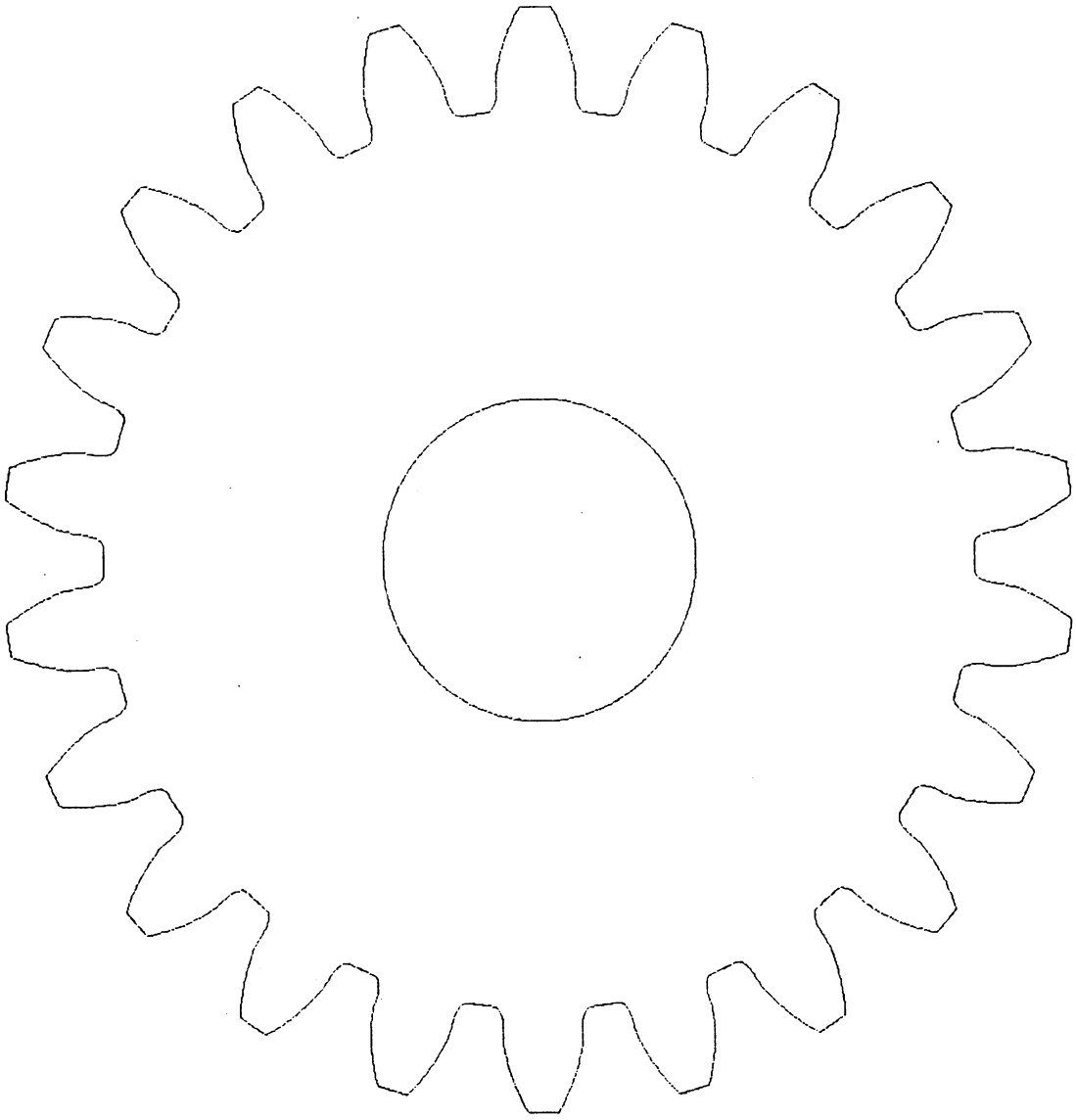
```
call GCLSG    (no parameters)
```

It is important to note that only one segment can be open at any given time.

5.2.9 THE GKS SUBROUTINE AND EXAMPLES

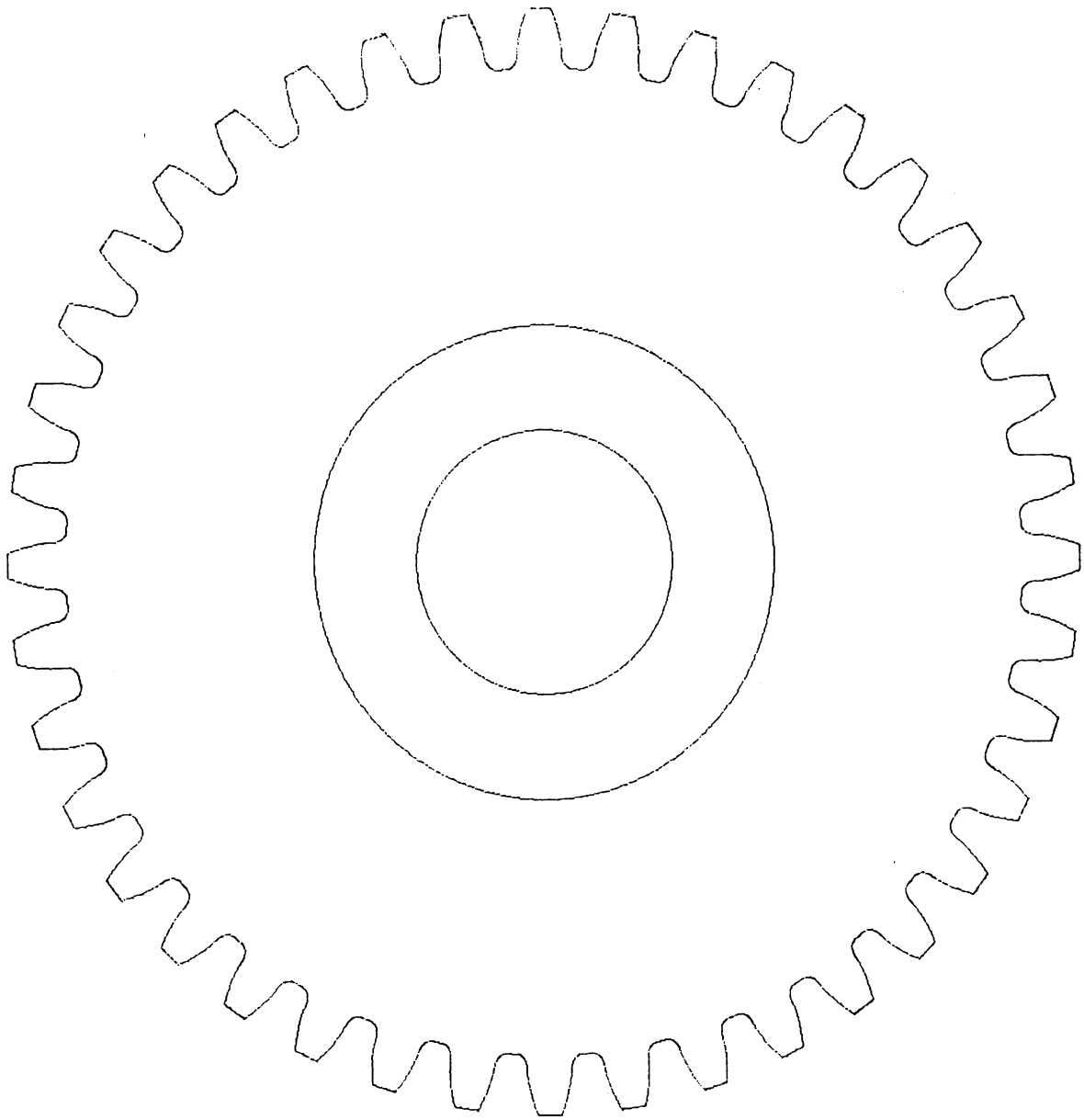
The GKS subroutine opens everything up, including only one segment, draws the entire gear by incrementing the tooth data around a circle using $2\pi/N$ increments, draws the gear blank geometry, and then closes the segment. Typical gear drawings generated by the program are shown in Figures 21, 22, and 23 along with a listing of the input parameters. Figure 21 shows a 2-D gear with 22 teeth, no hub, and cut with a 20 degree pressure angle hob. Figure 22 shows a gear with 40 teeth and an existing hub. Figure 23 demonstrates the shape of an undercut gear having only 10 teeth.

At a later time, once the drawing is established, the gears could be used as part of an overall machine. The drawing can be placed on the screen where necessary to help the designer visualize the entire mechanism. This idea will be discussed further in the concluding chapter using CADAM.



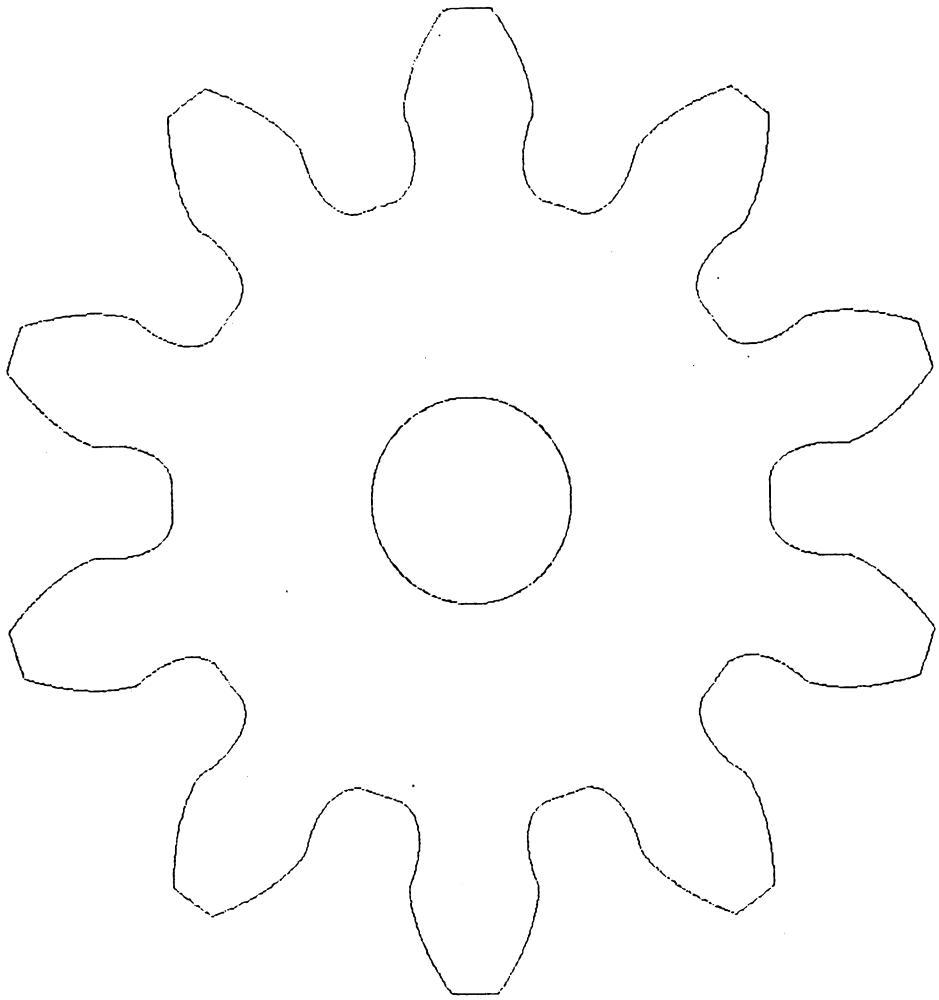
DIAMETRAL PITCH	= 1
ADDENDUM FRACTION	= 1
DEDENDUM FRACTION	= 1.25
HOB TIP RADIUS	= 0
NUMBER OF TEETH	= 22
IS THE GEAR STANDARD?	= YES
CUTTING PRESSURE ANGLE	= 20
INTERVALS PER CURVE	= 25
TYPE OF GEAR BLANK	= PLAIN
FACE WIDTH	= 3
IS THERE A HUB?	= NO
BORE DIAMETER	= 7

Figure 21. A 22 Tooth Gear with No Hub - GKS



DIAMETRAL PITCH	= 1
ADDENDUM FRACTION	= 1
DEDENDUM FRACTION	= 1.25
HOB TIP RADIUS	= 0.35
NUMBER OF TEETH	= 40
IS THE GEAR STANDARD?	= YES
CUTTING PRESSURE ANGLE	= 14.5
INTERVALS PER CURVE	= 30
TYPE OF GEAR BLANK	= PLAIN
FACE WIDTH	= 4
IS THEPE A HUB?	= YES
HUB ON HOW MANY SIDES?	= 2
HUB PROTRUSION DISTANCE	= 4
BORE DIAMETER	= 10

Figure 22. A 40 Tooth Gear with a Hub - GKS



DIAMETRAL PITCH	= 1
ADDENDUM FRACTION	= 1
DEDENDUM FRACTION	= 1.25
HOB TIP RADIUS	= 0.1
NUMBER OF TEETH	= 10
IS THE GEAR STANDARD?	= NO
CUTTING PITCH RADIUS	= 5
CUTTING PRESSURE ANGLE	= 20
TOOTH THICKNESS	=1.570796
INTERVALS PER CURVE	= 50
TYPE OF GEAR BLANK	= PLAIN
FACE WIDTH	= 4
IS THERE A HUB?	= NO
BORE DIAMETER	= 2.5

Figure 23. An Undercut 10 Tooth Gear without a Hub - GKS

5.3 COMPUTER-AUGMENTED DESIGN AND MANUFACTURING (CADAM-CADCD)

5.3.1 INTRODUCTION

Computer-graphics Augmented Design and Manufacturing (CADAM) is a set of computer programs used to generate mechanical drawings and 3-D design models and solve a variety of engineering problems. Developed by Lockheed Aircraft Corporation in the middle 1960's, CADAM continues to be developed and updated. It is used by many companies throughout the world, including IBM. If a more complete overview or further instruction is desired, the information can be found in the IBM manuals [36-39].

CADAM uses a common central data base for the CAD and CAM systems. The CAD (Computer-Aided Design) portion of CADAM is a high performance multi-function design and drafting system. The CAM (Computer-Aided Manufacturing) portion automatically generates numerical control (NC) data. Figure 24 pictorially demonstrates a product development process and where CADAM can be applied.

Features in CADAM include:

(1) 2-D and 3-D geomtric constructions

(2) Display and data management

(3) Calculation and analysis functions (e.g. section and mass properties, volume, weight, moments of inertia)

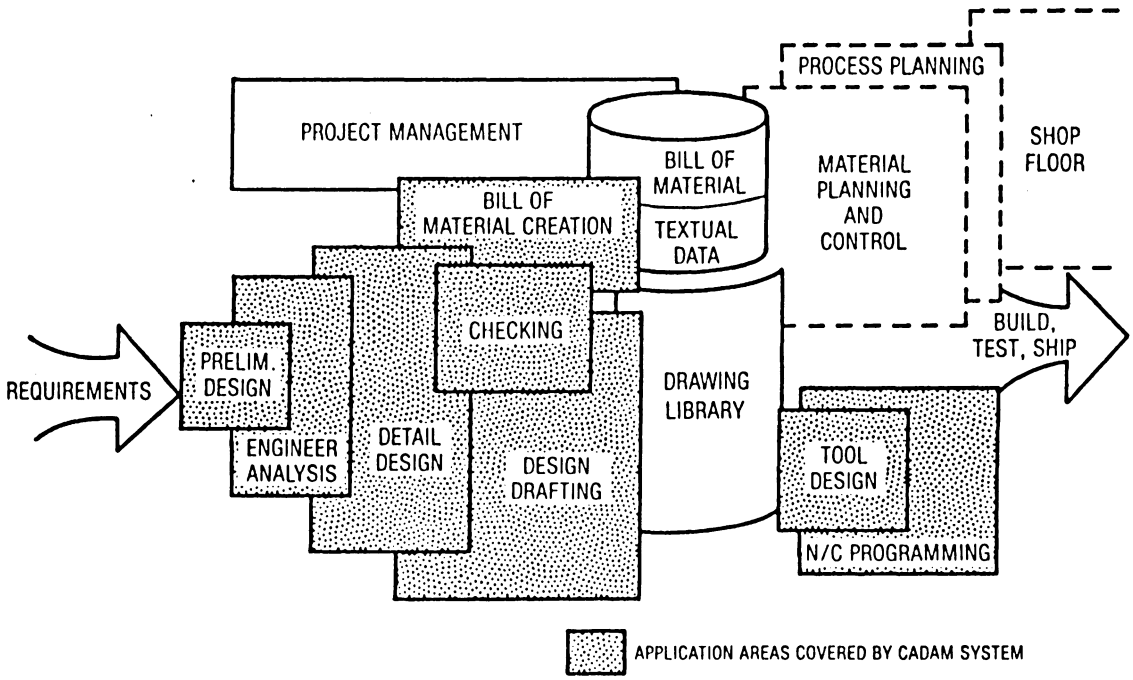


Figure 24. Product Development Process [39]

(4) Numerical control facilities and an Automatically Programmed Tool (APT) interface

(5) Finite Element Models (FEM) interface with the 3-D mesh geometry module

(6) A geometry interface module

(7) A bill of materials interface

(8) A technical publications interface - document composition facility

(9) Various hardcopy formats

(10) Macrogeometry - similar parts design.

Figure 25 shows the CADAM system interface capabilities.

The geometry interface module is the feature used by the gear design program. The flow of data from the interface module to the file is shown in Figure 26. The geometry interface is designed to provide the user with the ability to access and modify the CADAM data base. This can be done by creation and retrieval. Creation creates CADAM elements (points, lines, circles, ellipses, splines) through the applications program. Retrieval retrieves a model from the drawing file.

The four functions making up the Geometry Interface module are:

(1) CADCD - a collection of subroutines driven by call statements to produce CADAM models

(2) CADET - a collection of call statements which drive passive subroutines to receive disassembled CADAM elements (details, etc.)

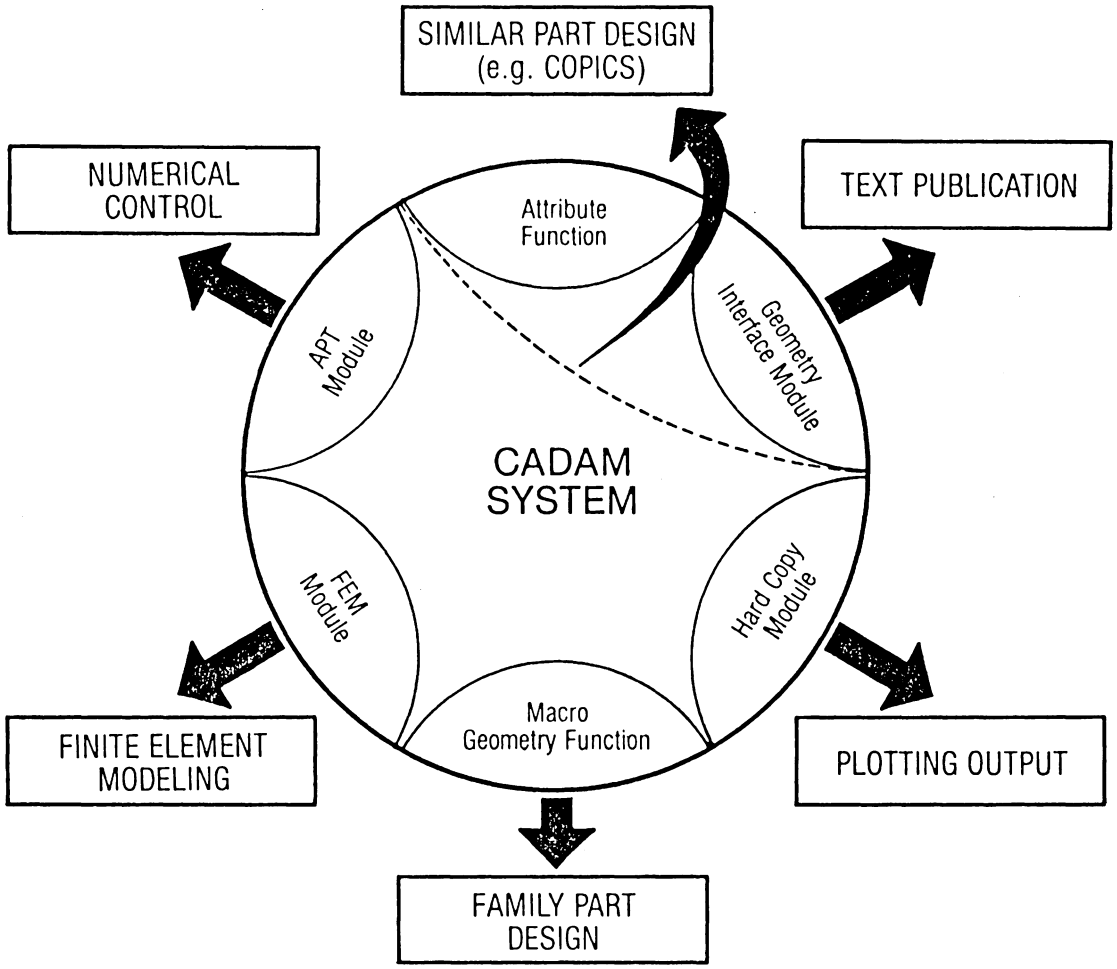


Figure 25. CADAM System Interface Capability [39]

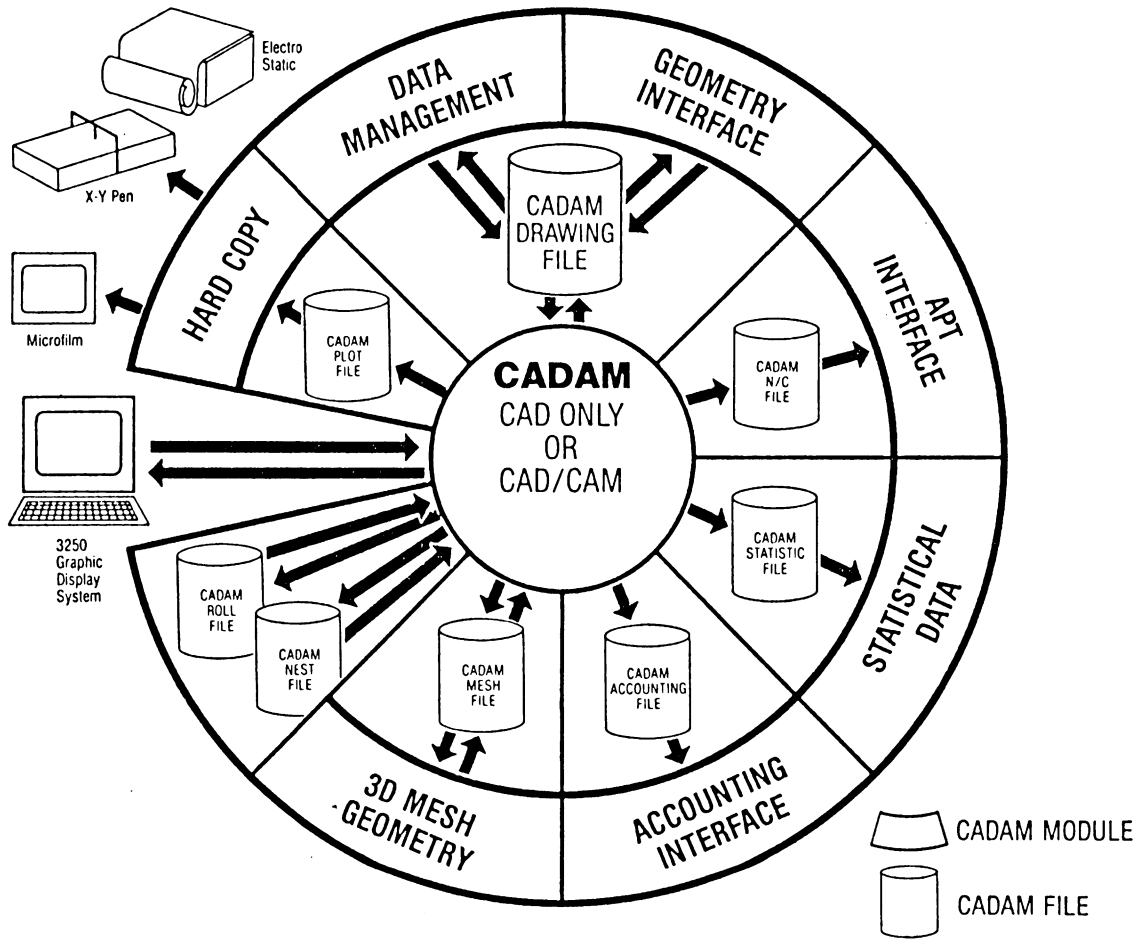


Figure 26. Interrelationship between Modules and Files [39]

(3) CADMACGM - a collection of subroutines which are driven by call statements to produce CADAM elements interactively

(4) SCLIB - a program that generates a listing of subgroups or drawing IDs.

CADCD is the set of utility programs acting interactively with FORTRAN that is used to draw the gear to the CADAM data base. The elements are created by a batch process rather than creating elements at the scope. CADCD will allow the user to place new models into the CADAM data base or make additions or modifications to existing drawings.

The subroutines used in this program are CADST, CADFIL, and BEGVU which are CADCD utility routines; and PT3D, LINE3D, C3DCRN, ARC3D, and SPLINE which are CADCD geometry generation routines.

5.3.2 CADCD UTILITY ROUTINES

If a new drawing is being created CADST must be used. If an existing drawing is being modified, CADST and CADFIL must be used. CADST will initialize a model by the routine:

```
call CADST (DRAWID,USERID,GROUP)
     where DRAWID = a 20 character array containing the drawing ID
           USERID = a 8 character array containing the user ID
           GROUP  = a 4 character array containing the Group name.
```

BEGVU, which is necessary because it begins a trap for the view geometry, is called by:

```
call BEGVU (IOPT,IDVU,XYZ,XVECTR,YVECTR,ZVECTR,*E)
     where IOPT   = an option flag which must be set to 1
           IDVU   = a 4 character array of which the last two
                    characters are used to contain the view ID
                    (e.g. "XXIS" for isometric view)
           XYZ    = 3-D array, XYZ(3), which represents the
                    x, y, and z locations of the view
           XVECTR = 3-D array, XVECTR(3), which gives the
                    direction cosines for the x-axis
           YVECTR = 3-D array, YVECTR(3), which gives the
                    direction cosines for the y-axis
           ZVECTR = 3-D array, ZVECTR(3), which gives the
                    direction cosines for the z-axis
           *E     = error.  A view is already started, or
                    no more space for views.
```

The last utility routine CADFIL will file a drawing by calling:

```
call CADFIL (IOPT,NOGOOD,IDUMMY)
     where IOPT    = 0 - file a drawing
                     1 - retieve a drawing
                     2 - overfile a drawing
```

NOGOOD = an integer to be tested for after filing,
if not = 1, the file call is no good

IDUMMY = a dummy variable for later use

5.3.3 CADCD GEOMETRY ROUTINES

As for the geometry generation drawing routines, both 2-D and 3-D routines are available with CADCD. But, with GKS available in 2-D, only the 3-D routines are programmed and used.

To create a 3-D point use:

```
call PT3D (XYZ,*E)
     where XYZ = a 3-D array, XYZ(3), containing the
                 x, y, and z coordinates of the point
           *E  = error, not as yet specified
```

To create a 3-D line the following subroutine is used:

```
call LINE3D (XYZ1,XYZ2,*E)
     where XYZ1 = a 3-D array containing the x, y, and z
                  absolute coordinates of one of the
                  end points
```

XYZ2 = a 3-D array containing the x, y, and z absolute coordinates of the other end point

*E = error return indicating the length of line is less than tolerance value.

Lines are used to connect the faces of the gear or the face of the hub to the face of the gear. This will give surfaces a contour and the user a better perception of depth.

The 3-D circles are written to the data base as follows, using the center point, radius, and normal vector as parameters:

call C3DCRN (CENTER,RADIUS,CIRNRM,*E)

where CENTER = An array of length 3 that contains the x, y, and z coordinates of the center of the circle

RADIUS = The absolute length of the radius of the desired circle

CIRNRM = An array of length 3 that contains the coefficients of a vector normal to the plane of the circle

*E = Error return resulting from a radius too small.

The circles are used for the bore circles, and if a hub exists, the hub circles of the radius specified by the program.

The 3-D arcs, used at the root land and top land or root circle and addendum circle of the gear, are created with:

```
call ARC3D (ICODE,CENTER,XYZ1,XYZ2,CIRNRM,*E)
     where ICODE  = Not used at present, reserved for future use
           CENTER = An array of length 3 that contains the
                    x, y, and z absolute coordinates of the
                    center of the circle in which the arc is
                    contained
           XYZ1   = A 3-D array containing the x, y, and
                    z absolute coordinates of one of the
                    end points of the arc
           XYZ2   = A 3-D array containing the x, y, and
                    z absolute coordinates of the other
                    end point of the arc
           CIRNRM = An array of length 3 that contains the
                    A, B, and C coefficients of a vector
                    normal to the plane of the circle
           *E     = An error return indicating the resulting
                    radius is less than a tolerance value or the
                    distance between the two end points is less
                    than the tolerance value.
```

The last and most complex of the callable routines creates the 3-D spline with:

```
call SPLINE (NN,M,K,D,V,S,T1,T2,T3,*E)
```

where NN = integer value representing the number of input points

M = integer value specifying the number of dimensions in space, dimension of coordinates input through the D array

K = Integer value specifying end conditions

0 - curved end slopes unknown

1 - first slope supplied

2 - last slope supplied

3 - first and last slope supplied

D = A floating point array containing spline points coordinate data

V = A floating point array of spline tangent vector components at input points

S = A floating point array of chord lengths defined by input data

T1 = Temporary floating point working area must be of length NN

T2 = Temporary floating point working area must be of length NN

T3 = Temporary floating point working area must be of length NN

*E = Error

The splines are used to draw the involute curve and the trochoid fillet to the CADAM data base.

5.3.4 EXAMPLES FROM THE CADAM-CADCD SUBROUTINE

All of the data from the geometry section of the program is passed to the CADAM subroutine shown in Appendix D and manipulated, according to the structure in Figure 27, such that it draws a complete gear to the data base using the routines listed. Figures 28, 29, and 30 demonstrate the use of CADAM-CADCD in association with the user written program. Figure 28 shows a 22 tooth gear with no hub and cut with a 20 degree pressure angle hob. Figure 29 shows a 35 tooth gear with a hub on one side cut with a 20 degree pressue angle hob. And to demonstrate an undercut gear and its geometry, Figure 30 shows a gear with 10 teeth and a hub on both sides.

Manual manipulations, which can be incorporated as part of a 3-D or 2-D design, can be done on the gear as well. Figure 31 shows the same gear as in Figure 30, however, a manual hidden line elimination has been performed and a keyway has been added.

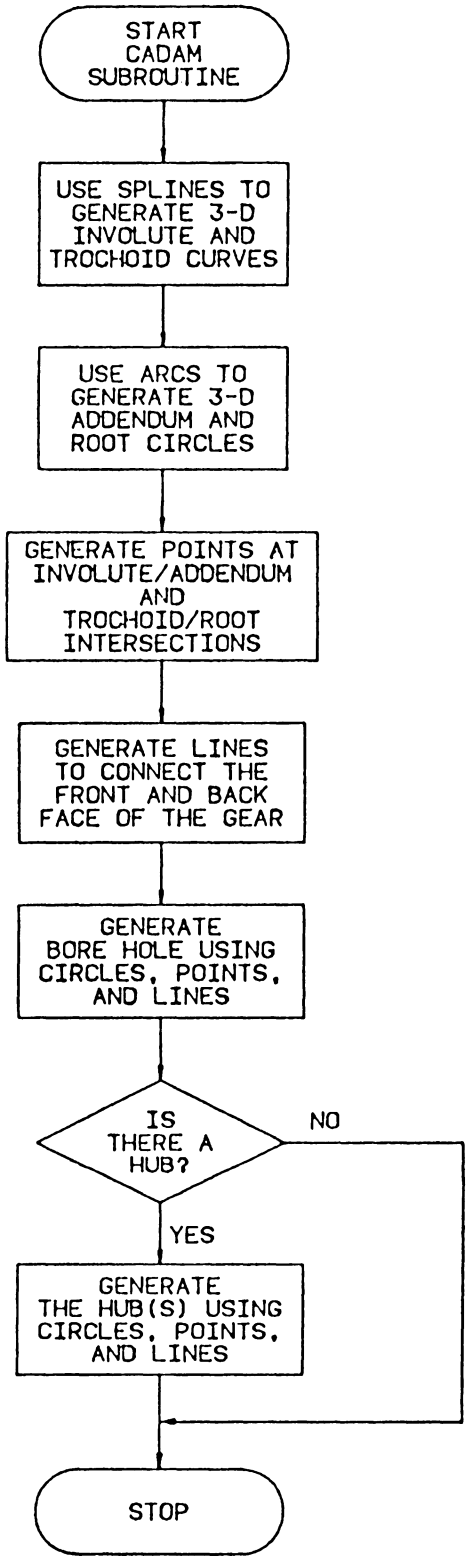


Figure 27. CADAM-CADCD Subroutine Structure to Generate Gear

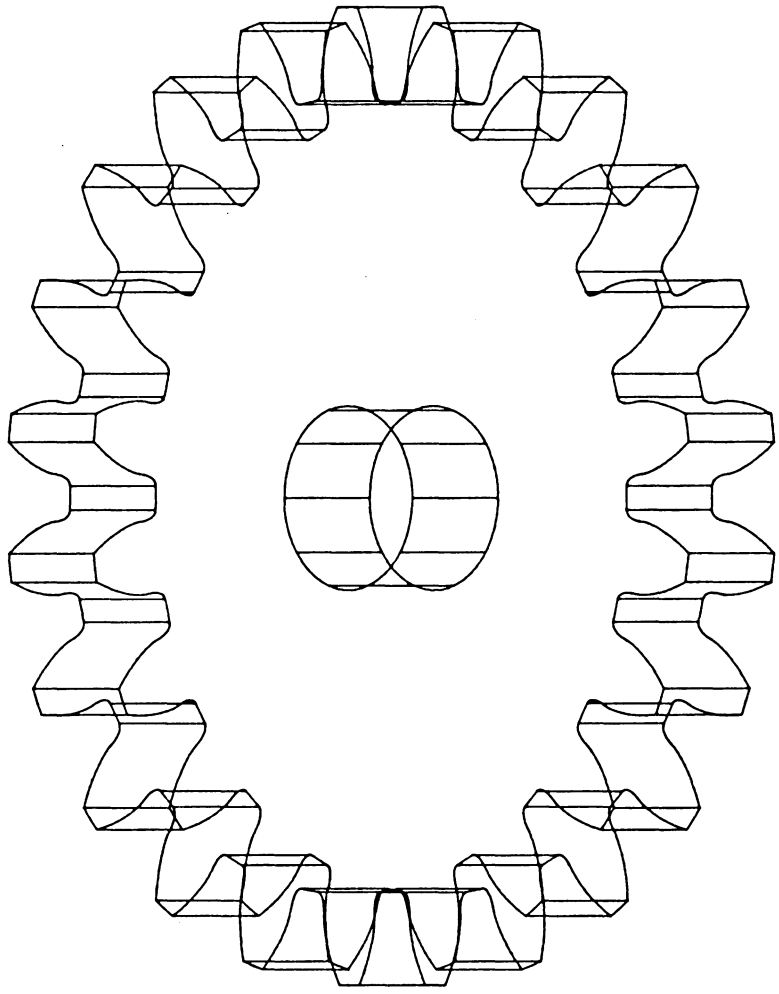


Figure 28. A 22 Tooth Gear with No Hub - CADAM

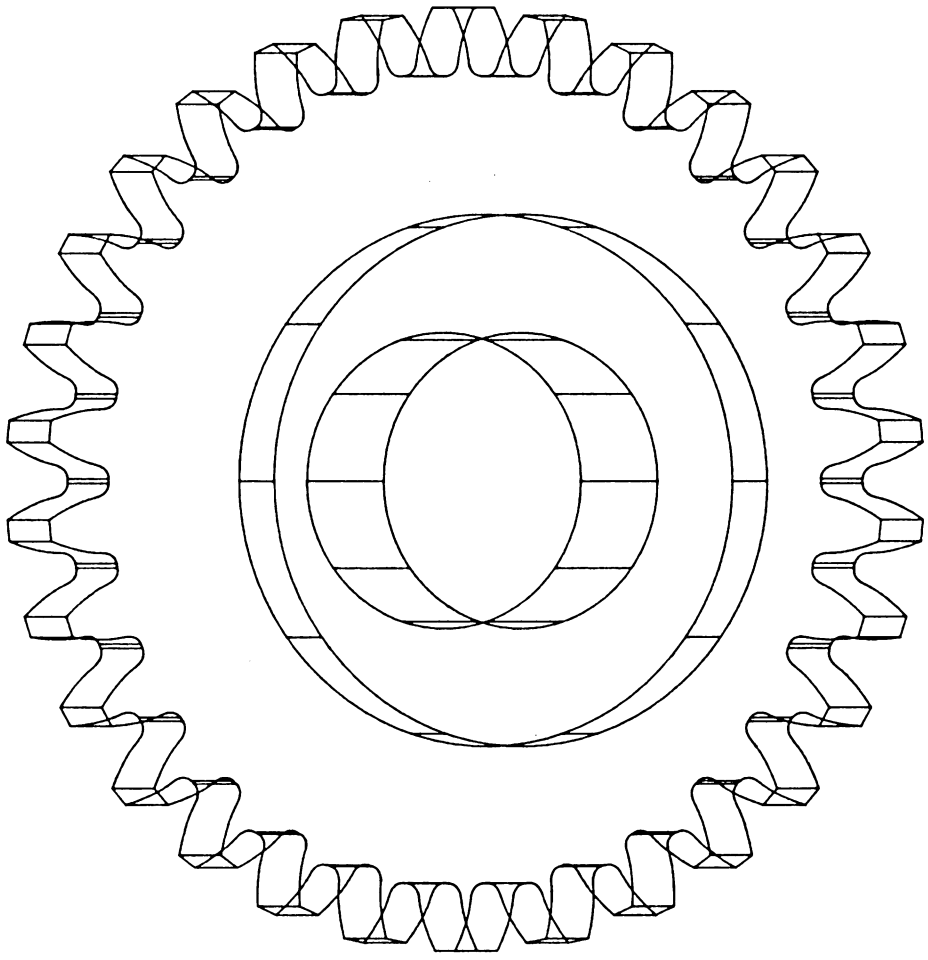


Figure 29. A 30 Tooth Gear with a Hub on One Side - CADAM

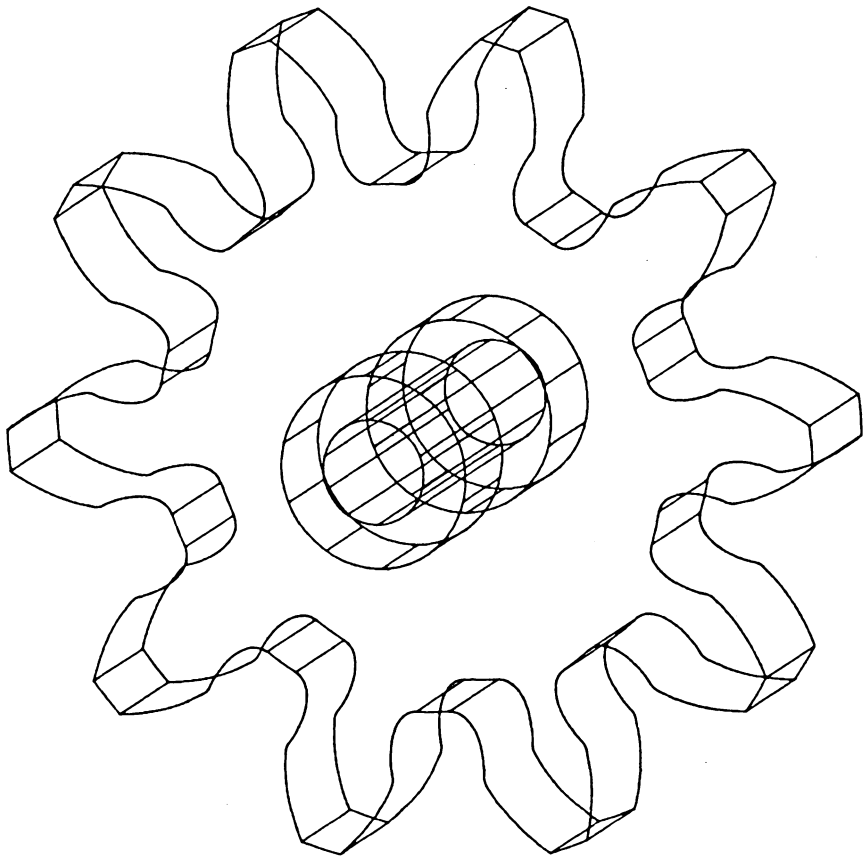


Figure 30. A 10 Tooth Gear with a Hub on Both Sides - CADAM

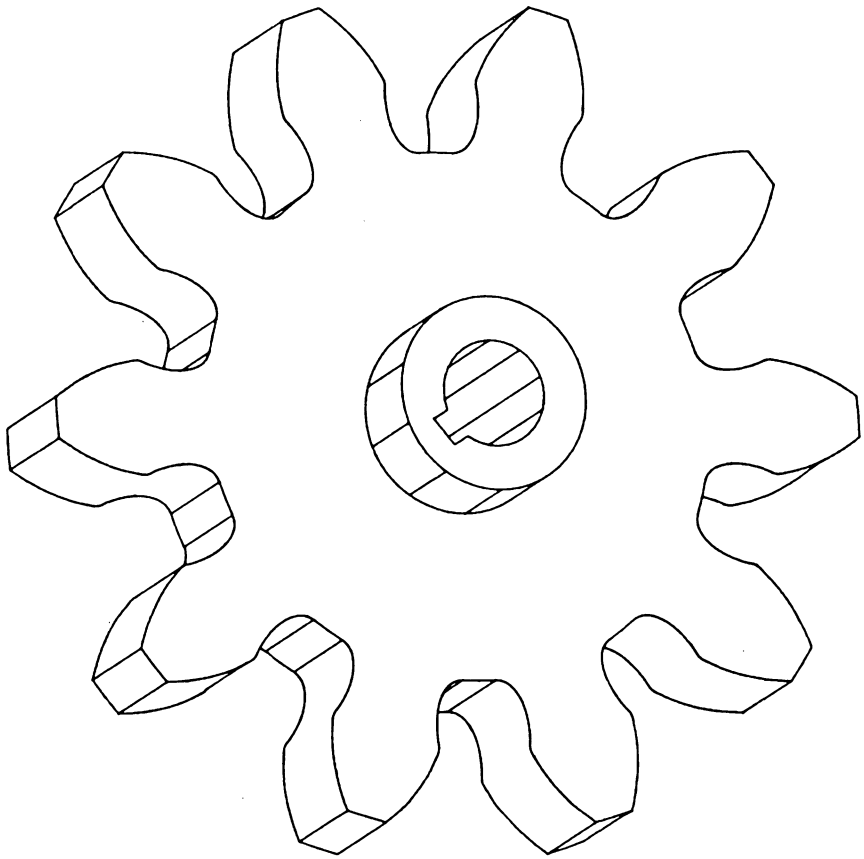


Figure 31. Example of Manual Hidden Line Removal - CADAM

5.4 MOVIE.BYU

5.4.1 INTRODUCTION

The final FORTRAN subroutine written to model and analyze the gear used MOVIE.BYU. MOVIE.BYU is a general purpose computer graphics system which allows for the modeling and analyzing of 3-D objects (panel and solid). It also provides still frame animation capabilities of the model.

MOVIE.BYU consists of seven FORTRAN programs which manipulate and display the geometry data in terms of polygonal elements, solid elements, or contour lines. The seven FORTRAN programs are Display, Utility, Section, Title, Update, Compose, and Mosaic. How they relate to the data base is shown in Figure 32. A short synopsis of each program is given below, but further information on MOVIE.BYU can be found in the MOVIE.BYU Document [40].

5.4.2 MOVIE.BYU FORTRAN PROGRAMS

Display is the heart of the system. It is an interactive program which displays and animates existing polygonal element models. It allows these models to be rotated, translated, and viewed on the screen with or without automatic hidden line elimination. It also gives information about the model and enables the user to animate the part.

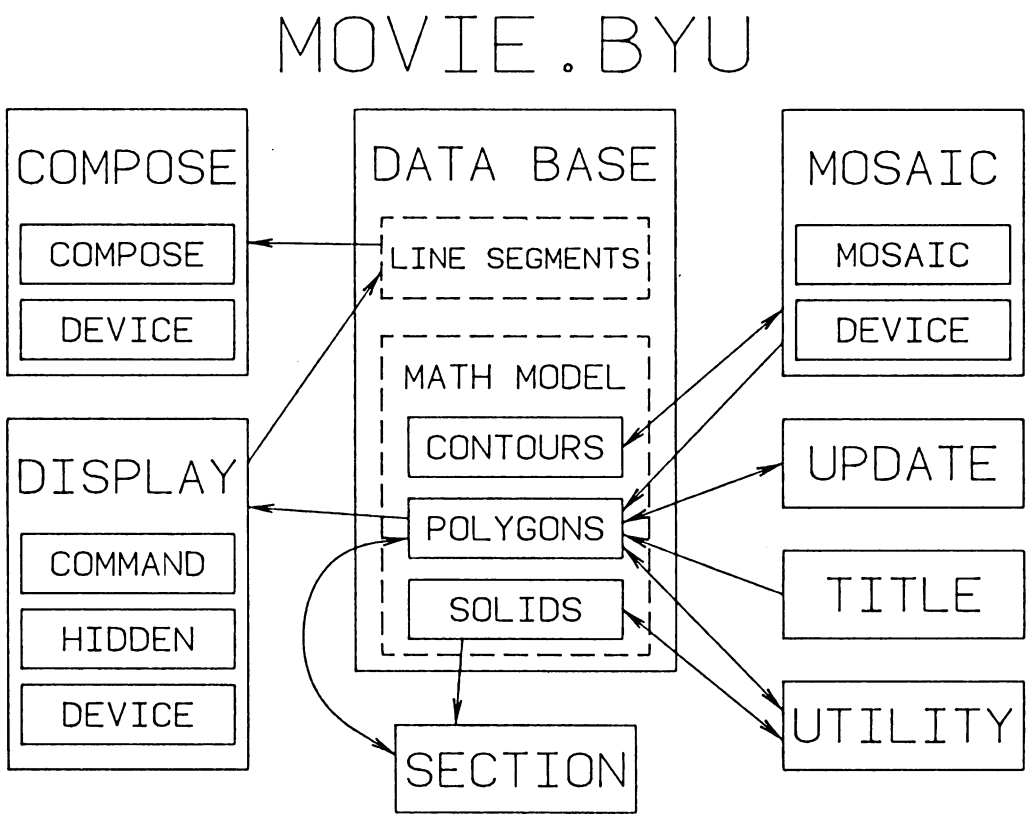


Figure 32. Relation between MOVIE.BYU and the Data Base [41]

Utility is a program for editing or generating data to produce or edit 2-D and 3-D polygonal models of either shell or solid elements. It can read, write, or change data files to create displacement files in a format which is compatible with the other programs in MOVIE.

Section is a special purpose program used to modify solid element data so it is compatible with the Display program.

Title generates 2-D and 3-D characters whose data format is similar to the other programs in MOVIE. Text is converted into standard or offset characters composed of polygons.

Update reads geometry data files done in previous MOVIE versions and converts them to the new format in the newest edition of MOVIE.

Compose is a program which can create multiple image line drawings. The user, after having already enabled the RECORD option in Display, may selectively retrieve saved files in an automatic or manual mode to build the multiple image displays.

Mosaic converts complex surfaces defined by contour lines into polygonal surface definitions. It is also the program which enables contouring.

5.4.3 THE GEAR PROGRAM SUBROUTINE FOR MOVIE

Although MOVIE.BYU is very useful, the gear program itself does not actually use any of the MOVIE programs directly. Instead, the gear program subroutine, shown in Appendix E, does the work of the Utility program. It generates, manipulates, and writes the geometry data to a file in such a way that the format is compatible with Display, see Figure 33. This is exactly what Utility would have done if Utility had been utilized. As it turned out, however, it was easier to write an independent program and use the write sequence from Utility to file the data in a compatible format for the other MOVIE programs. The write sequence used to file the data was as follows:

```
      WRITE(8,1500) NP, NJ, NPT, NCON, NTEST
      WRITE(8,1500) ((NPL(I,IJ),I=1,2),IJ=1,NP)
      WRITE(8,1600) ((COORD(I,IJ),I=1,3),IJ=1,NJ)
      WRITE(8,1500) (IP(I),I=1,NCON)
 1500 FORMAT(16I5)
 1600 FORMAT(6E12.5)
```

where NP = The number of parts used to make the gear

1 - If the no hub option was chosen

2 - If the hub is on one side

3 - If the hub is both sides

NJ = The number of nodes or joints

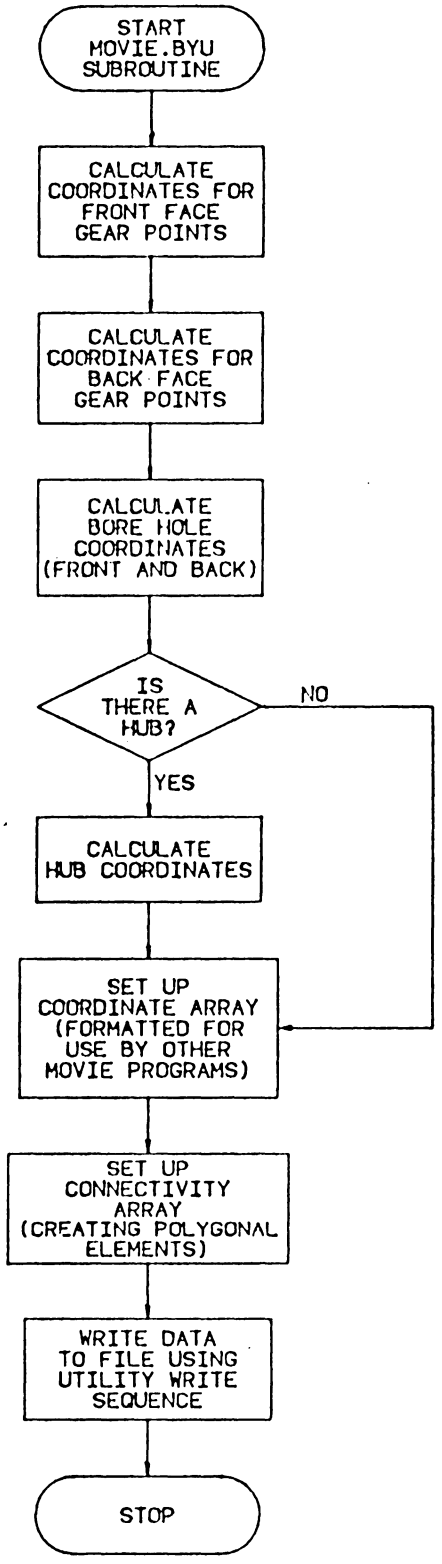


Figure 33. MOVIE.BYU Subroutine Structure to Generate a Gear

NPT = The number of elements or polygons

NCON = The number of entries in the connectivity array

NTEST = A format test variable (must = 0)

NPL = The parts list, contains the element numbers of the lower and upper bounds of the element grouping

COORD = An array of the coordinates of the nodes

IP = The connectivity of the elements or polygons

The geometry file created is found under the name FILE.GEAR.

The subroutine is complex due to the number of variable parameters. The gear and each hub, if a hub exists, are formed as separate parts. Also, each surface of the gear is divided into a certain number of elements determined by the user to ensure that the actual geometry is retained.

5.4.4 USE OF DISPLAY AND PROGRAM/MOVIE.BYU EXAMPLES

Display is accessed independently once a geometry file is created. An example of the sequence of commands to access the file is shown in Figure 34. The information in angular backets, < >, is what Display asks or tells the user and the statements starting with a period are entered by the user interactively.

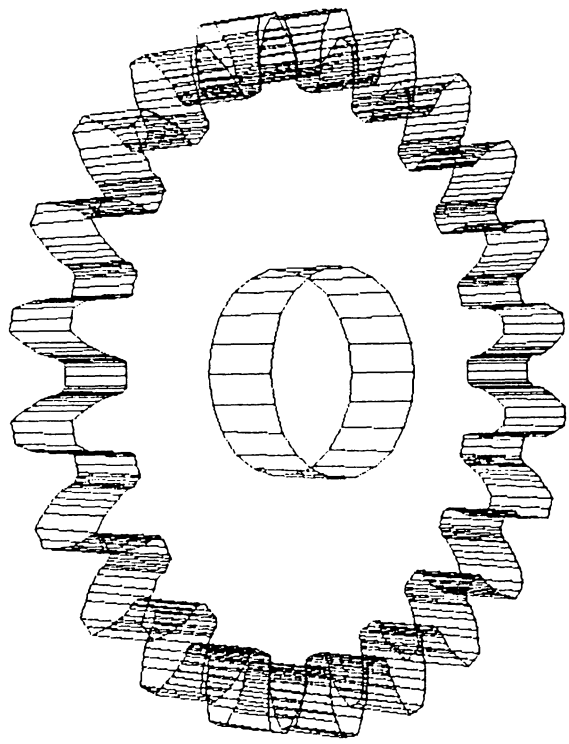
```
START
EXECUTION BEGINS...
<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
<READ:     1 PARTS; 1188 COORDINATES;  638 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
<PREVIOUS RANGE:>
<  -11.923 <X<     11.923   -12.000 <Y<     12.000    -3.000 <Z<      0.000>
<ORIGIN MOVED TO:     0.000     0.000    -1.500>
<DISTANCE TO ORIGIN:  84.00 ,ANGLE: 28.00 ,ZMIN:    0.10 ,ZMAX: 168.00>
< 1 PARTS WITH ELEMENT LIMITS:>
  45 638
>>
.help
ALIAS      DISTANCE   FLAT       MULTIPLE   RESTORE    SMOOTH
ANIMATE    DOTTED     FRINGE     NODE       ROTATE     SUMMARY
CENTER     DRAW       GLASS      PARTS      SAVE       TRANSLATE
COLOR      EXIT       HAZE       PIVOT      SCALE      UNIFORM
CONTOUR    EXPLODE    HELP       POLYGON    SCOPE      VIEW
DASH       FAST       IMMUNE     READ       SHADOW     WARP
DEVICE     FEATURE    LIGHT      RECORD     SHIFT
DIFFUSE    FIELD      LINEAR     RESET      SHRINK
OSEGMENT   CSEGMENT   CLSG       SAVTEK

>>
.draw
>>
.exit
```

Figure 34. Example Sequence of Commands Used to Access File

An explanation of the Display commands on the help list can be found in the MOVIE manuels [40,41]. DRAW and VIEW will display the gear to the screen and will be the only commands explained here. The DRAW command displays a line drawing without the use of Watkin's Algorithm. VIEW draws the model with the use of Watkin's Algorithm. Watkin's Algorithm is an automatic hidden line elimination algorithm that is used in association with VIEW.

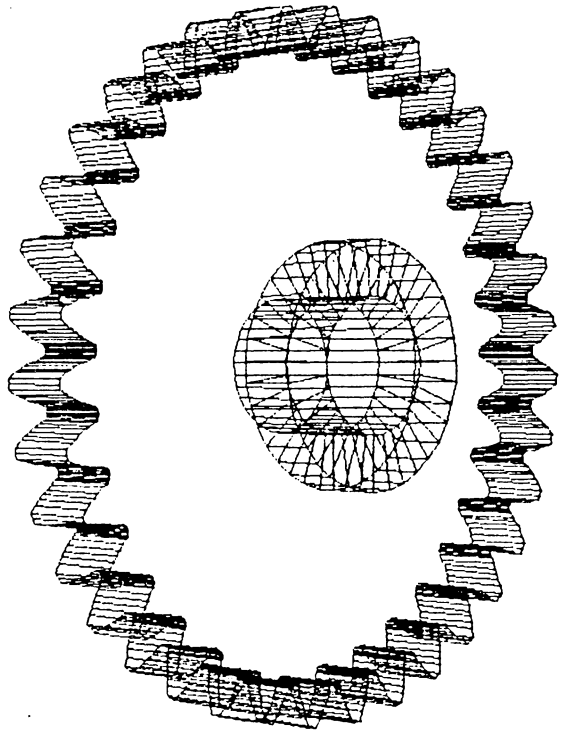
Examples of MOVIE.BYU in use are shown in Figures 35, 36, 37, 38, and 40. Figure 35 is a gear with 22 teeth, no hub, and cut with a 20 degree pressure angle hob. The command sequence used to get the gear in that position is shown as well. Figure 36 shows a 35 tooth gear with a hub on one side cut with a 25 degree pressue angle hob. An undercut gear is shown in Figure 37 with 10 teeth and a hub on both sides.

Manual manipulations in MOVIE.BYU are performed at the scope while using Display. Figure 38 shows a hidden line elimination using Watkin's Algorithm. Many of the hidden lines still exist because of a limitation in MOVIE.BYU, which only allows for an element to have a maximum of 10 sides. Thus, the front and back faces of the gear, which required the polygons to have more than 10 sides, were not used for this picture. If, however, Display is modified, a gear such as the one shown in Figure 39 can be produced. Also, an exploded view of the gear with the two hubs removed is demonstated in Figure 40 along with the commands used to produce it.



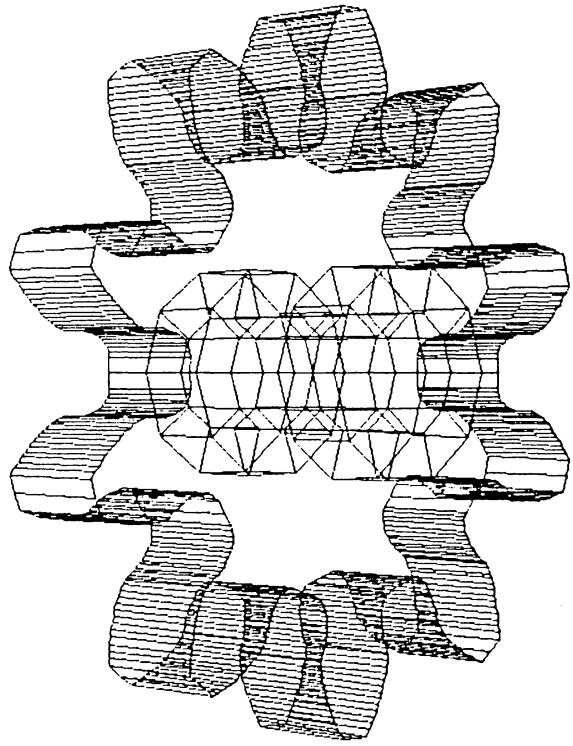
```
START
EXECUTION BEGINS...
<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
<READ:     1 PARTS; 1188 COORDINATES;  638 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
<PREVIOUS RANGE:>
<  -11.923 <X<     11.923  -12.000 <Y<     12.000    -3.000 <Z<      0.000>
<ORIGIN MOVED TO:     0.000     0.000    -1.500>
<DISTANCE TO ORIGIN:  84.00 ,ANGLE: 28.00 ,ZMIN:    0.10 ,ZMAX: 168.00>
< 1 PARTS WITH ELEMENT LIMITS:>
  45 638
>>
.rota
<AXIS, ANGLE>
.y 45
>>
.draw
>>
.exit
```

Figure 35. A 22 Tooth Gear with No Hub - MOVIE.BYU



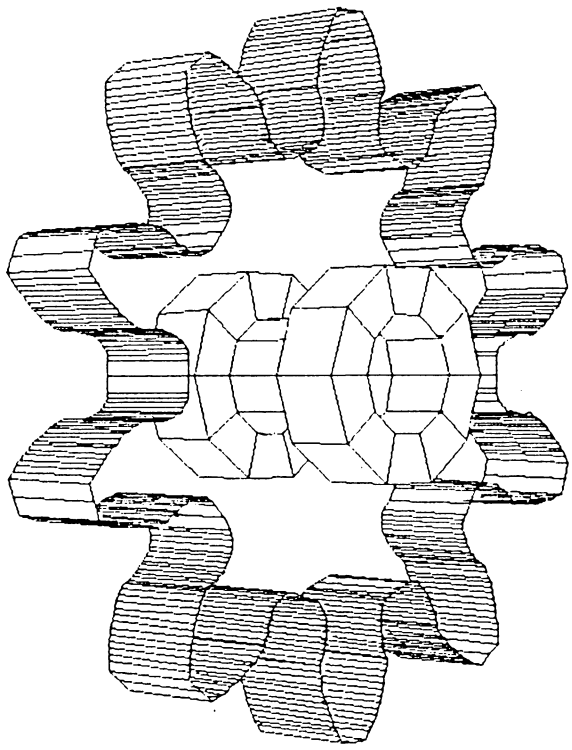
```
START
EXECUTION BEGINS...
<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
@<READ:    2 PARTS; 2275 COORDINATES; 1295 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
@<PREVIOUS RANGE:>
<  -18.492 <X<    18.492  -18.449 <Y<    18.500   -4.000 <Z<     3.000>
<ORIGIN MOVED TO:    0.000    0.025   -0.500>
<DISTANCE TO ORIGIN: 129.44 ,ANGLE: 28.00 ,ZMIN:   0.10 ,ZMAX: 258.89>
< 2 PARTS WITH ELEMENT LIMITS:>
  71115511561295
>>
.rota
<AXIS, ANGLE>
.y 45
>>
.draw
@>>
.exit
```

Figure 36. A 35 Tooth Gear with a Hub on One Side - MOVIE.BYU



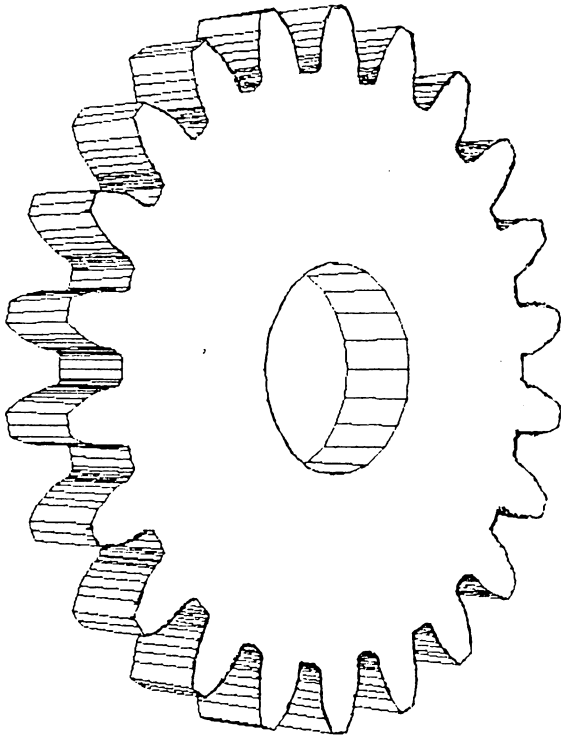
```
START
EXECUTION BEGINS...
@<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
@<READ:    3 PARTS; 1240 COORDINATES;  690 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
<PREVIOUS RANGE:>
<   -5.790 <X<     5.790   -6.000 <Y<     6.000   -3.300 <Z<     1.300>
<ORIGIN MOVED TO:    0.000    0.000   -1.000>
<DISTANCE TO ORIGIN:  42.00 ,ANGLE: 28.00 ,ZMIN:   0.10 ,ZMAX:  84.00>
< 3 PARTS WITH ELEMENT LIMITS:>
  21 610 611 650 651 690
>>
.rota
<AXIS, ANGLE>
.y 45
>>
.draw
```

Figure 37. A 10 Tooth Gear with a Hub on Both Sides - MOVIE.BYU



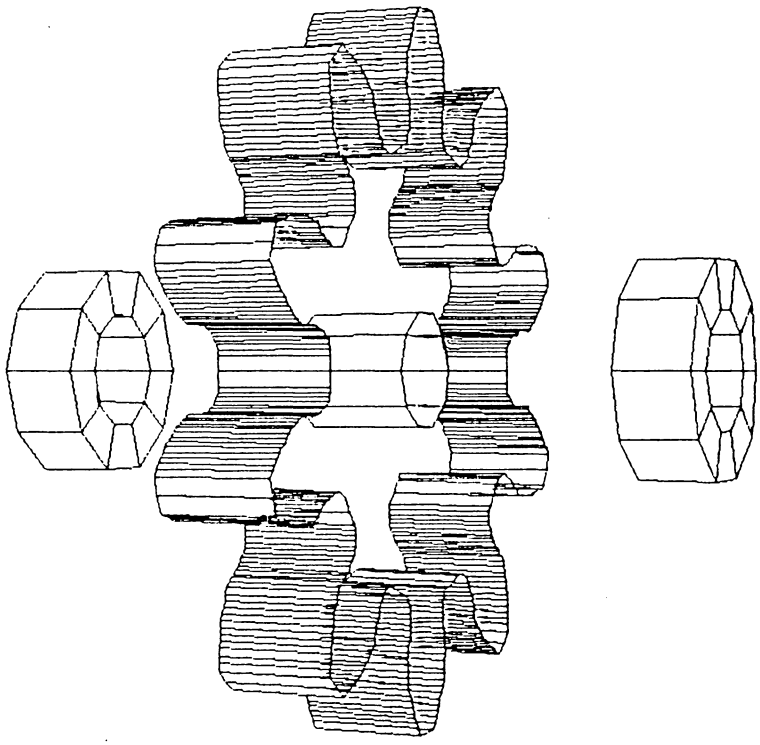
```
START
EXECUTION BEGINS...
@<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
@<READ:    3 PARTS; 1240 COORDINATES;  690 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
<PREVIOUS RANGE:>
<   -5.790 <X<     5.790   -6.000 <Y<     6.000   -3.300 <Z<     1.300>
<ORIGIN MOVED TO:    0.000    0.000   -1.000>
<DISTANCE TO ORIGIN:  42.00 ,ANGLE: 28.00 ,ZMIN:   0.10 ,ZMAX:  84.00>
< 3 PARTS WITH ELEMENT LIMITS:>
  21 610 611 650 651 690
>>
.rota
<AXIS, ANGLE>
.y 45
>>
.view
```

Figure 38. Example of Hidden Line Removal Using VIEW - MOVIE.BYU



```
START
EXECUTION BEGINS...
@<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
@<READ:     1 PARTS; 1188 COORDINATES;  638 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
@<PREVIOUS RANGE:>
<  -11.923 <X<     11.923  -12.000 <Y<     12.000    -3.000 <Z<      0.000>
<ORIGIN MOVED TO:     0.000     0.000    -1.500>
<DISTANCE TO ORIGIN:  84.00 ,ANGLE: 28.00 ,ZMIN:    0.10 ,ZMAX: 168.00>
< 1 PARTS WITH ELEMENT LIMITS:>
  45 638
>>
.view
```

Figure 39. Hidden Line Removal Using VIEW - Modified MOVIE.BYU



```
START
EXECUTION BEGINS...
<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>
.gear
<READ:    3 PARTS; 1240 COORDINATES;  690 ELEMENTS.>
<READ DISP FILE>
.
<READ FUNC FILE>
.
@<PREVIOUS RANGE:>
<   -5.790 <X<     5.790   -6.000 <Y<     6.000   -3.300 <Z<     1.300>
<ORIGIN MOVED TO:    0.000    0.000   -1.000>
<DISTANCE TO ORIGIN:  42.00 ,ANGLE: 28.00 ,ZMIN:   0.10 ,ZMAX:  84.00>
< 3 PARTS WITH ELEMENT LIMITS:>
  21 610 611 650 651 690
>>
.rota
<AXIS, ANGLE>
.y 65
>>
.explode
<PARTS I1/I2, LOCAL MOTION (X,Y,Z)>
>>>
.2 2 0 0 4
>>>
.3 3 0 0 -4
>>>
.
<LOCAL MOTION SCALE FACTOR (   0.000E+00)>
.1
>>
.view
```

Figure 40. Example of an Exploded View - MOVIE.BYU

5.5 DATA SUBROUTINE

The final choice is an option which will calculate some data about the gear being designed and write it to a file. The subroutine which performs this operation is shown in Appendix F. The data is filed under GEAR.OUT. This information may be needed or the designer may want to check a value before one of the software packages is used to represent and draw the gear. The input for the two gear examples were as follows:

	CASE 1	CASE 2
Pitch	= 1	= 1
Addendum fraction	= 1.00	= 1.00
Dedendum fraction	= 1.25	= 1.38
Hob tip radius	= 0.00	= 0.30
Number of teeth	= 22	= 27
Standard or Nonstandard	= Standard	= Standard
Pressure Angle (degrees)	= 20	= 20
What is the face width	= 3.00	= 3.25
Does the gear have a hub	= No	= Yes
On how many sides	= ---	= 2
What is the bore diameter	= 3.50	= 3.25
What is the hub extension	= ---	= 3.00

The output from this subroutine is shown in Figure 41.

```
     THIS IS THE DATA FOR THE GEAR YOU HAVE CHOSEN:


PITCH              =     1.0000000   CLEARANCE       =     0.2500000
ACTUAL ADDENDUM    =     1.0000000   WORKING DEPTH   =     2.0000000
ACTUAL DEDENDUM    =     1.2500000   WHOLE DEPTH     =     2.2500000
HOB TIP RADIUS     =     0.0000000   OUTSIDE RADIUS  =    12.0000000
NUMBER OF TEETH    =    22.0000000   BASE RADIUS     =    10.3366194
CUT. PITCH RADIUS  =    11.0000000   ROOT RADIUS     =     9.7500000
CUT. PRESS ANGLE   =    20.0000000   CIRCULAR PITCH  =     0.1784995
TOOTH THICKNESS    =     1.5707960   FACE WIDTH      =     3.0000000

  THE GEAR DOES NOT HAVE A HUB
 AND THE BORE RADIUS =     3.5000000

                         CASE 1


     THIS IS THE DATA FOR THE GEAR YOU HAVE CHOSEN:


PITCH              =     1.0000000   CLEARANCE       =     0.3800001
ACTUAL ADDENDUM    =     1.0000000   WORKING DEPTH   =     2.0000000
ACTUAL DEDENDUM    =     1.3800001   WHOLE DEPTH     =     2.3800001
HOB TIP RADIUS     =     0.3000000   OUTSIDE RADIUS  =    14.5000000
NUMBER OF TEETH    =    27.0000000   BASE RADIUS     =    12.6858511
CUT. PITCH RADIUS  =    13.5000000   ROOT RADIUS     =    12.1199999
CUT. PRESS ANGLE   =    20.0000000   CIRCULAR PITCH  =     0.1605703
TOOTH THICKNESS    =     1.5707960   FACE WIDTH      =     3.2500000

 THE GEAR HAS A HUB ON  2 SIDE(S)
 WITH A BORE RADIUS     =     3.2500000
      A HUB RADIUS      =     5.8750000
  AND A HUB PROTRUSION  =     3.0000000

                         CASE 2
```

Figure 41. Example of the Output Created from the DATA Subroutine

CHAPTER 6 CONCLUDING REMARKS AND PROGRAM EXPANDIBILITY

Realism has always been a primary objective of computer graphics. The major 3-D software packages continue to provide new facilities to help produce more lifelike pictures. Color, solid objects, shading, reflections, multiple light sources, and animation are all becoming standard in the graphics software field. But, the initial step of any drawing and the solving of many problems deal with being able to realisticly and accurately create the model as a wireframe (geometrical design model). This interactive FORTRAN program has done just that by using GKS, CADAM-CADCD, and MOVIE.BYU to represent and draw 2-D and 3-D spur gears cut by a hob. With the correct geometry written to the data base, a convincing model can be built and displayed on which to perform analysis, design review and evaluation, drafting, and NC programming.

A very realistic problem, besides those that are solved directly by checking the drawing itself, comes when the designer wants to envision how the gear will fit into a machine. He may also want to visualize how the gear will move to make sure that there are no interference problems. The computer is a much better tool to use in this case as compared to having to build an actual model or prototype. Advantages include speed, accuracy of drawings, models which are flexible, overall coordination from one central data base, and extensive analysis and simulation capabilities.

As an example, consider the CADAM sequence shown in Figures 42, 43, 44, and 45. Figure 42 shows a picture of a gear generated by CADAM for the patented Blakemore Drive mechanism [44]. The Blakemore Drive is a drive system that has an output shaft that is always connected to the prime mover and its output speed can be continuously varied from zero to double the input speed. The designer calculated that a standard spur gear with 30 teeth, and no hub would be the correct size and strength and would transfer the required power. The designer next realized that the maximum travel of the gear only accounted for eight teeth and that the rest of the teeth were not necessary, so a manual modification was made to the gear and results in Figure 43. Now with this complete, the gear can be transferred into the correct position in the mechanism to mesh with the rack. Figure 44 shows the completed mechanism after the modified 3-D spur gear has been placed into the correct position. If 2-D is desired the design can be rotated and 2-D drop-offs produced to create a 2-D isometric shown in Figure 45.

Other problems involving the gear can be solved as well if other subroutines for analysis, drawing, or modeling are written and applied.

The interactive FORTRAN program using the CAD interfaces creates a better and more accurate model than previously programmed since all three dimensions are now within the data base. Finite element programs for stress calculations are the best examples of this analysis improvement. A more precise calculation can be found since the mesh is produced across all three dimensions and not just a profile view of the gear tooth. It should

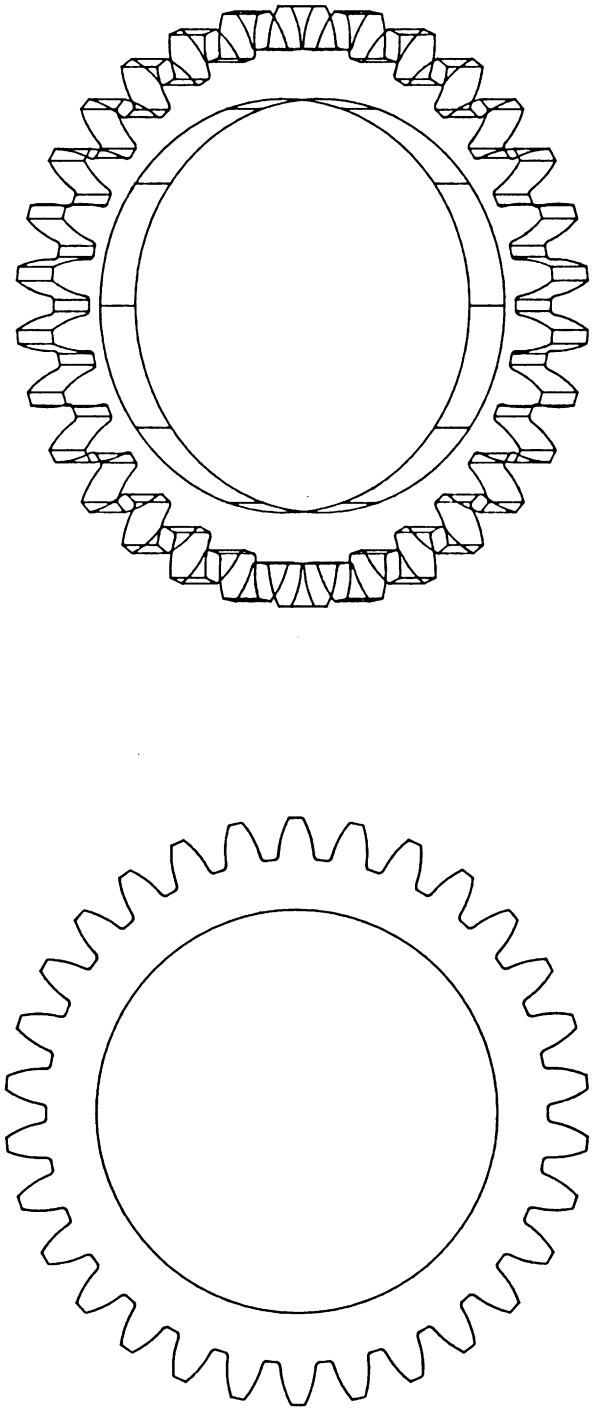


Figure 42. Gear Created by CADAM-CADCD

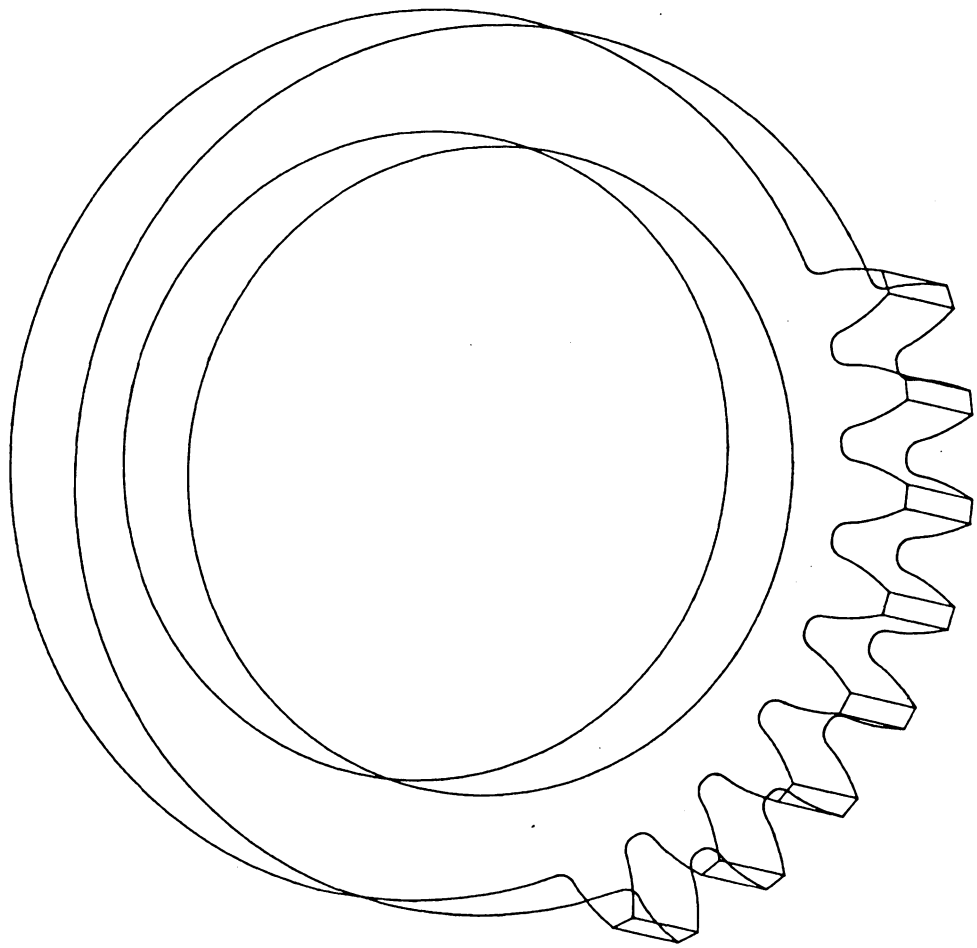


Figure 43. A Modified CADAM Gear

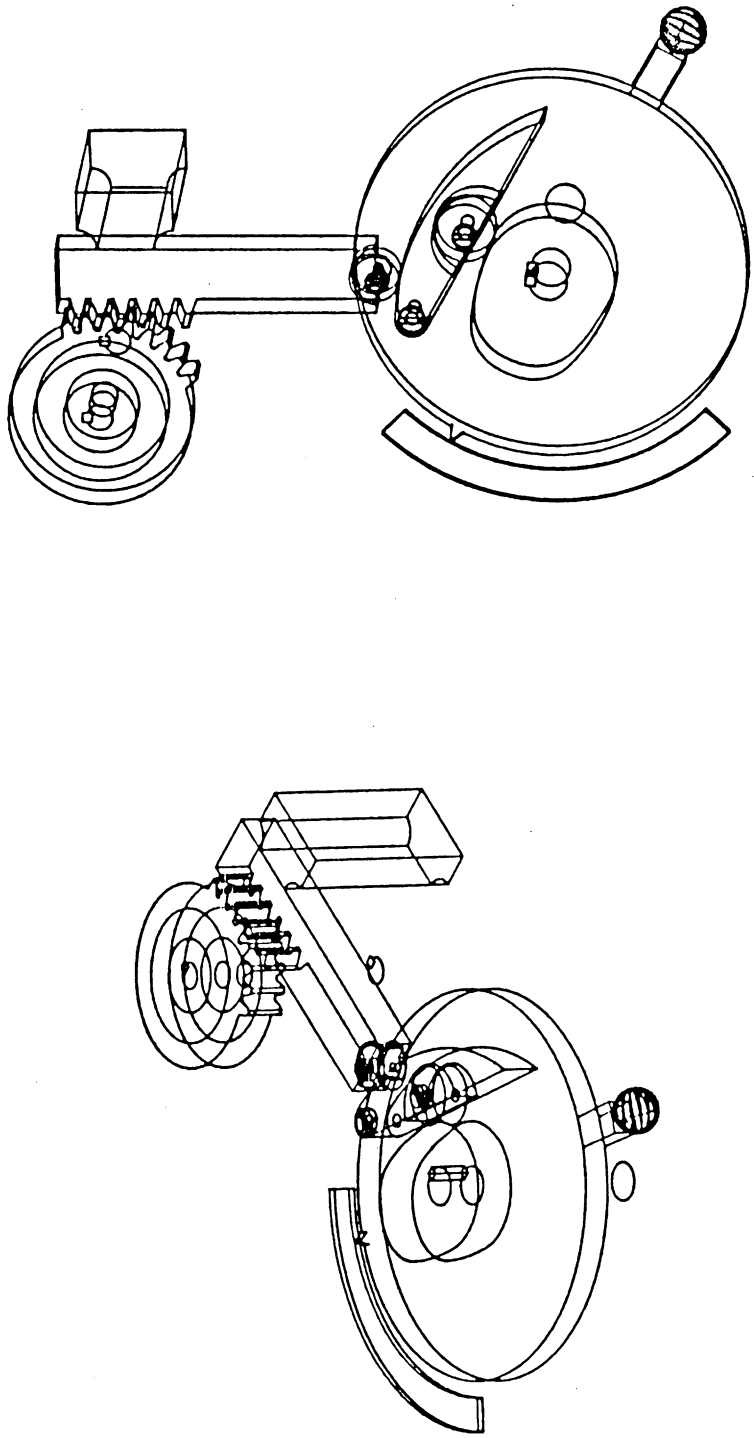


Figure 44. A Completed Blakemore Drive Mechanism with a Gear

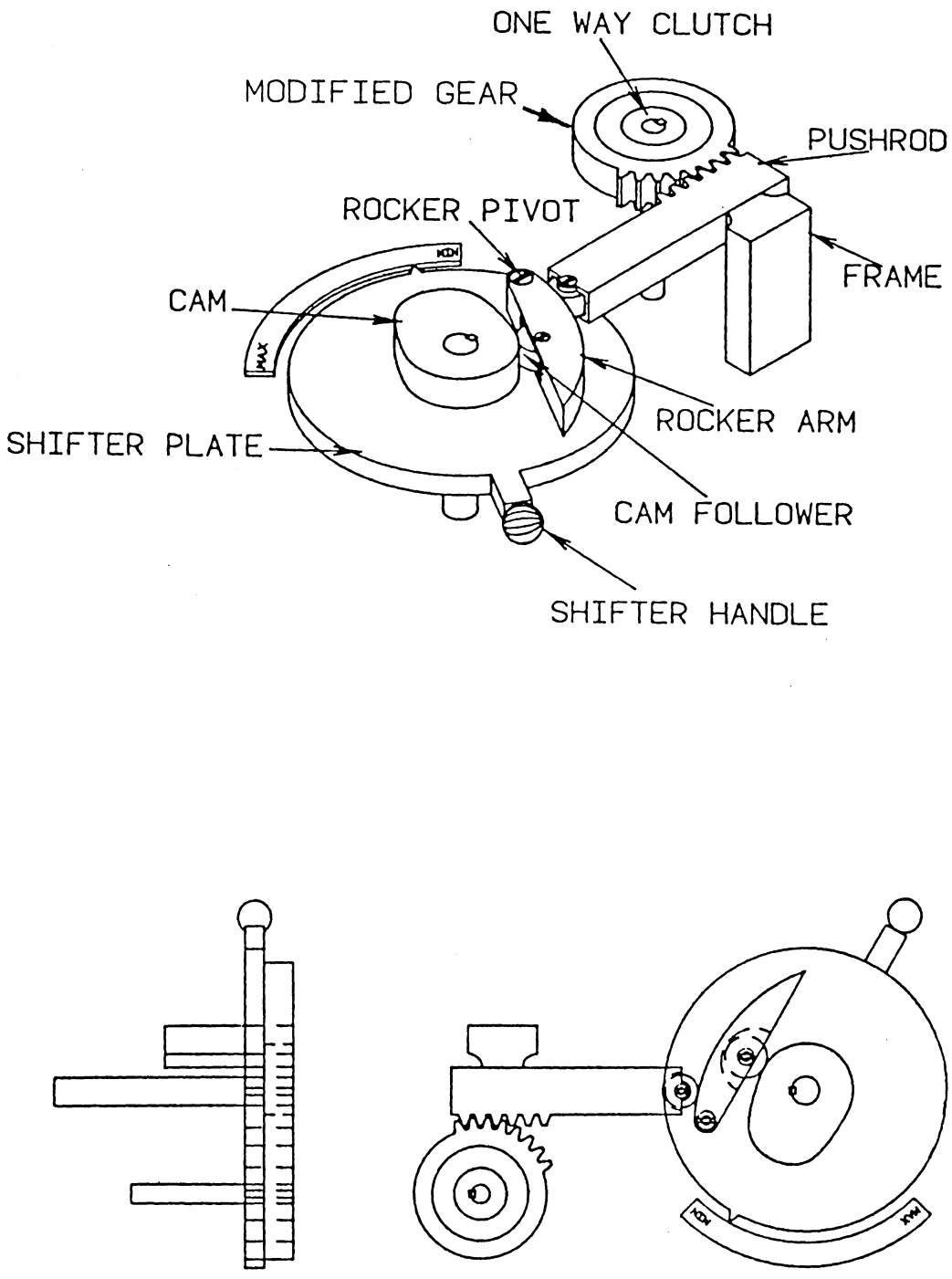


Figure 45. Front, Top, and Isometric Views of Blakemore Drive

be mentioned that CADAM has its own MESH function key and both ANSYS and NASTRAN can be used with CADAM to calculate stresses.

The J factor, K factor, and amount of undercutting produced by the hob cutter are also subroutines that have been developed in the past and could be programmed with 3-D to help in the analysis of the gear.

If Kinematic qualities were important in the design, a subroutine can be written to cut the gears by a pinion cutter instead of a hob [21]. A pinion cutter can produce a nonundercut pinion with fewer teeth than a hob, with a longer line of action, and a greater contact ratio.

Other extensions of the program could include the creation of the other gear blanks, such as the webbed and spoked types, or a material selection choice could be added, since this helps determine the strength of the gear as well as some of the geometry. Even other gear types can be programmed. Helical, bevel, and worm gears each have their own special functions and are used for certain jobs where a spur gear cannot be used or would not be a wise choice.

The CAD system, its interface, and the interactive program allow the gears to be produced quickly as a geometrical design model wherever they are needed on the screen. The part can be created and analyzed, and subtle or major changes can be made if the design is inadequate. Also, there are times when the design stage can be incorporated into a complete

process so that the data generated by the design program can be fed by numerical control to operate the cutting machine.

The possibilities and benefits of the CAD system in conjunction with interactive, user written programs, such as the design and analysis of 2-D and 3-D spur gears cut by a hob, are many. They continue to produce faster, more economical ways for engineers to perform their job.

REFERENCES

1. Baumeister, Theodore, Editor, Marks' Standard Handbook for Mechanical Engineers, Eigth Edition, McGraw-Hill Book Co., New York, NY, 1978.

2. Buckingham, Earle, Spur Gears, McGraw-Hill Book Co., New York, NY, 1928.

3. Cornell, R. W., "Compliance and Stress Sensitivity of Spur Gear Teeth," ASME Journal of Mechanical Design, Vol. 103, April 1981, pp. 447-459.

4. D. O. James: Gear Speed Reducers and Cut Gears - Catalog No. 1000, D. O. James Maufacturing Co., Chicago, IL, 1944.

5. Dudley, Darle W., Editor, Gear Handbook, McGraw-Hill Book Co., New York, NY, 1962.

6. Dudley, Darle W., Practical Gear Design, McGraw-Hill Book Co., New York, NY, 1954.

7. Khiralla, T. W., On the Geometry of External Involute Spur Gears, C/I Leaming Co., North Hollywood, CA, 1976.

8. Mabie, H. H., and Ocvirk, F. W., Mechanisms and Dynamics of Machinery, Third Edition - SI Version, John Wiley and Sons, New York, NY, 1978.

9. Martin, G. H., "Undercutting of Spur Gear Teeth," ASME Paper No. 57-S-3, 1957.

10. Michalec, George W., Precision Gearing, John Wiley and Sons, New York, NY, 1966.

11. Philadelphia Application Engineered Gearing - Catalog G-76, Philadelphia Gear, King of Prussia, PA, 1977.

12. Spotts, M. E., "How to Predict Effect of Undercutting Hobbed Spur Gear Teeth," Machine Design, Vol. 28, No. 8, 1956.

13. Trautschold, Reginald, Standard Gear Book, McGraw-Hill Book Co., New York, NY, 1935.

14. Wilcox, Lowell. E., "Gear-Tooth Stresses," Machine Design, September 1981, pp. 88-92.

15. Kochanek, G. E., "Plotting Tooth Profiles for Gears with the Computer," ASME Paper No. 68-DE-7, 1968.

16. Cooley, P., "Gear-Tooth Generation with Interactive Graphics," Computer Aided Design, Vol. 11, No. 6, Nov. 1979, pp. 353-357.

17. Mitchiner, R. G., and Mabie, H. H., "The Determination of the Lewis Form Factor and the AGMA Geometry Factor J for External Spur Gear Teeth," ASME Journal of Mechanical Design, Vol. 104, Jan. 1982, pp. 148-158.

18. Mitchiner, R. G., Mabie, H. H., and Moosavi-Rad, H., "The Undercutting of Hobbed Spur Gear Teeth," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, March 1983, pp. 122-128.

19. Moosavi-Rad, H., "The Geometry of External Spur Gear Teeth Cut by a Rounded-Tip Hob Tooth," Masters Thesis, VPI and SU, 1981.

20. Jalilvand, J., "Stress Concentrations in Undercut Spur Gear Teeth Via the Finite Element Method," Masters Thesis, VPI and SU, 1982.

21. Rhomberg, E., "A Finite Element Analysis of Pinion Shaped Spur Gear Teeth," Masters Thesis, VPI and SU, 1984.

22. Wadlington, R. P., and Hirschfeld, Fritz, "Computer-Designed Gearing," Mechanical Engineering, June 1979, pp. 32-33.

23. Cockerham G., and Waite, D., "Computer-Aided Design of Spur or Helical Gear Trains," Computer Aided Design, Vol. 8, No. 2, April 1976, pp. 84-88.

24. Ivanov, Yu I., and Matyukhina, S. I., "Computer-Aided Design of Hobs for Cylindrical Gears," Machines and Tools, Vol. XLIV, No. 12, 1972, pp. 29-30.

25. "Old Ties are Strengthening in Drive for Tech Edge," ASME NEWS, ASME, Vol. 4, No. 4, October 1984.

26. Baer, Tony, "Solid Modeling on a PC," Mechanical Engineering, Vol. 107, No. 4, April 1985, pp. 26-31.

27. Watson, H. J., Modern Gear Production, Pergamon Press Ltd., Oxford, England, 1970.

28. Crockett, J. C., Gear Cutting Practice, Machines and Tools, The Machinery Publishing Co. Ltd., Sussex, England, 1971.

29. Tucker, A. I., "The Gear Design Process," ASME Paper No. 80-C2/DET-13, 1980.

30. Foley, J. D. and VanDam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Co., Reading, MA, 1982.

31. Groover, Mikell P., and Zimmers, Emory W., CAD/CAM: Computer - Aided Design and Manufacturing, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

32. Parslow, R. D., Prowse, R. W., and Green R. E., Computer Graphics, Plenum Press, London, England, 1969.

33. Hopgood, F. R. A., Duce, D. A., Gallop, J. R., and Sutcliffe, D. C., Introduction to the Graphical Kernel System - GKS, Academic Press, London, England, 1983.

34. Harris, Dennis, Computer Graphics and Applications, Chapman and Hall, London, England, 1984.

35. Tektronix, PLOT10 Graphical Kernel System (GKS), Tektronix, Beaverton, OR, 1984.

36. CADAM, Basic Operator Course - Student Text, First Edition, CADAM/IBM, February 1983.

37. CADAM, CADAM - Geometry Interface Module and Istallation and Programmers Guide, Program No.: 5796-ATJ, Release 19.1, Document No. SH20-2099-6, Eigth Edition, CADAM/IBM, August 1983.

38. CADAM, CADAM - Geometry Interface Installation Guide, Program No.: 5668-842, Document No. SH20-6227-0, First Edition, CADAM/IBM, January 1985.

39. CADAM, CADAM - General Information Manual, Program Nos.: 5796-ATA to ATJ, AWR, AWT, Document No. G320-6667-0, First Edition, CADAM/IBM, January 1982.

40. Christiansen, H., and Stephenson, S., MOVIE.BYU: A General Purpose Computer Graphics System, MOVIE Document, Version 5.1, April 1983 Edition, January 1984 Revision.

41. Christiansen, H., and Stephenson, S., MOVIE.BYU Training Manuel, 1986.

42. Carnahan, B., Luther, H. A., and Wilkes, G. O., Applied Numerical Methods, John Wiley and Sons, 1969.

43. Suh, C. H., and Radcliffe, C. W., Kinematics and Mechanisms Design, Robert E. Krieger Publishing Co., Malabar, FL, 1983.

44. Soderholm, L. G., "Clutch Cluster Collects Reciprocating Motions," Design News, November 6, 1972, p. 86.

APPENDIX A. MAIN PROGRAM - HOBBED SPUR GEARS

```
**********************************************************************************
**********************************************************************************
********                                                          ********
******  THIS PART OF THE PROGRAM WILL DRIVE THE GEAR SUBROUTINE,  ******
******          IT ASKS FOR THE TYPE OF GEAR TO DESIGN.           ******
********                                                          ********
**********************************************************************************
**********************************************************************************


*    THIS PART OF THE PROGRAM ALLOWS THE USER TO SELECT THE TYPE OF
*    GEAR TO BE DESIGNED. (ONLY THE SPUR TYPE OF GEAR IS FUNCTIONAL.)


*       ARGUMENTS:
*   IGTYPE =   INTEGER NUMBER REPRESENTING THE TYPE OF GEAR TO DESIGN AND
*              DIRECTING THE PROGRAM TO THE CORRECT SUBROUTINE.
*   SYSCAL =   SUBROUTINE AT VIRGINIA TECH CALLED TO CLEAR THE SCREEN


*   VARIABLE DECLARATIONS
        INTEGER IGTYPE, NO, NI, IERR

        CHARACTER*5 CLE

        DATA NO/6/, NI/5/
        DATA CLE /'CLEAR'/

        CALL SYSCAL (CLE, 5, IERR)
        WRITE(NO,*)
        WRITE(NO,*)'> GEAR'
        WRITE(NO,*)
2000    WRITE(NO,*)
        WRITE(NO,*)
        WRITE(NO,*)' WHAT TYPE OF GEAR WOULD YOU LIKE TO DESIGN?'
        WRITE(NO,*)
        WRITE(NO,*)
        WRITE(NO,*)'              1)  SPUR      '
        WRITE(NO,*)'              2)  HELICAL   '
        WRITE(NO,*)'              3)  BEVEL     '
        WRITE(NO,*)'              4)  WORM      '
        WRITE(NO,*)
        WRITE(NO,*)
        WRITE(NO,*)'                CHOOSE 1,2,3 OR 4  (5 TO QUIT)'
        WRITE(NO,*)
```

```
      READ(NI,*) IGTYPE

      IF (IGTYPE .EQ. 1) THEN
         GOTO 2001
      ELSEIF (IGTYPE .EQ. 2) THEN
         GOTO 2002
      ELSEIF (IGTYPE .EQ. 3) THEN
         GOTO 2003
      ELSEIF (IGTYPE .EQ. 4) THEN
         GOTO 2004
      ELSEIF (IGTYPE .EQ. 5) THEN
         GOTO 2999
      ELSE
         GOTO 2000
      ENDIF

2001  CALL SPUR
      GOTO 2999

2002  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 2000

2003  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 2000

2004  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 2000

2999  END
```

```
**********************************************************************************
**********************************************************************************
********                                                                 ********
*****     THIS SELECTS THE TYPE OF CUTTER USED TO CUT THE GEAR          *****
********                                                                 ********
**********************************************************************************
**********************************************************************************

      SUBROUTINE SPUR

*    THIS SUBROUTINE ASKS THE USER TO CHOOSE THE TYPE OF CUTTER TO
*    FORM THE GEAR. (ONLY THE HOB IS FUNCTIONAL.)


*       ARGUMENTS:
*   ICUTTR =   INTEGER NUMBER REPRESENTING THE TYPE OF CUTTER USED TO
*              DESIGN THE GEAR AND DIRECTING THE PROGRAM TO THE CORRECT
*              SUBROUTINE.
*   SYSCAL =   SUBROUTINE AT VIRGINIA TECH CALLED TO CLEAR THE SCREEN


*   VARIABLE DECLARATIONS
      INTEGER ICUTTR, NO, NI,IERR

      CHARACTER*5 CLE

      DATA NO/6/, NI/5/
      DATA CLE /'CLEAR'/


      CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)'>> SPUR'
      WRITE(NO,*)
3000  WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)' WHAT TYPE OF CUTTER WOULD YOU LIKE TO USE TO CUT'
      WRITE(NO,*)'               THE GEAR TEETH?'
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)'              1)    HOB'
      WRITE(NO,*)'              2)    PINION CUTTER'
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)'                 CHOOSE 1 OR 2  (3 TO QUIT)'

      READ(NI,*) ICUTTR

      IF (ICUTTR .EQ. 1) THEN
         GOTO 3001
      ELSEIF (ICUTTR .EQ. 2) THEN
```

```
         GOTO 3002
      ELSEIF (ICUTTR .EQ. 3) THEN
         GOTO 3999
      ELSE
         GOTO 3000
      ENDIF


3001  CALL HOB
      GOTO 3999

3002  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 3000

3999  END
```

```
************************************************************************************************
************************************************************************************************
**********                                                                          **********
********     THIS SUBROUTINE WILL SET UP THE GEAR TOOTH PROFILE                   ********
******     GEOMETRY WHEN THE GEAR IS CUT BY A HOB.  IT PRODUCES                     ******
********       AN INVOLUTE TOOTH AND ITS TROCHOID FILLET                          ********
**********                                                                          **********
************************************************************************************************
************************************************************************************************

      SUBROUTINE HOB


*     ARGUMENTS:
*  A       =  ADDENDUM FRACTION
*  ADDEND  =  ACTUAL ADDENDUM (FUNCTION OF P)
*  ALPHA   =  ANGLE USED IN CONSTRUCTION OF THE TROCHOID AND INVOLUTE
*             CURVES
*  ALPHAI  =  ALPHA OF THE INVOLUTE INTERSECTION AND ANGLE USED IN THE
*             CONSTRUCTION OF THE INVOLUTE CURVE
*  AMOUNT  =  DISTANCE THE HUB PROTRUDES FROM THE GEAR BLANK FACE
*  ANSWER  =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ANS2    =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ANS4    =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  B       =  DEDENDUM FRACTION
*  BASERD  =  BASE RADIUS OF GEAR
*  BETA    =  ANGLE MEASURED FROM THE CENTERLINE OF THE TOOTH TO THE
*             CENTERLINE OF THE TOOTH SPACE MINUS 'ETA'
*  CHKN1   =  VARIABLE TO CHECK THE MINIMUM NUMBER OF TEETH WITHOUT
*             UNDERCUTTING
*  CHKRHF  =  VARIABLE TO CHECK THE HOB TIP RADIUS INPUT VS 'RF' - THE
*             MAXIMUM ALLOWABLE HOB TIP RADIUS
*  CLEAR   =  CLEARANCE
*  DEDEND  =  ACTUAL DEDENDUM (FUNCTION OF P)
*  DEL     =  INCREMENT REPRESENTING THE CHANGE IN ANGLES
*  DELTA   =  HALF-WIDTH OF THE HOB TOOTH LAND
*  ETA     =  ANGLE USED IN CONSTRUCTION OF TROCHOID
*  FACE    =  FACE WIDTH OF THE GEAR
*  FINAM   =  10-CHARACTER STRING FOR THE NAME OF A FILE
*  HBTPRD  =  ACTUAL HOB TIP RADIUS (FUNCTION OF P)
*  HK      =  WORKING DEPTH OF GEAR TOOTH
*  HOLDIA  =  DIAMETER OF THE BORE HOLE
*  HOLRAD  =  RADIUS OF THE BORE HOLE
*  HT      =  WHOLE DEPTH OF GEAR TOOTH
*  HUBRAD  =  RADIUS OF THE HUB
*  I       =  INTEGER COUNTER
*  IBLANK  =  INTEGER REPRESENTING THE TYPE OF GEAR BLANK TO BE DESIGNED
*  ICHKN2  =  INTEGER VALUE OF 'CHKN1' RAISED TO THE NEXT WHOLE NUMBER
*  IDTRCH  =  INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*             BETWEEN THE ROOT CIRCLE (DEDENDUM CIRCLE) AND TROCHOID
*  IEND    =  INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
```

```
*              BETWEEN CENTERLINE OF THE TOOTH AND ADDENDUM CIRCLE
*  IERR    =   INTEGER USED AS AN ERROR CONTROL VARIABLE
*  IGRAPH  =   INTEGER REPRESENTING THE TYPE OF GRAPHICS DISPLAY CHOSEN
*  IINVAD  =   INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*              BETWEEN THE INVOLUTE CURVE AND THE ADDENDUM CIRCLE
*  INTRVL  =   INTEGER VALUE OF 'XINT' TO THE NEAREST WHOLE NUMBER
*  IRPLY   =   1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ITER    =   INTEGER COUNTER USED TO COUNT THE NUMBER OF ITERATIONS
*              USED TO FIND THE INVOLUTE/TROCHOID INTERSECTION
*  ITRINV  =   INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*              BETWEEN THE TROCHOID CURVE AND THE INVOLUTE CURVE
*  IYES    =   1-CHARACTER WORD 'Y' (MEANING YES) USED TO CHECK AN ANSWER
*              TO A QUESTION
*  LEND    =   INTEGER USED TO SAVE THE COORDINATES OF THE LAST POINT
*              ON THE FIRST TOOTH GEOMETRY GENERATED
*  LL      =   INTEGER USED TO CHECK THE MAXIMUM STEP BETWEEN POINTS IN
*              THE GEAR GEOMETRY
*  N       =   NUMBER OF TEETH IN THE GEAR
*  NI      =   INTEGER USED TO READ DATA FROM THE SCREEN
*  NINVPT  =   INTEGER COUNTER FOR INVOLUTE POINTS
*  NN      =   INTEGER COUNTER
*  NNN     =   INTEGER COUNTER
*  NNX     =   INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*              POINT BETWEEN THE INVOLUTE CURVE AND THE ADDENDUM CIRCLE
*  NO      =   INTEGER USED TO WRITE DATA OR STATEMENTS TO THE SCREEN
*  NTROPT  =   INTEGER COUNTER FOR TROCHOID POINTS
*  OUTDIA  =   OUTSIDE DIAMETER OF GEAR
*  OUTRAD  =   OUTSIDE RADIUS OF GEAR
*  P       =   DIAMETRAL PITCH
*  PCIRC   =   CIRCULAR PITCH
*  PHI     =   THE CUTTING PRESSURE ANGLE IN RADIANS
*  PHIA    =   ANGLE USED IN THE DEVELOPMENT OF ADDENDUM CIRCLE
*  PHIDEG  =   THE CUTTING PRESSURE ANGLE IN DEGREES
*  PI      =   3.1415927
*  PITCHD  =   CUTTING PITCH DIAMETER OF GEAR
*  PITCHR  =   CUTTING PITCH RADIUS OF GEAR
*  PROTRU  =   INTEGER NUMBER   0 - NO HUB ON GEAR
*                               1 - HUB ON ONE SIDE OF GEAR
*                               2 - HUB ON BOTH SIDES OF GEAR
*  R       =   THE CUTTING PITCH RADIUS OF THE GEAR BEING DESIGNED
*              (STANDARD OR NON-STANDARD)
*  RADIUS  =   RADIUS USED IN CONSTRUCTION OF ENTIRE GEAR
*  RF      =   TIP RADIUS OF A FULLY ROUNDED HOB
*  RHF     =   HOB-TOOTH TIP RADIUS (0.0 REPRESENTS A SHARP-TIPPED HOB)
*  RI      =   RADIUS OF INVOLUTE/TROCHOID INTERSECTION/TANGENCY POINT
*  RM      =   RADIUS CHECK TO MAKE SURE INVOLUTE CURVE DOES NOT EXCEED
*              OUTSIDE RADIUS OF GEAR
*  RO      =   OUTSIDE RADIUS OF GEAR
*  ROOTD   =   ROOT DIAMETER OF GEAR
*  ROOTRD  =   ROOT RADIUS OF GEAR
*  SP      =   LENGTH BETWEEN TWO SUCCESSIVE POINTS OF GEAR GEOMETRY
```

```
*            (A CHECK AGAINST 'SPMINN'/'SPMAXX')
*  SPACNG =  VARIABLE USED IN A SEQUENCE TO DECIDE THE SPACING BETWEEN
*            POINTS
*  SPMAX  =  MAXIMUM ALLOWABLE SPACING BETWEEN POINTS
*            (BASED ON 'SPACNG')
*  SPMAXX =  MODIFIED MAXIMUM ALLOWABLE SPACING BETWEEN POINTS
*  SPMIN  =  MINIMUM ALLOWABLE SPACING BETWEEN POINTS
*            (BASED ON 'SPACNG')
*  SPMINN =  MODIFIED MINIMUM ALLOWABLE SPACING BETWEEN POINTS
*  SYSCAL =  SUBROUTINE AT VIRGINIA TECH CALLED TO CLEAR THE SCREEN
*  TH     =  ANGLE USED IN CONSTRUCTION OF ROOT AND ADDENDUM CIRCLES AND
*            TROCHOID CURVE
*  THETAI =  THETA OF THE TROCHOID INTERSECTION
*  THKNSS =  THE TOOTH THICKNESS OF THE GEAR BEING DESIGNED
*  TO     =  TOOTH THICKNESS AT ADDENDUM CIRCLE
*  TT     =  THE TOOTH THICKNESS OF A STANDARD GEAR
*  TYPE   =  7-CHARACTER STRING FOR THE FILE TYPE
*  WW     =  THE CUTTING PITCH RADIUS OF A STANDARD GEAR
*  X      =  X-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  Y      =  Y-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  XE     =  X-COORDINATE OF THE INTERSECTION POINT BETWEEN THE ROOT
*            CIRCLE AND THE TROCHOID CURVE
*  YE     =  Y-COORDINATE OF THE INTERSECTION POINT BETWEEN THE ROOT
*            CIRCLE AND THE TROCHOID CURVE
*  XI     =  X-COORDINATE OF INVOLUTE/TROCHOID INTERSECTION/TANGENCY
*            POINT
*  YI     =  Y-COORDINATE OF INVOLUTE/TROCHOID INTERSECTION/TANGENCY
*            POINT
*  XINT   =  THE NUMBER OF INTERVALS PER CURVE
*  XL     =  DISTANCE BETWEEN THE CENTERLINE OF THE TOOTH SPACE/ROOT
*             CIRCLE INTERSECTION POINT AND THE ROOT CIRCLE/TROCHOID
*             CURVE INTERSECTION POINT
*  XS     =  X-COORDINATE AT THE INTERSECTION POINT BETWEEN THE
*            CENTERLINE OF THE GEAR TOOTH SPACE AND THE ROOT CIRCLE
*  YS     =  Y-COORDINATE AT THE INTERSECTION POINT BETWEEN THE
*            CENTERLINE OF THE GEAR TOOTH SPACE AND THE ROOT CIRCLE
*  XSAVE  =  SPECIAL X-COORDINATES SAVED ON THE GEAR TOOTH
*  YSAVE  =  SPECIAL Y-COORDINATES SAVED ON THE GEAR TOOTH
*  XX     =  X-COORDINATE DIFFERENCE BETWEEN SUCCESSIVE POINTS IN
*            GEAR GEOMETRY
*  YY     =  Y-COORDINATE DIFFERENCE BETWEEN SUCCESSIVE POINTS IN
*            GEAR GEOMETRY
*  XXX    =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  YYY    =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  XX2    =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*            GEOMETRY
```

```
*  YY2     =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*             GEOMETRY


*  VARIABLE DECLARATIONS
      INTEGER INTRVL, I, IDTRCH, NN, NTROPT, ITER, IERR, NO, NI,
     C        ITRINV, NINVPT, IINVAD, IEND, LL, JJ, LEND, NNN,
     C        ICHKN2, IGRAPH, PROTRU, MIDSTP, NNX, IBLANK


      REAL X(3000), Y(3000), PI, N, P, A, B, RHF, R, PHIDEG, THKNSS,
     C     XINT, PHI, TT, WW, XI, YI, RI, ALPHAI, THETAI, DELTA,
     C     ETA, BETA, XE, YE, XS, YS, XL, DEL, TH, SPACNG,
     C     SPMIN, SPMAX, SPMINN, SPMAXX, XX, YY, RM, ALPHA, RO,
     C     SP, PHIA, TO, INCR1, INCR2, ANGLE(3000), RADIUS(3000),
     C     ANGL2(3000), X1(3000), Y1(3000), CHKRHF, RF, CHKN1, FACE,
     C     DPHI, RPHI, XXX(722), YYY(722), ADDEND, DEDEND, CLEAR,
     C     HK, HT, HBTPRD, OUTRAD, PITCHR, BASERD, ROOTRD, BORECK,
     C     KEYWAY, HUBTHK, PITCHD, ROOTD, HUBDIA, OUTDIA, HOLDIA,
     C     HUBCHK, XX2(722), YY2(722), XSAVE(8),
     C     YSAVE(8), PCIRC


      CHARACTER*10 FINAM
      CHARACTER*7 TYPE
      CHARACTER*5 CLE
      CHARACTER*1 IYES, IRPLY, ANSWER, ANS2, ANS3, ANS4, ANS5, ANS6

      COMMON /TODATA/ ADDEND, DEDEND, HBTPRD, PHIDEG, CLEAR, HK, HT,
     C                OUTRAD, BASERD, ROOTRD, PCIRC
      COMMON /TOOTH/ N, P, A, B, RHF, THKNSS, PHI, R
      COMMON /INTPRO/ PROTRU
      COMMON /RELPRO/ AMOUNT, HOLRAD, FACE, HUBRAD

      DATA PI/3.1415927/, IYES/'Y'/
      DATA NO/6/, NI/5/
      DATA CLE /'CLEAR'/


*****************************************************************************
*****************************************************************************
      CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)'>>> HOB'
      WRITE(NO,*)
*
*                        THIS IS FROM AGMA 201.02, AUGUST 1968
*                        COARSE PITCH (1 - 19.99 P)
*
      WRITE(NO,*)' FOR STANDARD GEARS, 20 AND 25 DEGREE PRESSURE'
      WRITE(NO,*)'         ANGLE, FULL-DEPTH, COARSE PITCH:'
```

```
      WRITE(NO,*)' THE ADDENDUM FRACTION = 1.000  (0.800 FOR STUB)'
      WRITE(NO,*)' THE DEDENDUM FRACTION = 1.250  (1.000 FOR STUB)'
      WRITE(NO,*)
*
*                       THE 14.5 DEGREE PRESSURE ANGLE IS
*                       OBSOLETE UNDER AGMA STANDARDS
*
      WRITE(NO,*)' FOR STANDARD GEARS, 14.5 DEGREE PRESSURE ANGLE,'
      WRITE(NO,*)'          FULL-DEPTH, COARSE PITCH:'
      WRITE(NO,*)'        THE ADDENDUM FRACTION = 1.000'
      WRITE(NO,*)'        THE DEDENDUM FRACTION = 1.157'
      WRITE(NO,*)
      WRITE(NO,*)' A HOB TIP RADIUS OF 0.0 MEANS A SHARP-TIPPED HOB'
      WRITE(NO,*)
      WRITE(NO,*)

322   WRITE(NO,*)' ENTER THE FOLLOWING DATA:'
      WRITE(NO,*)
      WRITE(NO,*)'       DIAMETRAL PITCH, ADDENDUM FRACTION,'
      WRITE(NO,*)'       DEDENDUM FRACTION, HOB TIP RADIUS,'
      WRITE(NO,*)'       AND NUMBER OF TEETH'

      READ(NI,*) P, A, B, RHF, N


*                     THIS IS FROM AGMA 207.06, NOVEMBER 1974
*                     FINE PITCH (20-200 P)
*
*                  FOR STANDARD GEARS, 20 DEGREE PRESSURE ANGLE,
*                           FULL-DEPTH, FINE PITCH
*                       THE ADDENDUM FRACTION = 1.000
*                       THE DEDENDUM FRACTION = 1.200 + 0.002*P
*

      CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)' IS THIS GOING TO BE A STANDARD GEAR (Y OR N)?'
      WRITE(NO,*)

      READ(NI,10) IRPLY
10    FORMAT (A1)

      TT = PI / (2. * P)
      WW = N / (2. * P)

      IF (IRPLY .NE. IYES) THEN
         WRITE(NO,*)
         WRITE(NO,15) WW, TT
15       FORMAT(1X,'ON THE CUTTING PITCH CIRCLE ENTER:',//,
     C    ' THE CUTTING PITCH RADIUS (',F15.9,' FOR STANDARD GEARS)',/,
```

```
C     ' THE CUTTING PRESSURE ANGLE (IN DEGREES)',/,
C     ' AND THE TOOTH THICKNESS (',F15.9,' FOR STANDARD GEARS)')
      READ(NI,*) R, PHIDEG, THKNSS

      ELSE
         R = WW
         THKNSS = TT
         WRITE(NO,*)
         WRITE(NO,*)' ENTER THE CUTTING PRESSURE ANGLE (IN DEGREES)'
         READ(NI,*)PHIDEG

      ENDIF


      PHI = PHIDEG * PI / 180.

*     THIS SECTION FUNCTIONS AS A CHECK OF THE INPUT FROM ABOVE.
*     IT CHECKS THE HOB TIP RADIUS, AND THE NUMBER OF TEETH TO SEE
*     IF UNDERCUTTING EXISTS.


*     THIS EQUATION IS FROM THE WORK OF MITCHINER, MABIE, MOOSAVI-RAD
*
*     THE VARIABLE 'RF' REPRESENTS THE VALUE OF THE TIP OF A FULLY
*     ROUNDED HOB CUTTER TOOTH PROFILE.  SO ANY HOB TIP RADIUS SHOULD
*     BE LESS THAN THIS NUMBER.  A HOB TIP RADIUS OF 0 (ZERO) REPRESENTS
*     A SHARP TIPPED HOB.

      CHKRHF = RHF / P
      RF = (1./(1.-SIN(PHI))) * ((PI/(4.*P))*COS(PHI)-B/P*SIN(PHI))

      IF (CHKRHF .GT. RF) THEN
         WRITE(NO,*)
         WRITE(NO,*)
         WRITE(NO,*)'***** THE HOB TIP RADIUS IS TOO LARGE.'
         WRITE(NO,16) RF
16       FORMAT(1X,'      CHOOSE A NUMBER THAT IS LESS THEN',F9.5)
         WRITE(NO,*)
         WRITE(NO,*)' ENTER THE NEW CHOICE OF THE HOB TIP RADIUS.'
         READ(NI,*) RHF
      ENDIF


*      THIS EQUATION IS DEVELOPED AS A CHECK FOR THE NUMBER OF
*      TEETH THAT WILL ELIMINATE UNDERCUTTING.  IF THE NUMBER OF
*      TEETH SPECIFIED IS LESS THAN THE MINIMUM TO AVOID UNDERCUTTING
*      FROM THE EQUATION, THE PROGRAM ASKS THE USER IF THE
*      UNDERCUTTING IS ACCEPTABLE.  IF IT IS NOT ACCEPTABLE, THE USER
*      CAN CHANGE THE NUMBER OF TEETH IN THE GEAR.

      CHKN1 = 2.*P*(B/P+RHF/P*(SIN(PHI)-1.)) / SIN(PHI)**2
```

```
      ICHKN2 = INT(CHKN1 + 0.999)

      IF (CHKN1 .GT. N) THEN
         WRITE(NO,*)
         WRITE(NO,*)
         WRITE(NO,*)' ***** THE GEAR IS BEING UNDERCUT!'
         WRITE(NO,*)
         WRITE(NO,17) ICHKN2
17       FORMAT ('      TO AVOID UNDERCUTTING THE GEAR TEETH',/,
     C           '      INCREASE THE NUMBER OF TEETH TO ',I4)
         WRITE(NO,*)
         WRITE(NO,*)' WOULD YOU LIKE TO CHANGE THE NUMBER OF TEETH IN TH
     CE GEAR (Y OR N)?'
         READ(NI,18) ANS2
18       FORMAT(A1)

         IF (ANS2 .EQ. 'Y') THEN
            WRITE(NO,*)'  ENTER THE NUMBER OF TEETH.'
            READ(NI,*) N
            IF (IRPLY .EQ. IYES) THEN
               WW = N / (2. * P)
               R = WW
            ENDIF
         ENDIF
      ENDIF


      CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)
19    WRITE(NO,20) P, A, B, RHF, N, R, PHIDEG, THKNSS
20    FORMAT('   THE INPUT DATA IS:',//,
     C       '    PITCH                = ',F15.7,/,
     C       '    ADDENDUM FRACTION    = ',F15.7,/,
     C       '    DEDENDUM FRACTION    = ',F15.7,/,
     C       '    HOB TIP RADIUS       = ',F15.7,/,
     C       '    NUMBER OF TEETH      = ',F15.7,/,
     C       '    CUTTING PITCH RADIUS = ',F15.7,/,
     C       '    CUTTING PRESS ANGLE  = ',F15.7,/,
     C       '    TOOTH THICKNESS      = ',F15.7,///,
     C       '    IS THE INPUT DATA CORRECT (Y OR N)?')

      READ(NI,22) ANSWER
22    FORMAT(A1)

      IF (ANSWER .EQ. 'N') THEN
         WRITE(NO,*)
         WRITE(NO,*)' REENTER DATA!'
         WRITE(NO,*)
         GOTO 322
      ELSEIF (ANSWER .EQ. 'Y') THEN
```

```
         GOTO 23
      ELSE
         GOTO 19
      ENDIF

23    ADDEND = A / P
      DEDEND = B / P
      CLEAR  = DEDEND - ADDEND
      HK     = 2. * ADDEND
      HT     = ADDEND + DEDEND
      HBTPRD = RHF / P
      OUTRAD = R + ADDEND
      OUTDIA = 2. * OUTRAD
      PITCHR = R
      PITCHD = 2. * R
      BASERD = PITCHR * COS(PHI)
      ROOTRD = R - DEDEND
      ROOTD  = 2. * ROOTRD
      PCIRC  = PI * B / N


*  THE MAXIMUM NUMBER OF INTERVALS FOR EACH CURVE IS LIMITED BY THE
*  SOFTWARE PACKAGE ITSELF OR THE HARDWARE AT VIRGINIA TECH
*  GKS CAN HANDLE A LARGE NUMBER OF INTERVALS.  80 WAS CHOSEN AS A
*     LIMIT BECAUSE OF THE SINGLE PRECISION OF THE PROGRAM.
*  CADAM IS LIMITED ONLY BY THE AMOUNT OF DISK SPACE THAT
*     VIRGINIA TECH GIVES EACH CADAM MODEL (30,000 WORDS)
*  MOVIE.BYU IS LIMITED BY THE WAY DATA IS READ IN AND OUT OF THE
*     DATA FILES

      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)'   ENTER THE NUMBER OF INTERVALS PER CURVE'
      WRITE(NO,*)'        (4 MIN - 80 MAX FOR GKS)'
      WRITE(NO,*)'        (4 MIN - 25 MAX FOR CADAM)'
      WRITE(NO,*)'        (4 MIN - 15 MAX FOR MOVIE.BYU)'

      READ(NO,*) XINT

      INTRVL = INT(XINT + 0.5)
      SPACNG = 0.01 * ((A + B) / P)
      SPMIN = 0.95 * SPACNG
      SPMAX = 1.05 * SPACNG

      X(1) = 0.0
      Y(1) = 0.0
*
*
*     LOCATE THE INTERSECTION/TANGENCY POINT
*
*
```

```
      CALL INTRSC (RI, XI, YI, ALPHAI, THETAI, ITER, 0, IERR)


      IF (IERR .NE. 0) THEN
         WRITE(NO,*)' CANNOT LOCATE TROCHOID/INVOLUTE INTERSECTION'

         IF (IERR .EQ. 99) THEN
            WRITE(NO,*)' LACK OF CONVERGENCE IN 2000 ITERATIONS'
         ELSEIF (IERR .EQ. 1) THEN
            WRITE(NO,*)' POINT OF INTERSECTION IS NOT WITHIN REASON'
*                                                NOT LOCATED PLAUSIBLY
            WRITE(NO,*)
            WRITE(NO,*)'   REENTER INPUT!'
            GOTO 322
         ENDIF

      ENDIF

      IF (IERR .EQ. 0) THEN
         CALL SYSCAL (CLE, 5, IERR)
         WRITE(NO,*)
         WRITE(NO,25) XI, YI, RI, ITER
25       FORMAT(1X, 'INTERSECTION POINT LOCATED AT:',/,
     C          1X,' X = ',F12.8, 3X,'  Y= ',F12.8,//,
     C          '  AT A RADIUS OF ',F12.8,//,
     C          '  AND LOCATED IN',I5,' ITERATIONS.')
      ENDIF
*
*
*     DEVELOP ROOT CIRCLE
*
*
      DELTA = ((PI/P - THKNSS)/2.0) - (B/P - RHF/P) * TAN(PHI) -
     C        RHF/(P * COS(PHI))
      ETA = DELTA / R
      BETA = PI / N - ETA
      XE = (R - B / P) * SIN(BETA)
      YE = (R - B / P) * COS(BETA)
      XS = (R - B / P) * SIN(PI / N)
      YS = (R - B / P) * COS(PI / N)
      XL = SQRT((XS - XE)**2 + (YS - YE)**2)

      BETA = PI / N - ATAN(XE/YE)
      LL = INT(XL / SPACNG)

      IF (LL .EQ. 0) THEN
         LL = 1
      ELSEIF (LL .GT. INTRVL) THEN
         LL = INTRVL
      ENDIF
```

```
      DEL = 23.0 / 24.0 * BETA / REAL(LL)
      TH = PI / N + DEL

      DO 100 I = 2, 3000
         TH = TH - DEL
         X(I) = (R - B / P) * SIN(TH)
         Y(I) = (R - B / P) * COS(TH)

         IF (X(I) .LE. XE) THEN
            GOTO 30
         ENDIF

100   CONTINUE

30    X(I) = XE
      Y(I) = YE
      IDTRCH = I
      NN = I + 1
      DEL = 0.01
      TH = -DEL + 1.E-7
*
*
*      DEVELOP TROCHOID CURVE
*
*
      SPMINN = SPMIN
      SPMAXX = SPMAX
35    NNN = NN
      TH = -DEL + 1.E-7
      NTROPT = 0

      DO 200 I = NNN, 3000
         NTROPT = NTROPT + 1
40       TH = TH + DEL

         CALL TROCH (X(I), Y(I), TH)
         XX = X(I) - X(I-1)
         YY = Y(I) - Y(I-1)
         SP = SQRT(XX * XX + YY * YY)

         IF (SP .GE. SPMINN .AND. SP .LE. SPMAXX) THEN
            GOTO 45
         ENDIF

         TH = TH - DEL

         IF (SP .GT. SPMAXX) THEN
            DEL = .99 * DEL
         ELSEIF (SP .LT. SPMINN) THEN
            DEL = 1.01 * DEL
         ENDIF
```

```
         GOTO 40

45       RM = SQRT(X(I)**2 + Y(I)**2)

         IF (RM .GE. RI .AND. NTROPT .LT. INTRVL) THEN
            GOTO 50
         ELSEIF (RM .LT. RI) THEN
            GOTO 200
         ENDIF

         SPMAXX = SPMAXX * 1.1
         SPMINN = SPMAXX * 0.9
         GOTO 35

200   CONTINUE

50    X(I) = XI
      Y(I) = YI
      ITRINV = I
      NN = I + 1
      DEL = .01
      ALPHA = ALPHAI
      RO = R + A / P
*
*
*       DEVELOP INVOLUTE CURVE
*
*
      SPMAXX = SPMAX
      SPMINN = SPMIN
55    NNN = NN
      NINVPT = 0
      ALPHA = ALPHAI - DEL

      DO 300 I = NNN, 3000
         NINVPT = NINVPT + 1
60       ALPHA = ALPHA + DEL

         CALL INVOL(X(I), Y(I), ALPHA)

         XX = X(I) - X(I-1)
         YY = Y(I) - Y(I-1)
         SP = SQRT(XX * XX + YY * YY)

         IF (SP .GE. SPMINN .AND. SP .LE. SPMAXX) THEN
            GOTO 65
         ENDIF

         ALPHA = ALPHA - DEL
```

```
         IF (SP .GT. SPMAXX) THEN
            DEL = 0.99 * DEL
         ELSEIF (SP .LT. SPMINN) THEN
            DEL = 1.01 * DEL
         ENDIF

         GOTO 60

65       RM = SQRT(X(I)**2 + Y(I)**2)

         IF (RM .GE. RO .AND. NINVPT .LT. INTRVL) THEN
            GOTO 70
         ELSEIF (RM .LT. RO) THEN
            GOTO 300
         ENDIF

         SPMAXX = SPMAXX * 1.1
         SPMINN = SPMAXX * 0.9
         GOTO 55
300   CONTINUE

70    NN = I - 1
      IINVAD = I - 1
      NNX = NN
*
*
*           DEVELOP ADDENDUM CIRCLE
*
*
      PHIA = ACOS(R / (R + A / P) * COS(PHI))
      TO = (R+A/P) * (THKNSS/(2. * R) + XINV(PHI) - XINV(PHIA))
      BETA = TO / (R + A / P)
      LL = INT(TO / SPACNG)

      IF (LL .EQ. 0) THEN
         LL = 1
      ELSEIF (LL .GT. INTRVL) THEN
         LL = INTRVL
      ENDIF

      DEL = 1.001 * BETA / REAL(LL)
      TH = BETA + DEL

      DO 400 I = NN, 3000
         TH = TH - DEL
         X(I) = (R + A / P) * SIN(TH)
         Y(I) = (R + A / P) * COS(TH)

         IF (X(I) .LE. 0) THEN
            GOTO 75
         ENDIF
```

```
400   CONTINUE

75    X(I) = 0.0
      Y(I) = R + A / P
      NN = I
      IEND = I

*******************************************************************************
*
*          GENERATE THE 2ND HALF OF TOOTH IN X AND Y
*

      LEND = 2 * IEND - 3
      LL = IEND

      DO 500 I = IEND+1, LEND
         LL = LL - 1
         X(I) = -X(LL)
         Y(I) = Y(LL)
500   CONTINUE

*******************************************************************************
*
*          SAVE SOME OF THE NUMBERS THAT WILL BE USED TO
*          CREATE LINES ON THE CADAM DRAWING
*
      XSAVE(1) = X(2)
      YSAVE(1) = Y(2)
      XSAVE(2) = XE
      YSAVE(2) = YE
      XSAVE(3) = XI
      YSAVE(3) = YI
      XSAVE(4) = X(NNX)
      YSAVE(4) = Y(NNX)
      XSAVE(5) = -X(NNX)
      YSAVE(5) = Y(NNX)
      XSAVE(6) = -XI
      YSAVE(6) = YI
      XSAVE(7) = -XE
      YSAVE(7) = YE
      XSAVE(8) = -X(2)
      YSAVE(8) = Y(2)


*******************************************************************************
*******************************************************************************

6000  WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)' THERE ARE THREE BASIC TYPES OF GEAR BLANK DESIGNS.'
```

```
      WRITE(NO,*)' WHICH TYPE OF GEAR BLANK WOULD YOU LIKE TO DESIGN.'
      WRITE(NO,*)
      WRITE(NO,*)'                1)  PLAIN'
      WRITE(NO,*)'                2)  WEBBED'
      WRITE(NO,*)'                3)  SPOKED '
      WRITE(NO,*)
      WRITE(NO,*)'                     CHOOSE 1, 2 OR 3'

      READ(NI,*) IBLANK

      IF (IBLANK .EQ. 1) THEN
         GOTO 6001
      ELSEIF (IBLANK .EQ. 2) THEN
         GOTO 6002
      ELSEIF (IBLANK .EQ. 3) THEN
         GOTO 6003
      ELSE
         GOTO 6000
      ENDIF


6001  CALL PLAIN (XXX, YYY, HT, ROOTRD, BASERD, PI, PITCHD, OUTDIA, XX2,
     C YY2, ANS4)
      GOTO 4000

6002  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 6000

6003  WRITE(NO,*)
      WRITE(NO,*)' THIS SUBROUTINE IS NOT FUNCTIONAL'
      WRITE(NO,*)
      GOTO 6000

*******************************************************************************
*******************************************************************************

4000  CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)' HOW WOULD YOU LIKE TO DISPLAY THE GRAPHICS?'
      WRITE(NO,*)
      WRITE(NO,*)'        1)  GKS (GRAPHICAL KERNEL SYSTEM)'
      WRITE(NO,*)'        2)  CADAM'
      WRITE(NO,*)'        3)  MOVIE.BYU'
      WRITE(NO,*)'        4)  NO DISPLAY (ONLY DATA OUTPUT TO FILE)'
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)'          CHOOSE 1,2,3 OR 4   (5 TO QUIT)'
```

```
      READ(NI,*) IGRAPH

      IF (IGRAPH .EQ. 1) THEN
         CALL GKS (X, Y, XXX, YYY, XX2, YY2, PI, N, LEND, ANS4, IEND)
         GOTO 4999
      ELSEIF (IGRAPH .EQ. 2) THEN
         CALL CADCAM(X,Y,LEND,N,XSAVE,YSAVE)
         GOTO 4999
      ELSEIF (IGRAPH .EQ. 3) THEN
         CALL MOVIE (X, Y, LEND, PI, N, XSAVE, YSAVE, IEND)
         GOTO 4999
      ELSEIF (IGRAPH .EQ. 4) THEN
         CALL DATA (ANS4)
         GOTO 4999
      ELSEIF (IGRAPH .EQ. 5) THEN
         GOTO 4999
      ELSE
         GOTO 4000
      ENDIF

4999  END
```

```
C*************************************************************************
C*************************************************************************
C**********                                                     **********
C*********                    INVOLUTE FUNCTION                 *********
C**********                                                     **********
C*************************************************************************
C*************************************************************************

      FUNCTION XINV(X)
      XINV = TAN(X) - X
      RETURN
      END


C*************************************************************************
C*************************************************************************
C*********                                                     *********
C******       THIS SUBROUTINE FINDS THE INTERSECTION OR POINT      ******
C******    OF TANGENCY BETWEEN THE INVOLUTE AND TROCHOID CURVES    ******
C*********                                                     *********
C*************************************************************************
C*************************************************************************

      SUBROUTINE INTRSC(RI, X, Y, AOUT, TOUT, ITER, IDBUG, IERR)

C
C        THIS ROUTINE WILL LOCATE THE POINT OF TANGENCY OR THE
C        POINT OF INTERSECTION OF THE INVOLUTE TOOTH FLANK AND
C        TROCHOIDAL TOOTH ROOT FOR STANDARD OR NON-STANDARD
C        EXTERNAL INVOLUTE GEAR TEETH
C
C
C        ARGUMENTS:
C    RI       RADIUS OF INTERSECTION/TANGENCY POINT
C    X,Y      COORDINATES OF THE POINT OF INTERSECTION/TANGENCY
C    AOUT     ALPHA OF THE INVOLUTE         INTERSECTION
C    TOUT     THETA OF THE TROCHOID         INTERSECTION
C    ITER     ITERATION COUNT TO SOLUTION
C    IDBUG    0 FOR NO OUTPUT OF SUBROUTINE
C             1 FOR OUTPUT AT EACH ITERATION
C    IERR     0 FOR REASONABLE OUTPUT
C             1 FOR RI IN AN UNREASONABLE RANGE
C             99 FOR LACK OF CONVERGENCE
C
C
C         SUBROUTINES CALLED:  DERIV, INVOL, TROCH

      IMPLICIT REAL*8 (A-H, O-Z)

      INTEGER IERR, ITER
```

```
      REAL N

      COMMON/TOOTH/N,P,A,B,RHF,THKNSS,PHI,R

      REAL*4 X,Y,AOUT,TOUT,XI1S,YI1S,XT1S,YT1S,F1AS,F1TS,
     C       F2AS,F2TS,RI,P,B,RHF,THKNSS,PHI,R,A,RO,RMIN

      IERR = 0
      ITER = 0
      FACTOR = 1.D0
      EPS = 1.D-4
*
*
*         APPROXIMATE ALPHA AND THETA TO BEGIN ITERATION AND
*         SET THE HISTORY VARIABLES
*
*
*     ALPHA = .3D0 * (1.D0 - DEXP(-(DBLE(N)-20.D0)/25.D0))
      ALPHA = .70
      ALPHAP = ALPHA
      THETA = .3D0 * (DEXP(-(DBLE(N)-20.D0)/45.D0))
      THETAP = THETA
*
*
*           BEGIN THE LOOP
*
*
*         THIS LOOP LOCATES THE POINT OF TANGENCY OR INTERSECTION
*         BETWEEN THE INVOLUTE AND THE TROCHOID
*
30    ITER = ITER + 1

*         TEST ITERATION COUNTER FOR LACK OF CONVRGENCE
*
      IF (ITER .GT. 2000) THEN
         GOTO 36
      ENDIF
*
*         SET THE FACTORS TO DAMPEN OSCILLATIONS
*
      IF (ITER .GE. 10) THEN
         FACTOR = .10D0
      ENDIF
      IF (ITER .GE. 50) THEN
         FACTOR = .01D0
      ENDIF
      IF (ITER .GE. 200) THEN
         FACTOR = .001D0
```

```
      ENDIF
*
*        GET THE DERIVATIVES OF XI-XT (F1) AND YI-YT (F2)
*        SINGLE PRECISION
*
      CALL DERIV (REAL(ALPHA), REAL(THETA),F1AS,F1TS,F2AS,F2TS)
*****************************************************************************


      F1A = DBLE(F1AS)
      F1T = DBLE(F1TS)
      F2A = DBLE(F2AS)
      F2T = DBLE(F2TS)
*
*        FORM THE JACOBIAN
*
      DJAC = F1A * F2T - F2A * F1T
*
*        GET THE CURRENT LOCATIONS ON THE INVOLUTE AND TROCHOID
*
      CALL INVOL (XI1S, YI1S, REAL(ALPHA))
      CALL TROCH (XT1S, YT1S, REAL(THETA))


      XI1 = DBLE(XI1S)
      YI1 = DBLE(YI1S)
      XT1 = DBLE(XT1S)
      YT1 = DBLE(YT1S)
      DTEMP1 = (YI1 - YT1) * F1T - (XI1 - XT1) * F2T
      DTEMP2 = (XI1 - XT1) * F2A - (YI1 - YT1) * F1A
*
*         COMPUTE THE NEWTON-RHAPSON CORRECTIONS
*
      DELA = DTEMP1 / DJAC * FACTOR
      DELTH = DTEMP2 / DJAC * FACTOR
      ALPHA = ALPHA + DELA
      THETA = THETA + DELTH
*
*         COMPUTE ERRORS AND COMPARE WITH ERROR CRITERION
*
      EPSA = ABS((ALPHA - ALPHAP)/ALPHA)
      EPST = ABS((THETA - THETAP)/THETA)
*
*         IF BOTH ERRORS ARE SMALL THEN EXIT
*
      IF (EPST .LT. EPS .AND. EPSA .LT. EPS) THEN
         GOTO 35
      ENDIF
*
```

```
*          IF DEBUG SWITCH .NE. 0 WRITE ITERATION RESULTS
*
      IF (IDBUG .NE. 0) THEN
         WRITE (NO,32) ITER, ALPHA, THETA
32       FORMAT(1X,'INTERSECT  ***  ITER= ',I5,' ALPHA= ',D15.8,
     C           ' THETA= ',D15.8)
      ENDIF

      ALPHAP = ALPHA
      THETAP = THETA
*
*          DO NOT LET ALPHA OR THETA GO NEGATIVE
*
      IF (ALPHA .LT. 0) THEN
         ALPHA = 1.D-10
      ENDIF
      IF (THETA .LT. 0) THEN
         THETA = 1.D-10
      ENDIF
*
*         START LOOP AGAIN
*
      GOTO 30
*
*         EXIT LOOP
*
35    ALPHAS = REAL(ALPHA)

      CALL INVOL(X,Y,ALPHAS)

      AOUT = REAL(ALPHA)
      TOUT = REAL(THETA)
      RI = SQRT(X*X + Y*Y)
      RO = R + A / P
      RMIN = R - B / P
      IERR = 1

      IF (RI .GE. RMIN .AND. RI .LE. RO) THEN
         IERR = 0
      ENDIF

*
*          LOOP FOR CONVERGENCE FAILURE
*
36    IF (ITER .GT. 2000) THEN
         IERR = 99
      ENDIF
37    RETURN
      END
```

```
**********************************************************************************
**********************************************************************************
*********                                                          *********
******        THIS SUBROUTINE COMPUTES THE TROCHOID POINTS          ******
*********                                                          *********
**********************************************************************************
**********************************************************************************

      SUBROUTINE TROCH(XTR,YTR,TH)
*
*
*          COMPUTE THE LOCATION OF A POINT ON THE TROCHOID
*          (XTR, YTR) GIVEN THE PARAMETER TH (THETA)
*
      REAL N, PI, P, A, B, RHF, THKNSS, PHI, R, DELTA, ETA, BETA,
     CXTR, YTR, TH

      COMMON/TOOTH/N,P,A,B,RHF,THKNSS,PHI,R
*
*         TROCHOID
*
      PI = 3.1415927
      DELTA = ((PI/P-THKNSS)/2.)-(B/P-RHF/P)*TAN(PHI)-RHF/(P*COS(PHI))
      ETA = DELTA/R
      BETA = PI/N - ETA

      XTR = -TH*R*COS(BETA+TH)+(R-B*P**(-1)+P**(-1)*RHF)*SIN(BETA+
     CTH)-P**(-1)*RHF*(TH**(-1)*R**(-1)*P**(-1)*(B-RHF)*(TH**(-2)*
     CR**(-2)*P**(-2)*(B-RHF)**2+1.)**(-.5)*SIN(BETA+TH)+(TH**(-2)*R
     C**(-2)*P**(-2)*(B-RHF)**2+1.)**(-.5)*COS(BETA+TH))

      YTR = TH*R*SIN(BETA+TH)+(R-B*P**(-1)+P**(-1)*RHF)*COS(BETA+TH)+
     CP**(-1)*RHF*(-TH**(-1)*R**(-1)*P**(-1)*(B-RHF)*(TH**(-2)*
     CR**(-2)*P**(-2)*(B-RHF)**2+1.)**(-.5)*COS(BETA+TH)+(TH**(-2)*
     CR**(-2)*P**(-2)*(B-RHF)**2+1.)**(-.5)*SIN(BETA+TH))

      RETURN
      END
```

```
**********************************************************************************
**********************************************************************************
*********                                                              *********
******        THIS SUBROUTINE COMPUTES THE INVOLUTE POINTS             ******
*********                                                              *********
**********************************************************************************
**********************************************************************************

      SUBROUTINE INVOL(XINV,YINV,ALPHA)
*
*
*          COMPUTE THE LOCATION OF A POINT ON THE INVOLUTE
*          (XINV, YINV) GIVEN THE PARAMETER ALPHA
*
*
      REAL N, P, A, B, RHF, THKNSS, PHI, R, XINV, YINV, ALPHA

      COMMON/TOOTH/N,P,A,B,RHF,THKNSS,PHI,R
*
*          INVOLUTE
*
      XINV = R*COS(PHI)/COS(ALPHA)*SIN(((THKNSS/2.)/R)
     C        +TAN(PHI)-PHI-TAN(ALPHA) + ALPHA)

      YINV = R*COS(PHI)/COS(ALPHA)*COS(((THKNSS/2.)/R)
     C        +TAN(PHI)-PHI-TAN(ALPHA) + ALPHA)

      RETURN
      END
```

```
**********************************************************************************
**********************************************************************************
**********                                                            **********
******        THIS SUBROUTINE IS USED TO COMPUTE DERIVATIVES          ******
**********                                                            **********
**********************************************************************************
**********************************************************************************

      SUBROUTINE DERIV(ALPHA,THETA,F1A,F1T,F2A,F2T)
*
*         FORM F1 AS XINV-XTROCH
*              F2 AS YINV-YTROCH
*         TAKE DERIVATIVES W/R ALPHA -- F1A AND F2A
*                              THETA -- F1T AND F2T
*
      REAL N, P, A, B, RHF, THKNSS, PHI, R, ALPHA, THETA, F1A, F1T,
     CF2A, F2T, XI1, YI1, XI2, YI2, XT1, YT1, XT2, YT2, ALDEL, THDEL

      COMMON/TOOTH/N,P,A,B,RHF,THKNSS,PHI,R

      ALDEL = ALPHA + .001
      THDEL = THETA + .001

      CALL INVOL (XI1, YI1, ALPHA)
      CALL INVOL (XI2, YI2, ALDEL)
      CALL TROCH (XT1, YT1, THETA)
      CALL TROCH (XT2, YT2, THDEL)
      F1A = (XI2-XI1)*1000.
      F1T = -(XT2-XT1)*1000.
      F2A = (YI2-YI1)*1000.
      F2T = -(YT2-YT1)*1000.

      RETURN
      END
```

APPENDIX B. GEAR BLANK SUBROUTINE

```
************************************************************************************
************************************************************************************
**********                                                              **********
********      THIS SUBROUTINE SETS UP THE GEAR BLANK GEOMETRY           *******
********      FOR A PLAIN TYPE BLANK WITH A HUB AS AN OPTION            *******
**********                                                              **********
************************************************************************************
************************************************************************************

      SUBROUTINE PLAIN(XXX, YYY, HT, ROOTRD, BASERD, PI, PITCHD,
     C OUTDIA, XX2, YY2, ANS4)

*          THIS SUBROUTINE WILL SET UP THE DATA FOR
*          THE SHAPE OF A 'PLAIN' GEAR BLANK.
*
*        THERE ARE THREE BASIC TYPES OF GEAR BLANK DESIGNS
*            1) PLAIN     2) WEBBED    3) SPOKED
*        THIS PROGRAM WILL ONLY DEAL WITH PLAIN GEAR BLANKS


*     ARGUMENTS:
*  AMOUNT =  DISTANCE THE HUB PROTRUDES FROM THE GEAR BLANK FACE
*  ANS3   =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ANS4   =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ANS5   =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  ANS6   =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  BASERD =  BASE RADIUS OF GEAR
*  BORECK =  VARIABLE WHICH CHECKS THE SIZE OF THE BORE DIAMETER CHOSEN
*  DPHI   =  ANGLE USED IN CONSTRUCTION OF BORE AND HUB CIRCLES (IN
*            DEGREES)
*  FACE   =  FACE WIDTH OF THE GEAR
*  HOLDIA =  DIAMETER OF THE BORE HOLE
*  HOLRAD =  RADIUS OF THE BORE HOLE
*  HT     =  WHOLE DEPTH OF GEAR TOOTH
*  HUBCHK =  VARIABLE WHICH CHECKS THE SIZE OF THE HUB AGAINST THE HUB
*            THICKNESS
*  HUBCK2 =  VARIABLE WHICH CHECKS THE SIZE OF THE HUB AGAINST THE ROOT
*            RADIUS
*  HUBDIA =  DIAMETER OF THE HUB
*  HUBRAD =  RADIUS OF THE HUB
*  HUBTHK =  THICKNESS OF HUB (HUB DIAMETER MINUS BORE DIAMETER)
*  I      =  INTEGER COUNTER
*  KEYWAY =  KEYWAY DEPTH IF A KEYWAY WAS TO EXIST - ALSO USED TO SIZE
*            HUB
*  MIDSTP =  INTEGER VARIABLE USED AS A MIDDLE STEP IN HUB DIAMETER
```

```
*            CALCULATIONS
*  NI      =  INTEGER USED TO READ DATA FROM THE SCREEN
*  NO      =  INTEGER USED TO WRITE DATA OR STATEMENTS TO THE SCREEN
*  OUTDIA =  OUTSIDE DIAMETER OF GEAR
*  PI      =  3.1415927
*  PITCHD =  CUTTING PITCH DIAMETER OF GEAR
*  PROTRU =  INTEGER NUMBER   0 - NO HUB ON GEAR
*                             1 - HUB ON ONE SIDE OF GEAR
*                             2 - HUB ON BOTH SIDES OF GEAR
*  ROOTRD =  ROOT RADIUS OF GEAR
*  RPHI    =  ANGLE USED IN CONSTRUCTION OF BORE AND HUB CIRCLES (IN
*            RADIANS)
*  XXX     =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  YYY     =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  XX2     =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*            GEOMETRY
*  YY2     =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*            GEOMETRY


*  VARIABLE DECLARATIONS
      INTEGER PROTRU, I, MIDSTP, NO, NI

      REAL AMOUNT,FACE, HOLDIA, HOLRAD, BORECK, ROOTRD, BASERD, DPHI,
     C RPHI, PI, XXX(722), YYY(722), XX2(722), YY2(722), PITCHD, KEYWAY,
     C HUBDIA, HUBRAD, HUBTHK, HUBCHK, HUBCK2, HT, OUTDIA

      CHARACTER*1 ANS3, ANS4, ANS5, ANS6
      CHARACTER*5 CLE

      COMMON /INTPRO/ PROTRU
      COMMON /RELPRO/ AMOUNT, HOLRAD, FACE, HUBRAD

      DATA NO/6/, NI/5/
      DATA CLE /'CLEAR'/

      CALL SYSCAL (CLE, 5, IERR)
      WRITE(NO,*)
      WRITE(NO,*)'>>> PLAIN '
      WRITE(NO,*)
      WRITE(NO,*)
      WRITE(NO,*)'ENTER THE FACE WIDTH OF THE GEAR.'
      READ(NI,*) FACE



*   THE HUB PROTRUSION IS THE SAME AS A HUB EXTENSION.

79    WRITE(NO,*)
```

```
      WRITE(NO,*)'DOES THE GEAR HAVE A HUB PROTRUSION (Y OR N)?'
      READ(NI,80) ANS4
80    FORMAT (A1)

      IF (ANS4 .EQ. 'N') THEN
         PROTRU = 0
         HUBRAD = 0.
      ENDIF

      IF (ANS4 .EQ. 'Y') THEN
81       WRITE(NO,*)
         WRITE(NO,*)' ON BOTH SIDES OF THE GEAR OR ONLY ONE SIDE?'
         WRITE(NO,*)'      CHOOSE "1" FOR ONLY ONE SIDE OR '
         WRITE(NO,*)'             "2" FOR BOTH SIDES '
         READ(NI,*) PROTRU

         IF (PROTRU .NE. 1 .AND. PROTRU .NE. 2) THEN
            GOTO 81
         ENDIF

82       WRITE(NO,*)
         WRITE(NO,*)' HOW FAR DOES THE PROTRUSION EXTEND FROM THE'
         WRITE(NO,*)'        FLUSH FACE OF THE GEAR?'
         READ(NI,*) AMOUNT

         IF (AMOUNT .GT. FACE) THEN
            WRITE(NO,*)
            WRITE(NO,*)'THE PROTRUSION IS GREATER THAN THE FACE WIDTH.'
            WRITE(NO,*)'  IS THIS WHAT YOU MEANT TO INPUT (Y OR N)?'
            READ(NI,84) ANS5
84          FORMAT (A1)

            IF (ANS5 .EQ. 'N') THEN
               WRITE(NO,*)
               WRITE(NO,*)'REENTER THE PROTRUSION AMOUNT'
               WRITE(NO,*)
               GOTO 82
            ENDIF

         ENDIF

      ENDIF

86    WRITE(NO,*)
      WRITE(NO,*)'ENTER THE BORE DIAMETER OF THE HOLE IN THE BLANK.'
      READ(NI,*) HOLDIA

*    THIS SECTION CHECKS TO SEE IF THE RADIUS OF THE BORE IN THE BLANK
*    IS LARGER THAN THE BASE RADIUS OR ROOT RADIUS.

      HOLRAD = HOLDIA / 2.0
```

```
      BORECK = ROOTRD * 0.80

      IF (HOLRAD .GT. BASERD .OR. HOLRAD .GT. ROOTRD) THEN
         WRITE(NO,88)BASERD, ROOTRD
88     FORMAT(6X,'THE BASE RADIUS = ',F8.3,4X,'THE ROOT RADIUS = ',F8.3)
         WRITE(NO,*)' ***** THE RADIUS OF THE HOLE IN THE BLANK IS'
         WRITE(NO,*)'          LARGER THAN THE BASE OR ROOT RADIUS'
         WRITE(NO,*)'               ENTER A SMALLER VALUE.'
         WRITE(NO,*)
         GOTO 86
      ENDIF

      IF (HOLRAD .GT. BORECK) THEN
         WRITE(NO,*)
         WRITE(NO,*)' ***** THE RADIUS OF THE HOLE IN THE BLANK SEEMS'
         WRITE(NO,*)'              LARGE.  IT IS GREATER THAN'
         WRITE(NO,*)'                (0.8 X THE ROOT RADIUS)'
         WRITE(NO,*)
         WRITE(NO,*)'WOULD YOU LIKE TO CHANGE THE BORE RADIUS (Y OR N)?'
         READ(NI,90) ANS3
90       FORMAT (A1)

         IF (ANS3 .EQ. 'Y') THEN
            GOTO 86
         ENDIF

      ENDIF

      DPHI = -0.5
      DO 425 I = 1, 722

         DPHI = DPHI + 0.5
         RPHI = PI * DPHI / 180.
         XXX(I) = HOLRAD * COS(RPHI)
         YYY(I) = HOLRAD * SIN(RPHI)
425    CONTINUE


*     THE LIST BELOW IS AGMA'S RECOMMENDED KEYWAYS FOR BORES IN GEARING
*     THIS CHART LISTS A KEYWAY DEPTH THAT IS SOMETIMES USED TO
*     CALCULATE THE HUB THICKNESS OF A GEAR BY THE EQUATION
*          HUB THICKNESS = 2.5 X KEY SEAT DEPTH


      IF (HOLDIA .LT. .3124) THEN
         KEYWAY = 0.0
      ELSEIF (HOLDIA .GE. 0.3124 .AND. HOLDIA .LT. 0.5000) THEN
         KEYWAY = 0.046875
      ELSEIF (HOLDIA .GE. 0.5000 .AND. HOLDIA .LT. 0.6250) THEN
         KEYWAY = 0.06250
      ELSEIF (HOLDIA .GE. 0.6250 .AND. HOLDIA .LT. 0.9375) THEN
```

```
         KEYWAY = 0.09375
       ELSEIF (HOLDIA .GE. 0.9375 .AND. HOLDIA .LT. 1.3125) THEN
         KEYWAY = 0.12500
       ELSEIF (HOLDIA .GE. 1.3125 .AND. HOLDIA .LT. 1.4375) THEN
         KEYWAY = 0.15625
       ELSEIF (HOLDIA .GE. 1.4375 .AND. HOLDIA .LT. 1.8125) THEN
         KEYWAY = 0.18750
       ELSEIF (HOLDIA .GE. 1.8125 .AND. HOLDIA .LT. 2.3125) THEN
         KEYWAY = 0.25000
       ELSEIF (HOLDIA .GE. 2.3125 .AND. HOLDIA .LT. 2.8125) THEN
         KEYWAY = 0.31250
       ELSEIF (HOLDIA .GE. 2.8125 .AND. HOLDIA .LT. 3.3125) THEN
         KEYWAY = 0.37500
       ELSEIF (HOLDIA .GE. 3.3125 .AND. HOLDIA .LT. 3.8125) THEN
         KEYWAY = 0.43750
       ELSEIF (HOLDIA .GE. 3.8125 .AND. HOLDIA .LT. 4.5625) THEN
         KEYWAY = 0.50000
       ELSEIF (HOLDIA .GE. 4.5625 .AND. HOLDIA .LT. 5.5625) THEN
         KEYWAY = 0.43750
       ELSEIF (HOLDIA .GE. 5.5625 .AND. HOLDIA .LT. 6.5625) THEN
         KEYWAY = 0.50000
       ELSEIF (HOLDIA .GE. 6.5625 .AND. HOLDIA .LT. 7.5625) THEN
         KEYWAY = 0.62500
       ELSEIF (HOLDIA .GE. 7.5625 .AND. HOLDIA .LT. 9.0000) THEN
         KEYWAY = 0.75000
       ELSEIF (HOLDIA .GE. 9.0000 .AND. HOLDIA .LT. 11.0000) THEN
         KEYWAY = 0.87500
       ELSEIF (HOLDIA .GE. 11.0000 .AND. HOLDIA .LT. 13.0000) THEN
         KEYWAY = 1.00000
       ELSEIF (HOLDIA .GE. 13.0000 .AND. HOLDIA .LT. 15.0000) THEN
         KEYWAY = 1.25000
       ELSEIF (HOLDIA .GE. 15.0000 .AND. HOLDIA .LT. 18.0000) THEN
         KEYWAY = 1.50000
       ELSEIF (HOLDIA .GE. 18.0000 .AND. HOLDIA .LT. 21.0000) THEN
         KEYWAY = 1.75000
       ELSEIF (HOLDIA .GE. 21.0000) THEN
         KEYWAY = 2.00000
*        KEYWAY = 2.0 + 0.25 * (INT(HOLDIA / 3.) - 7.)
       ENDIF

*    HUB DIAMETERS (FROM PHILA. GEAR CATALOG):
*        STEEL AND HIGH TENSILE BRONZE     1.6 X BORE DIAMETER
*        CAST IRON AND BRONZE              1.8 X BORE DIAMETER
*
*      TO BE ON THE SAFE SIDE, USE 1.8 X THE BORE DIAMETER
*      FOR THIS PROGRAM SINCE IT DOES NOT DEAL WITH THE TYPE
*      OF MATERIAL THE GEAR IS MADE OUT OF AS OF YET.

       HUBDIA = 1.8 * HOLDIA

       HUBTHK = 2.5 * KEYWAY
```

```
      HUBCHK = (HUBDIA - HOLDIA) / 2.0

      IF (ANS4 .EQ. 'N') THEN
         GOTO 455
      ENDIF


         IF (HUBCHK .LT. HUBTHK) THEN
            HUBDIA = (HUBTHK * 2.0) + HOLDIA
         ENDIF

*    HUB DIAMETERS UP TO 8 INCHES TAKE TO THE NEAREST +0.125 INCHES
*                  8 TO 16 INCHES TAKE TO THE NEAREST +0.250 INCHES
*                  OVER 16 INCHES TAKE TO THE NEAREST +0.500 INCHES

      IF (HUBDIA .LT. 8.0) THEN
         MIDSTP = INT((HUBDIA + 0.1249) / 0.125)
         HUBDIA = MIDSTP * 0.125
      ELSEIF (HUBDIA .GE. 8.0 .AND. HUBDIA .LE. 16.0) THEN
         MIDSTP = INT((HUBDIA + 0.2499) / 0.25)
         HUBDIA = MIDSTP * 0.25
      ELSEIF (HUBDIA .GT. 16.0) THEN
         MIDSTP = INT((HUBDIA + 0.4999) / 0.5)
         HUBDIA = MIDSTP * 0.5
      ENDIF


* THIS CHECKS TO MAKE SURE THAT THE GEAR BLANK IS GEOMETRICALLY POSSIBLE


      HUBCK2 = (ROOTRD * 0.95) * 2.0
      IF (HUBDIA .GE. HUBCK2) THEN
         WRITE(NO,*) ' **** THE HUB DIAMETER IS TOO LARGE.  IT IS'
         WRITE(NO,*) '      GREATER THAN (0.95 X THE ROOT DIAMETER).'
         WRITE(NO,*)
         WRITE(NO,*) '            REENTER GEAR BLANK DATA.'
         WRITE(NO,*)
         GOTO 79
      ENDIF
      HUBRAD = HUBDIA / 2.0
      IF (ANS4 .EQ. 'Y') THEN

         DPHI = -0.5
         DO 445 I = 1, 722

            DPHI = DPHI + 0.5
            RPHI = PI * DPHI / 180.
            XX2(I) = HUBRAD * COS(RPHI)
            YY2(I) = HUBRAD * SIN(RPHI)
445      CONTINUE
```

```
      ENDIF

455   RETURN
      END
```

APPENDIX C. GKS SUBROUTINE

```
*****************************************************************************************
*****************************************************************************************
**********                                                                     **********
*******      THIS SUBROUTINE DRAWS THE GEAR GEOMETRY IN 2-D                    *******
*******            USING GKS (GRAPHICS KERNAL SYSTEM)                          *******
**********                                                                     **********
*****************************************************************************************
*****************************************************************************************

      SUBROUTINE GKS(X,Y,XXX,YYY,XX2,YY2,PI,N,LEND,ANS4,IEND)


*     ARGUMENTS:
*  ANGLE  =  VARIABLE ANGLE USED IN CONSTRUCTING ENTIRE GEAR GEOMETRY
*            FROM TOOTH GEOMETRY
*  ANGL2  =  VARIABLE ANGLE USED IN CONSTRUCTING ENTIRE GEAR GEOMETRY
*            FROM TOOTH GEOMETRY
*  ANS4   =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
*  FINAM  =  10-CHARACTER STRING FOR A NAME OF A FILE
*  I      =  INTEGER COUNTER
*  IEND   =  INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*            BETWEEN CENTERLINE OF THE TOOTH AND ADDENDUM CIRCLE
*  INCR1  =  RADIAN INCREMENT ANGLE USED TO STEP GEAR TOOTH GEOMETRY
*            AROUND CIRCLE TO CREATE ENTIRE GEAR GEOMETRY
*  INCR2  =  ANGLE USED TO STEP GEAR TOOTH GEOMETRY AROUND
*            CIRCLE TO CREATE ENTIRE GEAR GEOMETRY
*  J      =  LIST ELEMENT OF WORKSTATION TYPES
*  JJ     =  INTEGER COUNTER
*  KERROR =  ERROR INDICATOR:  0  - REQUESTED VALUE REPORTED
*                             >0 - REFER TO ERROR MESSAGE
*  KJUNK  =  NUMBER OF AVAILABLE WORKSTATION TYPES OR WORKSTATION TYPE
*            OF ELEMENT
*  KNUM   =  NUMBER OF AVAILABLE WORKSTATION TYPES
*  KTYPE  =  WORKSTATION TYPE OF ELEMENT
*  KWKTYP =  WORKSTATION/TERMINAL TYPE
*  LEND   =  INTEGER USED TO SAVE THE COORDINATES OF THE LAST POINT ON
*            THE FIRST TOOTH GEOMETRY GENERATED.  ALSO SAVES THE NUMBER
*            OF POINTS PER TOOTH.
*  N      =  NUMBER OF TEETH IN GEAR
*  NI     =  INTEGER USED TO READ DATA FROM SCREEN
*  NO     =  INTEGER USED TO WRITE DATA OR STATEMENTS FROM SCREEN
*  O      =  LIST ELEMENT OF WORKSTATION TYPES
*  PI     =  3.1415927
*  RADIUS =  RADIUS USED IN CONSTRUCTING ENTIRE GEAR GEOMETRY FROM
*            TOOTH GEOMETRY
```

```
*  TYPE   =  7-CHARACTER STRING TO CONTINUE WITH PROGRAM OR LOOP
*            FOR CORRECT WORKSTATION INPUT
*  WINDOW =  THE SIZE OF THE WINDOW USED FOR GKS (FUNCTION OF Y(IEND))
*  X      =  X-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  Y      =  Y-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  XXX    =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  YYY    =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE BORE
*            HOLE GEOMETRY
*  XX2    =  X-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*            GEOMETRY
*  YY2    =  Y-COORDINATE OF THE SYSTEM USED TO GENERATE THE HUB
*            GEOMETRY
*  X1     =  X-COORDINATE USED TO DRAW THE ENTIRE GEAR GEOMETRY
*  Y1     =  Y-COORDINATE USED TO DRAW THE ENTIRE GEAR GEOMETRY


*     GKS SUBROUTINES CALLED:  GACWK, GCLKS, GCLSG, GCLWK, GCRSG,
*                              GDAWK, GOPKS, GOPWK, GPL, GQEWK,
*                              GSELNT, GSWN, UOPNSF


*  VARIABLE DECLARATIONS
      INTEGER KJUNK, I, J, KTYPE, JJ, LEND, NO, NI, JERROR, KERFIL,
     C        KWKTYP, KERROR, KNUM, IEND

      REAL X(3000), Y(3000), XXX(722), YYY(722), INCR1, INCR2, N, PI,
     C     ANGLE(3000), RADIUS(3000), ANGL2(3000), X1(3000), Y1(3000),
     C     XX2(722), YY2(722), WINDOW

      CHARACTER*1 ANS4
      CHARACTER*7 TYPE

      DATA NO/6/, NI/5/


      CALL UOPNSF(1,'ERROR DATA',1,JERROR)
      IF (JERROR .NE. 0) THEN
         WRITE(*,*)'CANNOT OPEN ERROR LOGGING FILE ',JERROR
         STOP
      END IF
      KERFIL = 1

*  OPEN THE FILE FOR READING AND WRITING
*
      OPEN (UNIT = 20, FILE = 'ERRFIL')
*
*  OPEN GKS
*
```

```
      CALL GOPKS (20)
*
*  THIS SECTION ALLOWS THE USER TO INPUT THE WORKSTATION TYPE.
*
74    WRITE (5, *) 'PLEASE ENTER TERMINAL TYPE'
      READ (5, FMT=111) KWKTYP
111   FORMAT (I6)
*
*  VALIDATE WORKSTATION TYPE.  LOOP UNTIL VALID TYPE IS INPUT
*
      CALL GQEWK (0,KERROR,KNUM,KJUNK)
      TYPE = 'INVALID'
      DO 222 I=1,KNUM
        J=I
        CALL GQEWK (J,KERROR,KJUNK,KTYPE)
      IF (KWKTYP .EQ. KTYPE) TYPE = 'VALID'
222   CONTINUE
      IF (TYPE .EQ. 'INVALID') GOTO 74
*
*  OPEN AND ACTIVATE WORKSTATION 1, USE LUN 5, AND KWKTYP FOR TYPE
*
      CALL GOPWK (1,5,KWKTYP)
      CALL GACWK (1)

*
*  THIS SETS UP THE WINDOW SIZE FOR THE SCREEN.  COORDINATES FOR EACH
*  AXIS ARE CALCULATED FROM MAX(Y) ON THE GEAR TOOTH GEOMETRY
*
      WINDOW = Y(IEND) + 2.0

      CALL GSWN (1,-(WINDOW),WINDOW,-(WINDOW),WINDOW)
      CALL GSELNT (1)
*
*
*  CREATE A SEGMENT AND DRAW THE GEOMETRY
*
      CALL GCRSG(10)

      CALL GPL(722,XXX,YYY)

      IF (ANS4 .EQ. 'Y') THEN
         CALL GPL(722,XX2,YY2)
      ENDIF

      DO 500 I = 2, LEND
         X1(I-1) = X(I)
         Y1(I-1) = Y(I)
500   CONTINUE

      CALL GPL (LEND-1,X1,Y1)
*
```

```
*       STEP THE GEAR TOOTH PROFILE AROUND, MAKING THE WHOLE GEAR.
*
      INCR1 = 2. * PI / N
      INCR2 = INCR1

      DO 700 JJ = 1,N

         DO 600 I = 2,LEND

            ANGLE(I-1) = ATAN2(Y(I),X(I))
            RADIUS(I-1) = SQRT(X(I)*X(I) + Y(I)*Y(I))
            ANGL2(I-1) = ANGLE(I-1) + INCR1
            X1(I-1) = RADIUS(I-1) * COS(ANGL2(I-1))
            Y1(I-1) = RADIUS(I-1) * SIN(ANGL2(I-1))

600      CONTINUE

         CALL GPL(LEND-1,X1,Y1)
         INCR1 = INCR1 + INCR2

700   CONTINUE


      CALL GCLSG
*
*  CLOSE UP EVERYTHING
*
      CALL GDAWK (1)
      CALL GCLWK (1)
      CALL GCLKS
      CLOSE (20)


99    STOP
      END
```

APPENDIX D. CADAM-CADCD SUBROUTINE

```
*****************************************************************************************
*****************************************************************************************
**********                                                                     *********
********      THIS SUBROUTINE DRAWS THE GEAR GEOMETRY IN 3-D                  *******
********              USING CADCD - CADAM SUBROUTINES                         *******
**********                                                                     *********
*****************************************************************************************
*****************************************************************************************

      SUBROUTINE CADCAM(X,Y,LEND,N,XSAVE,YSAVE)

*        CADAM GEOMETRY INTERFACING
*        CALLING PROGRAM  - -  CADCD GEAR


*     ARGUMENTS:
*  AA      =  COEFFICIENT OF X-COORDINATE OF 'CIRNRM' VECTOR
*  AINCR   =  INCREMENT OF DO-LOOPS
*  AMOUNT  =  DISTANCE THE HUB PROTRUDES FROM THE GEAR BLANK FACE
*  ANGLE   =  ANGLE USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  ANGLE1  =  ANGLE USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  ANGLE2  =  ANGLE USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  ANGL1   =  ANGLE USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  ANGL2   =  ANGLE USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  API     =  VALUE LESS THAN (2 X PI) USED IN STOPPING DO-LOOP EXECUTION
*  APT1    =  END POINT VARIABLES USED TO HELP CONSTRUCT ARCS
*  APT2    =  END POINT VARIABLES USED TO HELP CONSTRUCT ARCS
*  ARCPT1  =  AN ARRAY OF LENGTH 3 THAT CONTAINS THE X,Y,Z ABSOLUTE
*             COORDINATES OF ONE END POINT OF THE ARC
*  ARCPT2  =  AN ARRAY OF LENGTH 3 THAT CONTAINS THE X,Y,Z ABSOLUTE
*             COORDINATES OF THE OTHER END POINT OF THE ARC
*  BB      =  COEFFICIENT OF Y-COORDINATE OF 'CIRNRM' VECTOR
*  CC      =  COEFFICIENT OF Z-COORDINATE OF 'CIRNRM' VECTOR
*  CENTER  =  AN ARRAY, LENGTH 3, THAT CONTAINS THE X,Y,Z COORDINATES
*             OF THE CENTER OF THE CIRCLE
*  CENTR1  =  AN ARRAY OF LENGTH 3 THAT CONTAINS THE X,Y,Z ABSOLUTE
*             COORDINATES OF THE CENTER OF THE CIRCLE IN WHICH THE ARC
*             IS CONTAINED
*  CIRNRM  =  AN ARRAY OF LENGTH 3 THAT CONTAINS THE AA,BB,CC
*             COEFFICIENTS OF A VECTOR NORMAL TO THE PLANE OF THE CIRCLE
*  D       =  A FLOATING POINT ARRAY CONTAINING SPLINE POINTS COORDINATE
*             DATA
*  DRAWID  =  5-WORD ARRAY CONTAINING THE DRAWING ID
*  ENDPT1  =  AN ARRAY (3) CONTAINING THE X,Y,Z, ABSOLUTE COORDINATES
*             OF ONE OF THE END POINTS OF THE LINE
```

```
*  ENDPT2 =  AN ARRAY (3) CONTAINING THE X,Y,Z, ABSOLUTE COORDINATES
*            OF THE OTHER END POINT OF THE LINE
*  EPT1   =  END POINT VARIABLE USED IN CONSTRUCTING LINES
*  FACE   =  FACE WIDTH OF THE GEAR
*  GROUP  =  A 1-WORD ARRAY CONTAINING THE GROUP ID
*  HOLRAD =  RADIUS OF THE BORE HOLE
*  HUBRAD =  RADIUS OF THE HUB
*  I      =  INTEGER COUNTER
*  ICODE  =  NOT USED AT PRESENT
*  IDUMMY =  A DUMMY VARIABLE FOR LATER USE
*  IDVU   =  4-CHARACTERS, OF WHICH THE LAST TWO ARE USED 'XXID'
*  II1    =  INTEGER VALUE USED TO SAVE CERTAIN POINTS TO DRAW SPLINES
*  II2    =  INTEGER VALUE USED TO SAVE CERTAIN POINTS TO DRAW SPLINES
*  II3    =  INTEGER VALUE USED TO SAVE CERTAIN POINTS TO DRAW SPLINES
*  II4    =  INTEGER VALUE USED TO SAVE CERTAIN POINTS TO DRAW SPLINES
*  INCR1  =  INCREMENT ANGLE USED TO STEP TOOTH GEOMETRY AROUND CIRCLE
*            TO CREATE ENTIRE GEAR GEOMETRY
*  INCR2  =  ANGLE USED TO STEP TOOTH GEOMETRY AROUND CIRCLE
*            TO CREATE ENTIRE GEAR GEOMETRY
*  INCR3  =  ANGLE USED TO STEP TOOTH GEOMETRY AROUND CIRCLE
*            TO CREATE ENTIRE GEAR GEOMETRY
*  IOPT   =  FOR CADFIL:  INTEGER VALUE  = 0  FILE A DRAWING
*                                        = 1  RETRIEVE A DRAWING
*                                        = 2  OVERFILE A DRAWING
*            FOR BEGVU:  OPTION FLAG WHICH MUST BE SET TO 1:  FLAG WILL
*                        EVENTUALLY DETERMINE HANDLING OF INPUT
*                        COORDINATES
*  IXY    =  DO-LOOP COUNTER
*  IZ     =  INTEGER VALUE REPRESENTING Z VALUE
*  J      =  INTEGER COUNTER
*  JJ     =  INTEGER COUNTER
*  K      =  FOR SPLINE:  INTEGER VALUE SPECIFYING END CONDITIONS
*                         = 0  CURVED END SLOPES UNKNOWN
*                         = 1  FIRST SLOPE SUPPLIED
*                         = 2  LAST SLOPE SUPPLIED
*                         = 3  FIRST AND LAST SLOPE SUPPLIED
*            INTEGER DO-LOOP COUNTER AFTER SPLINES COMPLETED
*  L      =  INTEGER DO-LOOP COUNTER
*  LEND   =  INTEGER VALUE SPECIFYING THE NUMBER OF POINTS USED TO FORM
*            THE FIRST TOOTH GEOMETRY
*  M      =  INTEGER VALUE SPECIFYING THE NUMBER OF DIMENSIONS IN SPACE
*            = 1,2, OR 3 COORDINATES INPUT THROUGH THE D ARRAY
*  N      =  THE NUMBER OF TEETH IN THE GEAR
*  NN     =  INTEGER VALUE REPRESENTING THE NUMBER OF INPUT POINTS
*  NOGOOD =  INTEGER TO BE TESTED AFTER FILING.  IF NOT EQUAL TO ONE,
*            THE FILE CALL WAS NO GOOD
*  NO     =  INTEGER USED TO WRITE DATA OR STATEMENTS TO THE SCREEN
*  PI     =  3.1415927
*  PROTRU =  INTEGER NUMBER   0 - NO HUB ON GEAR
*                             1 - HUB ON ONE SIDE OF GEAR
*                             2 - HUB ON BOTH SIDES OF GEAR
```

```
*  RADIUS =  RADIUS USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  RAD2   =  RADIUS USED IN CONSTRUCTION OF ENTIRE GEAR GEOMETRY
*  S      =  A FLOATING POINT ARRAY OF CHORD LENGTHS:  DEFINED BY
*            INPUT DATA
*  T1     =  TEMPORARY FLOATING POINT WORKING AREA MUST BE LENGTH 'NN'
*  T2     =  TEMPORARY FLOATING POINT WORKING AREA MUST BE LENGTH 'NN'
*  T3     =  TEMPORARY FLOATING POINT WORKING AREA MUST BE LENGTH 'NN'
*  USERID =  A 2-WORD ARRAY CONTAINING THE EMPLOYEE ID
*  V      =  A FLOATING POINT ARRAY OF SPLINE TANGENT VECTOR COMPONENTS
*            AT INPUT POINTS
*  X      =  X-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  Y      =  Y-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  Z      =  Z-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*            GENERATION OF THE GEAR TOOTH
*  XSAVE  =  SPECIAL X-COORDINATES SAVED ON THE GEAR TOOTH
*  YSAVE  =  SPECIAL Y-COORDINATES SAVED ON THE GEAR TOOTH
*  XVCTR  =  AN ARRAY, LENGTH 3, REPRESENTING THE A,B,C DIRECTION
*            COSINES OF THE NEW X-AXIS
*  YVCTR  =  AN ARRAY, LENGTH 3, REPRESENTING THE A,B,C DIRECTION
*            COSINES OF THE NEW Y-AXIS
*  ZVCTR  =  AN ARRAY, LENGTH 3, REPRESENTING THE A,B,C DIRECTION
*            COSINES OF THE NEW Z-AXIS
*  XYZ    =  AN ARRAY (3) REPRESENTING THE X,Y,Z LOCATION OF THE VIEW
*  XYZPT  =  AN ARRAY (3) CONTAINING THE X,Y,Z ABSOLUTE COORDINATES
*            OF A POINT
*  X1     =  X-COORDINATE USED TO DRAW THE ENTIRE GEAR GEOMETRY
*  Y1     =  Y-COORDINATE USED TO DRAW THE ENTIRE GEAR GEOMETRY


*     CADCD SUBROUTINES CALLED:  ARC3D, BEGVU, CADFIL, CADST,
*                                C3DCRN, LINE3D, PT3D, SPLINE


*   VARIABLE DECLARATIONS
      CHARACTER DRAWID*20, USERID*8, GROUP*4, IDVU*4, ANS

      INTEGER I, J, JJ, IZ, NN, M, K, LEND, II1, II2, II3, II4,
     CICODE, PROTRU, NO

      REAL XYZ(3),XVCTR(3),YVCTR(3),ZVCTR(3),D(3,200),V(3,200),
     CS(200),T1(200),T2(200),T3(200),Z(4),X(3000),Y(3000),
     CFACE,PI,N,INCR1,INCR2,INCR3, ANGLE(3000), RADIUS(3000),
     CANGL2(3000), CENTER(3), HOLRAD, AMOUNT, HUBRAD,
     CX1(3000),Y1(3000), XSAVE(8), YSAVE(8), ENDPT1(3), ENDPT2(3),
     CEPT1(10), AA, BB, CC, CENTR1(3),
     CARCPT1(3), ARCPT2(3), CIRNRM(3), ANGLE2(3000), RAD2(3000),
     CANGL1(3000), APT1(2), APT2(2), ANGLE1(3000), XYZPT(3),
     CIXY, API, AINCR
```

```
      COMMON /INTPRO/ PROTRU
      COMMON /RELPRO/ AMOUNT, HOLRAD, FACE, HUBRAD

      PARAMETER (PI=3.1415926)
      DATA DRAWID/'DRAW GEARS     '/,USERID/'GEARS '/,GROUP/'MECH'/


      DATA IDVU/'IDIS'/,XVCTR/1.,0.,0./,YVCTR/0.,1.,0./,ZVCTR/0.,0.,1./,
     >XYZ/0.,0.,0./,NO/6/

*
*  START A DRAWING
*
      CALL CADST (DRAWID, USERID, GROUP)
*
*  START A VIEW
*
      IOPT = 1
      CALL BEGVU (IOPT, IDVU, XYZ, XVCTR, YVCTR, ZVCTR, *1000)
*
*  SET UP SOME OF THE VALUES NEEDED FOR SPLINE
*
      M = 3
      K = 0
      Z(1) = 0.
      Z(2) = FACE
      Z(3) = FACE + AMOUNT
      Z(4) = 0. - AMOUNT
      INCR1 = 0.
      INCR2 = 2. * PI / N
*
*  STORE SOME SPECIAL POINTS
*
      DO 100 I = 2,LEND
         IF (X(I) .EQ. XSAVE(2)) THEN
            II1 = I
         ENDIF
         IF (X(I) .EQ. XSAVE(4)) THEN
            II2 = I
         ENDIF
         IF (X(I) .EQ. XSAVE(5)) THEN
            II3 = I
         ENDIF
         IF (X(I) .EQ. XSAVE(7)) THEN
            II4 = I
         ENDIF
100   CONTINUE

      NN = (II2 - II1) + 1
*
```

```
*  DRAW THE INVOLUTE AND TROCHOID CURVES WITH SPLINES
*
      DO 500 IZ = 1,2

         DO 400 JJ = 1,N
            J = 1

            DO 300 I = II1,II2
               ANGLE(I) = ATAN2(Y(I),X(I))
               RADIUS(I) = SQRT(X(I)*X(I) + Y(I)*Y(I))
               ANGL2(I) = ANGLE(I) + INCR1
               X1(I) = RADIUS(I) * COS(ANGL2(I))
               Y1(I) = RADIUS(I) * SIN(ANGL2(I))

               D(1,J) = X1(I)
               D(2,J) = Y1(I)
               D(3,J) = Z(IZ)

               J = J + 1
300         CONTINUE

            CALL SPLINE(NN,M,K,D,V,S,T1,T2,T3,*1001)
            J = 1

            DO 350 I = II3,II4
               ANGLE(I) = ATAN2(Y(I),X(I))
               RADIUS(I) = SQRT(X(I)*X(I) + Y(I)*Y(I))
               ANGL2(I) = ANGLE(I) + INCR1
               X1(I) = RADIUS(I) * COS(ANGL2(I))
               Y1(I) = RADIUS(I) * SIN(ANGL2(I))

               D(1,J) = X1(I)
               D(2,J) = Y1(I)
               D(3,J) = Z(IZ)

               J = J + 1
350         CONTINUE

            CALL SPLINE(NN,M,K,D,V,S,T1,T2,T3,*1001)
            INCR1 = INCR1 + INCR2

400      CONTINUE

500   CONTINUE
*
*  SET UP SOME OF THE VALUES NEEDED FOR ARCS AND CIRCLES
*
      AA = 0
      BB = 0
      CC = 20
      ICODE = 1
```

```
      CIRNRM(1) = AA
      CIRNRM(2) = BB
      CIRNRM(3) = CC
*
*  DRAW THE ADDENDUM CIRCLE OF THE GEAR TEETH USING ARCS
*
      DO 520 IZ = 1,2
         J = 1

         CENTR1(1) = 0
         CENTR1(2) = 0
         CENTR1(3) = Z(IZ)

         ARCPT1(1) = XSAVE(4)
         ARCPT1(2) = YSAVE(4)
         ARCPT1(3) = Z(IZ)

         ARCPT2(1) = XSAVE(5)
         ARCPT2(2) = YSAVE(5)
         ARCPT2(3) = Z(IZ)

         CALL ARC3D(ICODE, CENTR1, ARCPT1, ARCPT2, CIRNRM, *1004)

520   CONTINUE

      DO 540 IZ = 1,2
         INCR1 = INCR2

         DO 530 K = 1,N
            ANGLE1(K) = ATAN2(YSAVE(4),XSAVE(4))
            ANGLE2(K) = ATAN2(YSAVE(5),XSAVE(5))
            RADIUS(K) = SQRT(XSAVE(4)*XSAVE(4) + YSAVE(4)*YSAVE(4))
            RAD2(K) = SQRT(XSAVE(5)*XSAVE(5) + YSAVE(5)*YSAVE(5))
            ANGL1(K) = ANGLE1(K) + INCR1
            ANGL2(K) = ANGLE2(K) + INCR1
            APT1(1) = RADIUS(K) * COS(ANGL1(K))
            APT1(2) = RADIUS(K) * SIN(ANGL1(K))
            APT2(1) = RAD2(K) * COS(ANGL2(K))
            APT2(2) = RAD2(K) * SIN(ANGL2(K))

            CENTR1(1) = 0
            CENTR1(2) = 0
            CENTR1(3) = Z(IZ)

            ARCPT1(1) = APT1(1)
            ARCPT1(2) = APT1(2)
            ARCPT1(3) = Z(IZ)

            ARCPT2(1) = APT2(1)
            ARCPT2(2) = APT2(2)
            ARCPT2(3) = Z(IZ)
```

```
            CALL ARC3D(ICODE, CENTR1, ARCPT1, ARCPT2, CIRNRM, *1004)
            INCR1 = INCR1 + INCR2

530       CONTINUE

540    CONTINUE
*
*  DRAW THE ROOT CIRCLE OF THE GEAR TEETH USING ARCS
*
       DO 680 IZ = 1,2
          INCR1 = INCR2
          INCR3 = 0

          DO 660 K = 1,N

             ANGLE(K) = ATAN2(YSAVE(7),XSAVE(7))
             RADIUS(K) = SQRT(XSAVE(7)*XSAVE(7) + YSAVE(7)*YSAVE(7))
             ANGL1(K) = ANGLE(K) + INCR3
             APT1(1) = RADIUS(K) * COS(ANGL1(K))
             APT1(2) = RADIUS(K) * SIN(ANGL1(K))

             ANGLE2(K) = ATAN2(YSAVE(2),XSAVE(2))
             RAD2(K) = SQRT(XSAVE(2)*XSAVE(2) + YSAVE(2)*YSAVE(2))
             ANGL2(K) = ANGLE2(K) + INCR1
             APT2(1) = RAD2(K) * COS(ANGL2(K))
             APT2(2) = RAD2(K) * SIN(ANGL2(K))

             CENTR1(1) = 0
             CENTR1(2) = 0
             CENTR1(3) = Z(IZ)

             ARCPT1(1) = APT1(1)
             ARCPT1(2) = APT1(2)
             ARCPT1(3) = Z(IZ)

             ARCPT2(1) = APT2(1)
             ARCPT2(2) = APT2(2)
             ARCPT2(3) = Z(IZ)

             CALL ARC3D(ICODE, CENTR1, ARCPT1, ARCPT2, CIRNRM, *1004)
             INCR1 = INCR1 + INCR2
             INCR3 = INCR3 + INCR2
660       CONTINUE

680    CONTINUE
*
*
*  DRAW LINES CONNECTING FRONT FACE AND BACK FACE OF GEAR AT
*  INVOLUTE/ADDENDUM INTERSECTION AND ROOT/TROCHOID INTERSECTION
*
```

```
      INCR1 = 0.

      DO 650 K = 1, N

         DO 550 L = 4,5
            ANGLE(L-3) = ATAN2(YSAVE(L),XSAVE(L))
            RADIUS(L-3) = SQRT(XSAVE(L)*XSAVE(L) + YSAVE(L)*YSAVE(L))
            ANGL2(L-3) = ANGLE(L-3) + INCR1

            IF (L .EQ. 4) THEN
               EPT1(L-3) = RADIUS(L-3) * COS(ANGL2(L-3))
               EPT1(L-2) = RADIUS(L-3) * SIN(ANGL2(L-3))
            ELSEIF (L .EQ. 5) THEN
               EPT1(L-4) = RADIUS(L-3) * COS(ANGL2(L-3))
               EPT1(L-3) = RADIUS(L-3) * SIN(ANGL2(L-3))
            ENDIF

            ENDPT1(1) = EPT1(1)
            ENDPT1(2) = EPT1(2)
            ENDPT1(3) = Z(1)
            ENDPT2(1) = EPT1(1)
            ENDPT2(2) = EPT1(2)
            ENDPT2(3) = Z(2)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

550      CONTINUE

         INCR1 = INCR1 + INCR2

650   CONTINUE

      INCR1 = INCR2

      DO 850 K = 1, N

         DO 750 L = 2,7,5
            ANGLE(L-1) = ATAN2(YSAVE(L),XSAVE(L))
            RADIUS(L-1) = SQRT(XSAVE(L)*XSAVE(L) + YSAVE(L)*YSAVE(L))
            ANGL2(L-1) = ANGLE(L-1) + INCR1

            IF (L .EQ. 2) THEN
               EPT1(L-1) = RADIUS(L-1) * COS(ANGL2(L-1))
               EPT1(L) = RADIUS(L-1) * SIN(ANGL2(L-1))
            ELSEIF (L .EQ. 7) THEN
               EPT1(L-6) = RADIUS(L-1) * COS(ANGL2(L-1))
               EPT1(L-5) = RADIUS(L-1) * SIN(ANGL2(L-1))
            ENDIF

            ENDPT1(1) = EPT1(1)
            ENDPT1(2) = EPT1(2)
```

```
            ENDPT1(3) = Z(1)
            ENDPT2(1) = EPT1(1)
            ENDPT2(2) = EPT1(2)
            ENDPT2(3) = Z(2)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

750       CONTINUE

          INCR1 = INCR1 + INCR2

850    CONTINUE
*
*
*  DO THE FOLLOWING IF THE GEAR HAS NO HUB
*
       API = (2. * PI) - 0.2
       AINCR = PI / 5.

       IF (PROTRU .EQ. 0) THEN
*
*  DRAW BORE HOLE CIRCLES
*
          DO 900 IZ = 1,2

             CENTER(1) = 0
             CENTER(2) = 0
             CENTER(3) = Z(IZ)

             CALL C3DCRN(CENTER, HOLRAD, CIRNRM, *1005)
*
*        DRAW POINTS AROUND BORE CIRCLES
*
             DO 860 IXY = 0,API,AINCR
                XYZPT(1) = HOLRAD * COS(IXY)
                XYZPT(2) = HOLRAD * SIN(IXY)
                XYZPT(3) = Z(IZ)

                CALL PT3D (XYZPT, *1006)

860          CONTINUE

900       CONTINUE
*
*  DRAW LINES CONNECTING FRONT FACE BORE CIRCLE
*        AND BACK FACE BORE CIRCLE
*
          DO 905 IXY = 0,API,AINCR

             ENDPT1(1) = HOLRAD * COS(IXY)
             ENDPT1(2) = HOLRAD * SIN(IXY)
```

```
            ENDPT1(3) = Z(1)
            ENDPT2(1) = HOLRAD * COS(IXY)
            ENDPT2(2) = HOLRAD * SIN(IXY)
            ENDPT2(3) = Z(2)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

905      CONTINUE
*
*
*  DO THE FOLLOWING IF THE GEAR HAS A HUB ON ONLY ONE SIDE
*
      ELSEIF (PROTRU .EQ. 1) THEN
*
*  DRAW BORE CIRCLES AT HUB FACE AND BACK FACE OF GEAR
*
         DO 910 IZ = 1,3,2
            CENTER(1) = 0.
            CENTER(2) = 0.
            CENTER(3) = Z(IZ)

            CALL C3DCRN (CENTER, HOLRAD, CIRNRM, *1005)
*
*       DRAW POINTS AROUND BORE CIRCLES
*
            DO 906 IXY = 0,API,AINCR
               XYZPT(1) = HOLRAD * COS(IXY)
               XYZPT(2) = HOLRAD * SIN(IXY)
               XYZPT(3) = Z(IZ)

               CALL PT3D (XYZPT, *1006)

906         CONTINUE

910      CONTINUE
*
*  DRAW LINES CONNECTING THE 2 BORE CIRCLES
*
         DO 915 IXY = 0,API,AINCR

            ENDPT1(1) = HOLRAD * COS(IXY)
            ENDPT1(2) = HOLRAD * SIN(IXY)
            ENDPT1(3) = Z(1)
            ENDPT2(1) = HOLRAD * COS(IXY)
            ENDPT2(2) = HOLRAD * SIN(IXY)
            ENDPT2(3) = Z(3)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

915      CONTINUE
*
```

```
*  DRAW HUB CIRCLES
*
         DO 920 IZ = 2,3
            CENTER(1) = 0.
            CENTER(2) = 0.
            CENTER(3) = Z(IZ)

            CALL C3DCRN (CENTER, HUBRAD, CIRNRM, *1005)
*
*        DRAW POINTS AROUND HUB CIRCLES
*
            DO 916 IXY = 0,API,AINCR
               XYZPT(1) = HUBRAD * COS(IXY)
               XYZPT(2) = HUBRAD * SIN(IXY)
               XYZPT(3) = Z(IZ)

               CALL PT3D (XYZPT, *1006)

916         CONTINUE

920      CONTINUE
*
*  DRAW LINES CONNECTING HUB CIRCLES
*
         DO 925 IXY = 0,API,AINCR

            ENDPT1(1) = HUBRAD * COS(IXY)
            ENDPT1(2) = HUBRAD * SIN(IXY)
            ENDPT1(3) = Z(2)
            ENDPT2(1) = HUBRAD * COS(IXY)
            ENDPT2(2) = HUBRAD * SIN(IXY)
            ENDPT2(3) = Z(3)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

925      CONTINUE
*
*
*  DO THE FOLLOWING IF HUB IS ON BOTH SIDES OF THE GEAR
*
      ELSEIF (PROTRU .EQ. 2) THEN
*
*  DRAW BORE CIRCLES
*
         DO 930 IZ = 3,4

            CENTER(1) = 0.
            CENTER(2) = 0.
            CENTER(3) = Z(IZ)
```

```
            CALL C3DCRN (CENTER, HOLRAD, CIRNRM, *1005)
*
*       DRAW POINTS AROUND BORE CIRCLE
*
            DO 926 IXY = 0,API,AINCR

               XYZPT(1) = HOLRAD * COS(IXY)
               XYZPT(2) = HOLRAD * SIN(IXY)
               XYZPT(3) = Z(IZ)

               CALL PT3D (XYZPT, *1006)

926         CONTINUE

930      CONTINUE
*
*  DRAW LINES CONNECTING BORE CIRCLES
*
         DO 935 IXY = 0,API,AINCR

            ENDPT1(1) = HOLRAD * COS(IXY)
            ENDPT1(2) = HOLRAD * SIN(IXY)
            ENDPT1(3) = Z(3)
            ENDPT2(1) = HOLRAD * COS(IXY)
            ENDPT2(2) = HOLRAD * SIN(IXY)
            ENDPT2(3) = Z(4)

            CALL LINE3D (ENDPT1, ENDPT2, *1003)

935      CONTINUE
*
*  DRAW HUB CIRCLES
*
         DO 940 IZ = 1,4
            CENTER(1) = 0.
            CENTER(2) = 0.
            CENTER(3) = Z(IZ)

            CALL C3DCRN (CENTER, HUBRAD, CIRNRM, *1005)
*
*       DRAW POINTS AROUND HUB CIRCLES
*
            DO 936 IXY = 0,API,AINCR

               XYZPT(1) = HUBRAD * COS(IXY)
               XYZPT(2) = HUBRAD * SIN(IXY)
               XYZPT(3) = Z(IZ)

               CALL PT3D (XYZPT, *1006)

936         CONTINUE
```

```
940       CONTINUE
*
*  DRAW LINES CONNECTING FRONT HUB CIRCLES
*
          DO 945 IXY = 0,API,AINCR

             ENDPT1(1) = HUBRAD * COS(IXY)
             ENDPT1(2) = HUBRAD * SIN(IXY)
             ENDPT1(3) = Z(1)
             ENDPT2(1) = HUBRAD * COS(IXY)
             ENDPT2(2) = HUBRAD * SIN(IXY)
             ENDPT2(3) = Z(4)

             CALL LINE3D (ENDPT1, ENDPT2, *1003)

945       CONTINUE
*
*  DRAW LINES CONNECTING BACK HUB CIRCLES
*
          DO 946 IXY = 0,API,AINCR

             ENDPT1(1) = HUBRAD * COS(IXY)
             ENDPT1(2) = HUBRAD * SIN(IXY)
             ENDPT1(3) = Z(2)
             ENDPT2(1) = HUBRAD * COS(IXY)
             ENDPT2(2) = HUBRAD * SIN(IXY)
             ENDPT2(3) = Z(3)

             CALL LINE3D (ENDPT1, ENDPT2, *1003)

946       CONTINUE


       ENDIF


*  FILE THE DRAWING

99     IOPT=2
       NOGOOD=0

       CALL CADFIL (IOPT, NOGOOD, IDUMMY)
       IF(NOGOOD.NE.1)THEN
          WRITE(NO,*)'ERROR IN CADFIL'
       ENDIF

       GOTO 2333

1000   WRITE(NO,*)'ERROR EXISTS IN BEGVU'
1001   WRITE(NO,*)'ERROR EXISTS IN SPLINE'
```

```
1002  WRITE(NO,*)'ERROR EXISTS IN CADNL'
1003  WRITE(NO,*)'ERROR EXISTS IN LINE'
1004  WRITE(NO,*)'ERROR EXISTS IN ARC'
1005  WRITE(NO,*)'ERROR EXISTS IN CIRCLE'
1006  WRITE(NO,*)'ERROR EXISTS IN POINT'

2333  RETURN
      END
```

APPENDIX E. MOVIE.BYU SUBROUTINE

```
*****************************************************************************************
*****************************************************************************************
**********                                                                     *********
********      THIS SUBROUTINE DRAWS THE GEAR GEOMETRY IN 3-D                  *******
********                  USING MOVIE.BYU SUBROUTINES                         *******
**********                                                                     *********
*****************************************************************************************
*****************************************************************************************

      SUBROUTINE MOVIE(X, Y, LEND, PI, N, XSAVE, YSAVE, IEND)


*     ARGUMENTS:
*  AMOUNT =  DISTANCE THE HUB PROTRUDES FROM THE GEAR BLANK FACE
*  ANGLE  =  VARIABLE ANGLE USED IN CONSTRUCTING ENTIRE GEAR GEOMETRY
*            FROM TOOTH GEOMETRY
*  ANGLE2 =  VARIABLE ANGLE USED IN CONSTRUCTING ENTIRE GEAR GEOMETRY
*            FROM TOOTH GEOMETRY
*  BFLEND =  INTEGER VALUE - LAST END POINT FOUND ON THE BACK FACE OF
*            THE GEAR
*  COORD  =  AN ARRAY OF THE COORDINATES OF THE NODES
*  FACE   =  FACE WIDTH OF THE GEAR
*  FFLEND =  INTEGER VALUE - LAST END POINT FOUND ON THE FRONT FACE OF
*            THE GEAR
*  HOLRAD =  RADIUS OF THE BORE HOLE
*  HUBRAD =  RADIUS OF THE HUB
*  I      =  INTEGER DO-LOOP COUNTER
*  IEND   =  INTEGER USED TO SAVE THE COORDINATES OF THE INTERSECTION
*            BETWEEN CENTERLINE OF THE TOOTH AND ADDENDUM CIRCLE
*  II     =  INTEGER DO-LOOP COUNTER
*  III    =  INTEGER COUNTER
*  IJ     =  INTEGER DO-LOOP COUNTER
*  INCR1  =  INCREMENT ANGLE USED TO STEP TOOTH GEOMETRY AROUND CIRCLE
*            TO CREATE ENTIRE GEAR GEOMETRY
*  INCR2  =  ANGLE USED TO STEP TOOTH GEOMETRY AROUND CIRCLE
*            TO CREATE ENTIRE GEAR GEOMETRY
*  IP     =  THE CONNECTIVITY OF THE ELEMENTS OR POLYGONS
*  J      =  INTEGER COUNTER
*  JJ     =  INTEGER COUNTER
*  LEND   =  THE LAST END POINT FOR ONE TOOTH, THE NUMBER OF POINTS
*            TO FORM TOOTH GEOMETRY
*  LL     =  INTEGER DO-LOOP COUNTER
*  N      =  NUMBER OF TEETH ON THE GEAR
*  NCON   =  INTEGER VALUE - NUMBER OF ENTRIES IN CONNECTIVITY ARRAY
*  NHLEND =  INTEGER VALUE - THE LAST END POINT OF GEAR GEOMETRY WITH
```

```
*            NO HUB
*  NI      =  INTEGER USED TO READ DATA FROM THE SCREEN
*  NJ      =  INTEGER - THE NUMBER OF NODES OR JOINTS
*  NLEND   =  INTEGER COUNTER OF THE NUMBER OF POINTS IN THE GEAR
*             GEOMETRY
*  NO      =  INTEGER USED TO WRITE DATA OR STATEMENTS TO THE SCREEN
*  NP      =  INTEGER - THE NUMBER OF PARTS
*  NPL     =  INTEGER - THE PARTS LIST
*  NPT     =  INTEGER - THE NUMBER OF ELEMENTS OR POLYGONS
*  NTEST   =  INTEGER - FORMAT TEST VARIABLE (MUST = 0)
*  PA1END  =  INTEGER VALUE - FIRST END POINT AT 'AMOUNT' FROM FRONT
*             FACE AT HUB RADIUS
*  PA2END  =  INTEGER VALUE - FIRST END POINT AT '-AMOUNT' FROM BACK
*             FACE AT HUB RADIUS
*  PB1END  =  INTEGER VALUE - FIRST END POINT ON THE FRONT HUB
*             AT THE BORE RADIUS
*  PB2END  =  INTEGER VALUE - FIRST END POINT ON THE BACK HUB
*             AT THE BORE RADIUS
*  PF1END  =  INTEGER VALUE - FIRST END POINT ON THE FRONT FACE OF THE
*             GEAR AT THE HUB RADIUS
*  PF2END  =  INTEGER VALUE - FIRST END POINT ON THE BACK FACE OF THE
*             GEAR AT THE HUB RADIUS
*  PI      =  3.1415927
*  PROTRU  =  INTEGER NUMBER:   = 0  NO HUB ON GEAR
*                               = 1  HUB ON 1 SIDE OF GEAR ONLY
*                               = 2  HUB ON BOTH SIDES OF GEAR
*  RADIUS  =  RADIUS USED IN CONSTRUCTION OF ENTIRE GEAR FROM GEAR TOOTH
*             GEOMETRY
*  SFEND   =  INTEGER VALUE - FIRST END POINT ON THE BORE CIRCLE OF THE
*             FRONT FACE OF THE GEAR
*  SLEND   =  INTEGER VALUE - LAST END POINT ON THE BORE CIRCLE OF THE
*             FRONT FACE OF THE GEAR
*  X       =  X-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE GEAR TOOTH
*  Y       =  Y-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE GEAR TOOTH
*  Z       =  Z-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE GEAR TOOTH
*  X1      =  X-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE ENTIRE GEAR
*  Y1      =  Y-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE ENTIRE GEAR
*  Z1      =  Z-COORDINATE OF THE SYSTEM USED FOR THE GEOMETRY OF THE
*             GENERATION OF THE ENTIRE GEAR
*  XSAVE   =  SPECIAL X-COORDINATES SAVED ON THE GEAR TOOTH
*  YSAVE   =  SPECIAL Y-COORDINATES SAVED ON THE GEAR TOOTH


*   VARIABLE DECLARATIONS
      INTEGER I, II, J, LEND, NLEND, SFEND, SLEND, BFLEND ,FFLEND,
     CNO, NI, IP(9999), NP, NJ, NPT, NPL(2,3), NTEST, NCON,
```

```
      CPROTRU, PB1END, PF1END, PA1END, PB2END, PF2END, PA2END,
      CNHLEND, III, JJ, IEND, SVLEND

       REAL X(3000), Y(3000), Z(3000), N, PI, INCR1, INCR2, ANGLE(500),
      CRADIUS(500), ANGLE2(500), FACE, AMOUNT, HOLRAD, HUBRAD,
      CX1(6000), Y1(6000), Z1(6000), COORD(3, 6000), XSAVE(8), YSAVE(8)


       COMMON /INTPRO/ PROTRU
       COMMON /RELPRO/ AMOUNT, HOLRAD, FACE, HUBRAD

       DATA NO/6/, NI/5/
*
*  SAVE THE VALUE OF 'LEND' THAT WAS PASSED THROUGH THE SUBROUTINE
*
       SVLEND = LEND
*
*  TRANSFER THE FIRST TOOTH GEOMETRY X,Y,Z TO X1,Y1,Z1
*  ELIMINATE THE EXTRA POINTS THAT ARE NOT AS CRITICAL FOR THE DESIGN
*  AT THE ROOT AND ADDENDUM CIRCLES OF THE GEAR PROFILE
*
       I = 2
       X1(1) = X(2)
       Y1(1) = Y(2)
       Z1(1) = 0.

       DO 100 I = 3, LEND
          IF (XSAVE(2) .EQ. X(I) .AND. YSAVE(2) .EQ. Y(I)) THEN
             GOTO 105
          ENDIF
100    CONTINUE

105    X1(2) = X(I)
       Y1(2) = Y(I)
       Z1(2) = 0.

       J = 0
       II = I
       DO 110 I = II + 1, LEND
          J = J + 1
          X1(2+J) = X(I)
          Y1(2+J) = Y(I)
          Z1(2+J) = 0.
          IF (XSAVE(4) .EQ. X(I) .AND. YSAVE(4) .EQ. Y(I)) THEN
             GOTO 115
          ENDIF
110    CONTINUE

115    JJ = J + 3
       X1(JJ) = X(IEND)
       Y1(JJ) = Y(IEND)
```

```
      Z1(JJ) = 0.

      JJ = JJ + 1
      X1(JJ) = XSAVE(5)
      Y1(JJ) = YSAVE(5)
      Z1(JJ) = 0.

      DO 120 I = 2, LEND
         IF (XSAVE(5) .EQ. X(I) .AND. YSAVE(5) .EQ. Y(I)) THEN
            II = I
            GOTO 125
         ENDIF
120   CONTINUE

125   DO 130 I = II + 1, LEND
         JJ = JJ + 1
         X1(JJ) = X(I)
         Y1(JJ) = Y(I)
         Z1(JJ) = 0.
         IF (XSAVE(7) .EQ. X(I) .AND. YSAVE(7) .EQ. Y(I)) THEN
            GOTO 135
         ENDIF
130   CONTINUE
*
*  REDEFINE 'LEND' TO THE NEW NUMBER OF POINTS FOR THE GEAR TOOTH
*  GEOMETRY.  THE LAST NUMBER IS NOW THE TROCHOID/ROOT CIRCLE
*  INTERSECTION
*
135   LEND = JJ

*
*  CALCULATE THE POINTS FOR THE REST OF THE GEAR GEOMETRY (FRONT FACE)
*
      INCR1 = 2. * PI / N
      INCR2 = INCR1
      NLEND = LEND + 1

      DO 300 II = 2, N

         DO 200 I = 1, LEND

            ANGLE(I) = ATAN2(Y1(I),X1(I))
            RADIUS(I) = SQRT(X1(I)*X1(I) + Y1(I)*Y1(I))
            ANGLE2(I) = ANGLE(I) + INCR1

            X1(NLEND) = RADIUS(I) * COS(ANGLE2(I))
            Y1(NLEND) = RADIUS(I) * SIN(ANGLE2(I))
            Z1(NLEND) = 0.

            NLEND = NLEND + 1
```

```
200       CONTINUE

          INCR1 = INCR1 + INCR2

300    CONTINUE

       FFLEND = NLEND - 1

*
*  CALCULATE THE POINTS OF THE BACK FACE GEAR COORDINATES
*
       DO 400 I = 1, FFLEND

          X1(NLEND) = X1(I)
          Y1(NLEND) = Y1(I)
          Z1(NLEND) = -(FACE)

          NLEND = NLEND + 1

400    CONTINUE

       BFLEND = NLEND - 1
*
*  CALCULATE THE BORE HOLE CIRCLE COORDINATES - FRONT FACE
*
       X1(NLEND) = HOLRAD * SIN(PI/N)
       Y1(NLEND) = HOLRAD * COS(PI/N)
       Z1(NLEND) = 0.

       SFEND = NLEND
       NLEND = NLEND + 1
       INCR1 = 2. * PI / N


       DO 700 I = 2, N

          ANGLE(I) = ATAN2 (Y1(SFEND),X1(SFEND))
          ANGLE2(I) = ANGLE(I) + INCR1

          X1(NLEND) = HOLRAD * COS(ANGLE2(I))
          Y1(NLEND) = HOLRAD * SIN(ANGLE2(I))
          Z1(NLEND) = 0.

          NLEND = NLEND + 1
          INCR1 = INCR1 + INCR2

700    CONTINUE

       SLEND = NLEND - 1
*
```

```
*  CALCULATE THE BORE HOLE CIRCLE COORDINATES - BACK FACE
*
      DO 800 I = 1, N

         X1(NLEND) = X1(BFLEND + I)
         Y1(NLEND) = Y1(BFLEND + I)
         Z1(NLEND) = -(FACE)

         NLEND = NLEND + 1

800   CONTINUE

      NHLEND = NLEND - 1
*
*
*  DO THE FOLLOWING IF A HUB EXISTS
*
      IF (PROTRU .EQ. 1 .OR. PROTRU .EQ. 2) THEN
*
*  CALCULATE THE BORE CIRCLE COORDINATES AT THE DISTANCE 'AMOUNT' FROM
*  FRONT FACE
*
         PB1END = NLEND

         DO 820 I = 1,N
            X1(NLEND) = X1(BFLEND + I)
            Y1(NLEND) = Y1(BFLEND + I)
            Z1(NLEND) = AMOUNT

            NLEND = NLEND + 1
820      CONTINUE
*
*  CALCULATE THE FRONT HUB CIRCLE COORDINATES ON THE FRONT GEAR FACE
*
         PF1END = NLEND

         X1(NLEND) = HUBRAD * SIN(PI/N)
         Y1(NLEND) = HUBRAD * COS(PI/N)
         Z1(NLEND) = 0.

         NLEND = NLEND + 1
         INCR1 = 2. * PI / N

         DO 830 I = 2, N
            ANGLE(I) = ATAN2(Y1(PF1END),X1(PF1END))
            ANGLE2(I) = ANGLE(I) + INCR1

            X1(NLEND) = HUBRAD * COS(ANGLE2(I))
            Y1(NLEND) = HUBRAD * SIN(ANGLE2(I))
            Z1(NLEND) = 0.
```

```
            NLEND = NLEND + 1
            INCR1 = INCR1 + INCR2
830       CONTINUE
*
*  CALCULATE THE FRONT HUB CIRCLE COORDINATES AT DISTANCE 'AMOUNT' FROM
*  FRONT GEAR FACE
*
          PA1END = NLEND

          DO 840 I = 0, N-1
             X1(NLEND) = X1(PF1END + I)
             Y1(NLEND) = Y1(PF1END + I)
             Z1(NLEND) = AMOUNT

             NLEND = NLEND + 1
840       CONTINUE
*
*
*  DO THE FOLLOWING IF A BACK HUB EXISTS
*
          IF (PROTRU .EQ. 2) THEN
*
*  CALCULATE THE BORE CIRCLE COORDINATES AT '-AMOUNT' FROM BACK FACE
*
             PB2END = NLEND

             DO 850 I = 1, N
                X1(NLEND) = X1(BFLEND + I)
                Y1(NLEND) = Y1(BFLEND + I)
                Z1(NLEND) = -(FACE + AMOUNT)

                NLEND = NLEND + 1
850          CONTINUE
*
*  CALCULATE THE BACK HUB CIRCLE COORDINATES ON THE BACK GEAR FACE
*
             PF2END = NLEND

             DO 860 I = 0, N-1
                X1(NLEND) = X1(PF1END + I)
                Y1(NLEND) = Y1(PF1END + I)
                Z1(NLEND) = -(FACE)

                NLEND = NLEND + 1
860          CONTINUE
*
*  CALCULATE THE BACK HUB CIRCLE COORDINATES AT '-AMOUNT' FROM THE
*  BACK GEAR FACE
*
             PA2END = NLEND
```

```
            DO 870 I = 0, N-1
               X1(NLEND) = X1(PF1END + I)
               Y1(NLEND) = Y1(PF1END + I)
               Z1(NLEND) = -(FACE + AMOUNT)

               NLEND = NLEND + 1
870         CONTINUE

         ENDIF

      ENDIF

      NLEND = NLEND - 1
*
*  SET UP THE COORDINATE ARRAY IN THE CORRECT FORM FOR USE BY MOVIE
*
      DO 900 I = 1, NLEND

         COORD(1,I) = X1(I)
         COORD(2,I) = Y1(I)
         COORD(3,I) = Z1(I)

900   CONTINUE

*
*  SET UP THE CONNECTIVITY ARRAY
*
*
*  SET UP THE ELEMENTS CREATING THE FRONT FACE OF THE GEAR
*
      J = 1

      DO 1060 I = 1, N-1

         DO 1050 LL = LEND, 0, -1

            IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = J
            J = J + 1

1050     CONTINUE

         IP((LEND * I) + (2 * I) + (I - 1)) = SFEND + I
         IP((LEND * I) + (2 * I) + (I - 1) + 1) = -(SFEND + (I- 1))
         J = J - 1

1060  CONTINUE


      I = N

      DO 1070 LL = LEND, 1, -1
```

```
         IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = J
         J = J + 1

1070  CONTINUE

      IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = 1
      IP((LEND * I) + (2 * I) + (I - 1)) = SFEND
      IP((LEND * I) + (2 * I) + (I - 1) + 1) = -(SFEND + (I- 1))

*
*  SET UP THE ELEMENTS CREATING THE BACK FACE OF THE GEAR
*
      J = FFLEND + 1

      DO 1090 I = N+1, (2*N)-1

         DO 1080 LL = LEND, 0, -1

            IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = J
            J = J + 1

1080     CONTINUE

         IP((LEND * I) + (2 * I) + (I - 1)) = (SLEND + 1)+ (I - N)
         IP((LEND*I) + (2*I) + (I-1)+ 1) = -((SLEND+1) + (I-N-1))
         J = J - 1

1090  CONTINUE


      I = 2 * N

      DO 1100 LL = LEND, 1, -1
         IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = J
         J = J + 1
1100  CONTINUE

      IP((((LEND + 1) * I) - LL) + (2 * (I - 1))) = FFLEND + 1
      IP((LEND * I) + (2 * I) + (I - 1)) = SLEND+ 1
      IP((LEND * I) + (2 * I) + (I - 1) + 1) = -(SLEND + 1 + (I-N-1))

*
*  SET UP THE ELEMENTS FORMING THE TOOTH DEPTH OF THE GEAR
*
      J = 1

      DO 1110 I = 1, (FFLEND - 1)
         IP(((LEND + 3) * 2 * N) + (4 * I) - 3) = J
         IP(((LEND + 3) * 2 * N) + (4 * I) - 2) = J + FFLEND
         IP(((LEND + 3) * 2 * N) + (4 * I) - 1) = J + FFLEND + 1
```

```
         IP(((LEND + 3) * 2 * N) + (4 * I)) = -(J + 1)
         J = J + 1
1110  CONTINUE


      I = FFLEND

      IP(((LEND + 3) * 2 * N) + (4 * I) - 3) = FFLEND
      IP(((LEND + 3) * 2 * N) + (4 * I) - 2) = BFLEND
      IP(((LEND + 3) * 2 * N) + (4 * I) - 1) = FFLEND + 1
      IP(((LEND + 3) * 2 * N) + (4 * I)) = -(1)

*
*  SET UP THE ELEMENTS FORMING THE BORE HOLE OF THE GEAR
*
      DO 1120 I = 1, N - 1
         IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 3)) =
     C   SFEND + (I - 1)
         IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 2)) =
     C   SLEND + 1 + (I - 1)
         IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 1)) =
     C   SLEND + 2 + (I - 1)
         IP(((LEND+3) * 2*N) + (4* FFLEND) + (4*I)) =
     C   -(SFEND + 1 + (I - 1))
1120  CONTINUE

      I = N
      IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 3)) =
     CSLEND
      IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 2)) =
     CNHLEND
      IP(((LEND+3) * 2*N) + (4* FFLEND) + ((4*I) - 1)) =
     CSLEND + 1
      IP(((LEND+3) * 2*N) + (4* FFLEND) + (4*I)) =
     C-(SFEND)

*
*  DO THE FOLLOWING IF A HUB EXISTS
*
      IF (PROTRU .EQ. 1 .OR. PROTRU .EQ. 2) THEN
*
*  SET UP ELEMENTS FORMING THE BACK FACE OF THE FRONT HUB
*  AT FRONT GEAR FACE
*
         DO 1130 I = 1, N - 1
            IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 3)) =
     C      SFEND + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 2)) =
     C      PF1END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 1)) =
     C      PF1END + I
```

```
            IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + (4*I)) =
     C      -(SFEND + I)
1130     CONTINUE

         I = N
         IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 3)) =
     C   SFEND + (I-1)
         IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 2)) =
     C   PF1END + (I-1)
         IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + ((4*I) - 1)) =
     C   PF1END
         IP(((LEND+3)*2*N) + (4*FFLEND) + (4*N) + (4*I)) =
     C   -(SFEND)

*
*  SET UP ELEMENTS FORMING FRONT FACE OF FRONT HUB AT 'AMOUNT' FROM
*  FRONT FACE
*
         DO 1140 I = 1, N - 1
            IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 3)) =
     C      PB1END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 2)) =
     C      PA1END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 1)) =
     C      PA1END + I
            IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + (4*I)) =
     C      -(PB1END + I)
1140     CONTINUE

         I = N
         IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 3)) =
     C   PB1END + (I-1)
         IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 2)) =
     C   PA1END + (I-1)
         IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + ((4*I) - 1)) =
     C   PA1END
         IP(((LEND+3)*2*N) + (4*FFLEND) + 2*(4*N) + (4*I)) =
     C   -(PB1END)

*
*  SET UP ELEMENTS FORMING THE FRONT HUB AT HUB RADIUS
*
         DO 1150 I = 1, N - 1
            IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 3)) =
     C      PA1END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 2)) =
     C      PF1END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 1)) =
     C      PF1END + I
            IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + (4*I)) =
```

```
     C       -(PA1END + I)
1150      CONTINUE

          I = N
          IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 3)) =
     C    PA1END + (I-1)
          IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 2)) =
     C    PF1END + (I-1)
          IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + ((4*I) - 1)) =
     C    PF1END
          IP(((LEND+3)*2*N) + (4*FFLEND) + 3*(4*N) + (4*I)) =
     C    -(PA1END)

*
*  SET UP ELEMENTS FORMING THE BORE HOLE OF THE FRONT HUB
*
          DO 1160 I = 1, N - 1
             IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 3)) =
     C       PB1END + (I-1)
             IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 2)) =
     C       SFEND + (I-1)
             IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 1)) =
     C       SFEND + I
             IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + (4*I)) =
     C       -(PB1END + I)
1160      CONTINUE

          I = N
          IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 3)) =
     C    PB1END + (I-1)
          IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 2)) =
     C    SFEND + (I-1)
          IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + ((4*I) - 1)) =
     C    SFEND
          IP(((LEND+3)*2*N) + (4*FFLEND) + 4*(4*N) + (4*I)) =
     C    -(PB1END)

*
*
*  DO THE FOLLOWING IF A BACK HUB EXISTS
*
          IF (PROTRU .EQ. 2) THEN
*
*  SET UP THE ELEMENTS FORMING THE FACE OF THE BACK HUB AT THE
*  BACK GEAR FACE
*
            DO 1170 I = 1, N - 1
               IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 3)) =
     C         SLEND + 1 + (I-1)
               IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 2)) =
     C         PF2END + (I-1)
```

```
              IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 1)) =
     C        PF2END + I
              IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + (4*I)) =
     C        -((SLEND + 1) + I)
1170        CONTINUE

            I = N
            IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 3)) =
     C      SLEND + 1 + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 2)) =
     C      PF2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + ((4*I) - 1)) =
     C      PF2END
            IP(((LEND+3)*2*N) + (4*FFLEND) + 5*(4*N) + (4*I)) =
     C      -(SLEND + 1)

*
*  SET UP THE ELEMENTS FORMING THE FACE OF THE BACK HUB AT '-AMOUNT'
*  FROM THE BACK GEAR FACE
*
            DO 1180 I = 1, N - 1
              IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 3)) =
     C        PB2END + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 2)) =
     C        PA2END + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 1)) =
     C        PA2END + I
              IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + (4*I)) =
     C        -(PB2END + I)
1180        CONTINUE

            I = N
            IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 3)) =
     C      PB2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 2)) =
     C      PA2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + ((4*I) - 1)) =
     C      PA2END
            IP(((LEND+3)*2*N) + (4*FFLEND) + 6*(4*N) + (4*I)) =
     C      -(PB2END)

*
*  SET UP THE ELEMENTS FORMING THE BACK HUB DEPTH AT THE HUB RADIUS
*
            DO 1190 I = 1, N - 1
              IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 3)) =
     C        PF2END + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 2)) =
     C        PA2END + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 1)) =
     C        PA2END + I
```

```
              IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + (4*I)) =
     C        -(PF2END + I)
1190        CONTINUE

            I = N
            IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 3)) =
     C      PF2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 2)) =
     C      PA2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + ((4*I) - 1)) =
     C      PA2END
            IP(((LEND+3)*2*N) + (4*FFLEND) + 7*(4*N) + (4*I)) =
     C      -(PF2END)

*
*  SET UP THE ELEMENTS FORMING THE BORE HOLE OF THE BACK HUB
*
            DO 1200 I = 1, N - 1
              IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 3)) =
     C        SLEND + 1 + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 2)) =
     C        PB2END + (I-1)
              IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 1)) =
     C        PB2END + I
              IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + (4*I)) =
     C        -((SLEND + 1) + I)
1200        CONTINUE

            I = N
            IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 3)) =
     C      SLEND + 1 + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 2)) =
     C      PB2END + (I-1)
            IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + ((4*I) - 1)) =
     C      PB2END
            IP(((LEND+3)*2*N) + (4*FFLEND) + 8*(4*N) + (4*I)) =
     C      -(SLEND + 1)

          ENDIF

        ENDIF
*
*
*  THIS WILL SET UP THE DATA CORRECTLY FOR USE BY MOVIE.BYU
*       PROTRU CORRESPONDS TO THE TYPE OF BLANK CHOSEN
*       = 0 NO HUB,   = 1 HUB ON ONE SIDE,   = 2 HUB BOTH SIDES
*
*
        IF (PROTRU .EQ. 0) THEN
          NP = 1
          NJ = LEND * N * 2 + (2 * N)
```

```
         NPT = (2 * N) + FFLEND + N
         NPL(1,1) = (2 * N) + 1
         NPL(2,1) = (2 * N) + FFLEND + N
         NTEST = 0
         NCON = ((LEND + 3) * N * 2) + (FFLEND * 4) + (N * 4)

      ELSEIF (PROTRU .EQ. 1) THEN
         NP = 2
         NJ = LEND * N * 2 + (2 * N) + (3 * N)
         NPT = (2 * N) + FFLEND + N + (4 * N)
         NPL(1,1) = (2 * N) + 1
         NPL(2,1) = (2 * N) + FFLEND + N
         NPL(1,2) = ((2 * N) + FFLEND + N) + 1
         NPL(2,2) = ((2 * N) + FFLEND + N) + (4 * N)
         NTEST = 0
         NCON =((LEND+3)*N*2) + (FFLEND*4) + (N*4) + ((N*4)*4)

      ELSEIF (PROTRU .EQ. 2) THEN
         NP = 3
         NJ = LEND * N * 2 + (2 * N) + (3 * N) + (3 * N)
         NPT = (2 * N) + FFLEND + N + (4 * N) + (4 * N)
         NPL(1,1) = (2 * N) + 1
         NPL(2,1) = (2 * N) + FFLEND + N
         NPL(1,2) = ((2 * N) + FFLEND + N) + 1
         NPL(2,2) = ((2 * N) + FFLEND + N) + (4 * N)
         NPL(1,3) = ((2 * N) + FFLEND + N) + (4 * N) + 1
         NPL(2,3) = ((2 * N) + FFLEND + N) + (4 * N) + (4 * N)
         NTEST = 0
         NCON =((LEND+3)*N*2) + (FFLEND*4) + (N*4) + ((N*4)*4) +
     C          ((N*4)*4)

      ENDIF

*
*  WRITE OUT THE DATA TO A FILE IN THE CORRECT FORM FOR USE BY MOVIE
*
      WRITE(8,1500) NP,NJ,NPT,NCON,NTEST
      WRITE(8,1500) ((NPL(I,IJ),I=1,2),IJ=1,NP)
      WRITE(8,1600) ((COORD(I,IJ),I=1,3),IJ=1,NJ)
      WRITE(8,1500) (IP(I),I=1,NCON)
1500  FORMAT(16I5)
1600  FORMAT(6E12.5)
*
*  RETURN 'LEND' AS THE ORIGINAL NUMBER PASSED THROUGH THE SUBROUTINE
*
      LEND = SVLEND

      RETURN
      END
```

APPENDIX F. DATA SUBROUTINE

```
C*************************************************************************
C*************************************************************************
C**********                                                     **********
C**********        DATA OUTPUT TO SCREEN - NO GRAPHICS          **********
C**********                                                     **********
C*************************************************************************
C*************************************************************************

      SUBROUTINE DATA (ANS4)

C*  THIS SUBROUTINE WRITES OUT THE DATA TO THE SCREEN OF THE GEAR
C*  JUST DESIGNED

C*     ARGUMENTS:
C*  A       =  ADDENDUM FRACTION
C*  ADDEND  =  ACTUAL ADDENDUM (FUNCTION OF P)
C*  AMOUNT  =  DISTANCE THE HUB PROTRUDES FROM THE GEAR BLANK FACE
C*  ANS4    =  1-CHARACTER REPLY ASKING FOR A 'Y' OR 'N' TO A QUESTION
C*  B       =  DEDENDUM FRACTION
C*  BASERD  =  BASE RADIUS OF GEAR
C*  CLEAR   =  CLEARANCE
C*  DEDEND  =  ACTUAL DEDENDUM (FUNCTION OF P)
C*  FACE    =  FACE WIDTH OF THE GEAR
C*  HBTPRD  =  ACTUAL HOB TIP RADIUS (FUNCTION OF P)
C*  HK      =  WORKING DEPTH OF GEAR TOOTH
C*  HOLRAD  =  RADIUS OF THE BORE HOLE
C*  HT      =  WHOLE DEPTH OF GEAR TOOTH
C*  HUBRAD  =  RADIUS OF THE HUB
C*  N       =  NUMBER OF TEETH IN THE GEAR
C*  NI      =  INTEGER USED TO READ DATA FROM THE SCREEN
C*  NO      =  INTEGER USED TO WRITE DATA OR STATEMENTS TO THE SCREEN
C*  OUTRAD  =  OUTSIDE RADIUS OF GEAR
C*  P       =  DIAMETRAL PITCH
C*  PCIRC   =  CIRCULAR PITCH
C*  PHI     =  THE CUTTING PRESSURE ANGLE IN RADIANS
C*  PHIDEG  =  THE CUTTING PRESSURE ANGLE IN DEGREES
C*  PROTRU  =  INTEGER NUMBER   0 - NO HUB ON GEAR
C*                              1 - HUB ON ONE SIDE OF GEAR
C*                              2 - HUB ON BOTH SIDES OF GEAR
C*  R       =  THE CUTTING PITCH RADIUS OF THE GEAR BEING DESIGNED
C*             (STANDARD OR NON-STANDARD)
C*  RHF     =  HOB-TOOTH TIP RADIUS (0.0 REPRESENTS A SHARP-TIPPED HOB)
C*  ROOTRD  =  ROOT RADIUS OF GEAR
C*  SYSCAL  =  SUBROUTINE AT VIRGINIA TECH CALLED TO CLEAR THE SCREEN
C*  THKNSS  =  THE TOOTH THICKNESS OF THE GEAR BEING DESIGNED
```

```
*   VARIABLE DECLARATIONS
      INTEGER NO, NI, PROTRU

      REAL ADDEND, DEDEND, HBTPRD, PHIDEG, CLEAR, HK, HT, OUTRAD,
     C     BASERD, ROOTRD, PCIRC, N, P, A, B, RHF, THKNSS, PHI, R,
     C     AMOUNT, HOLRAD, FACE, HUBRAD

      CHARACTER*1 ANS4
      CHARACTER*5 CLE


      COMMON /TODATA/ ADDEND, DEDEND, HBTPRD, PHIDEG, CLEAR, HK, HT,
     C                OUTRAD, BASERD, ROOTRD, PCIRC
      COMMON /TOOTH/ N, P, A, B, RHF, THKNSS, PHI, R
      COMMON /INTPRO/ PROTRU
      COMMON /RELPRO/ AMOUNT, HOLRAD, FACE, HUBRAD


      DATA CLE /'CLEAR'/
      DATA NO/6/, NI/5/


      WRITE(12,*)
      WRITE(12,*)
      WRITE(12,*)'     THIS IS THE DATA FOR THE GEAR YOU HAVE CHOSEN:'
      WRITE(12,*)
      WRITE(12,200) P, CLEAR, ADDEND, HK, DEDEND, HT, HBTPRD, OUTRAD,
     CN, BASERD, R, ROOTRD, PHIDEG, PCIRC, THKNSS, FACE

200   FORMAT(/,
     C    ' PITCH             = ',F12.7,'   CLEARENCE      = ',F12.7,/,
     C    ' ACTUAL ADDENDUM   = ',F12.7,'   WORKING DEPTH  = ',F12.7,/,
     C    ' ACTUAL DEDENDUM   = ',F12.7,'   WHOLE DEPTH    = ',F12.7,/,
     C    ' HOB TIP RADIUS    = ',F12.7,'   OUTSIDE RADIUS = ',F12.7,/,
     C    ' NUMBER OF TEETH   = ',F12.7,'   BASE RADIUS    = ',F12.7,/,
     C    ' CUT. PITCH RADIUS = ',F12.7,'   ROOT RADIUS    = ',F12.7,/,
     C    ' CUT. PRESS ANGLE  = ',F12.7,'   CIRCULAR PITCH = ',F12.7,/,
     C    ' TOOTH THICKNESS   = ',F12.7,'   FACE WIDTH     = ',F12.7)

      IF (ANS4 .EQ. 'N') THEN
         WRITE(12,*)
         WRITE(12,*)'  THE GEAR DOES NOT HAVE A HUB'
         WRITE(12,210) HOLRAD
210      FORMAT('  AND THE BORE RADIUS = ',F12.7)
      ELSEIF (ANS4 .EQ. 'Y') THEN
         WRITE(12,*)
         WRITE(12,220) PROTRU
220      FORMAT('  THE GEAR HAS A HUB ON ',I2,' SIDE(S)')
```

```
          WRITE(12,230) HOLRAD, HUBRAD, AMOUNT
230       FORMAT('  WITH A BORE RADIUS    = ',F12.7,/,
     C           '       A HUB RADIUS     = ',F12.7,/,
     C           '   AND A HUB PROTRUSION = ',F12.7)
          ENDIF

          RETURN
          END
```

APPENDIX G. EXAMPLE OF THE I/O FOR THE PROGRAM

The following is an example of the prompts given the user by the gear design program, and the responses which create the model of the gear shown in Figure 30.

```
     SELECT DISPLAY SYSTEM

     1.  GKS
     2.  CADAM
     3.  MOVIE.BYU
     4.  NO DISPLAY (DATA)

   Type QUIT to exit from program.
?
2


   Hit RETURN to load for CADAM.
   Execution will begin 45 seconds
   after hitting RETURN key.
   If selection is to be changed,
   type INVALID to return to menu.

   Type QUIT to exit from the program.



EXECUTION BEGINS. . .
CADSEG SEGMENT BOUNDARIES-00CA0000 00CA1FFF.
CADAM SEGMENT NOW LOADED.
@EXECUTION BEGINS. . .
```

```
> GEAR


WHAT TYPE OF GEAR WOULD YOU LIKE TO DESIGN?

            1)  SPUR
            2)  HELICAL
            3)  BEVEL
            4)  WORM

             CHOOSE 1,2,3 OR 4  (5 TO QUIT)
?
1


>> SPUR


WHAT TYPE OF CUTTER WOULD YOU LIKE TO USE TO CUT
              THE GEAR TEETH?

             1)  HOB
             2)  PINION CUTTER

               CHOOSE 1 OR 2  (3 TO QUIT)
?
1
```

```
>>> HOB


FOR STANDARD GEARS, 20 AND 25 DEGREE PRESSURE
       ANGLE, FULL-DEPTH, COARSE PITCH:
THE ADDENDUM FRACTION = 1.000  (0.800 FOR STUB)
THE DEDENDUM FRACTION = 1.250  (1.000 FOR STUB)

FOR STANDARD GEARS, 14.5 DEGREE PRESSURE ANGLE,
         FULL-DEPTH, COARSE PITCH:
       THE ADDENDUM FRACTION = 1.000
       THE DEDENDUM FRACTION = 1.157

THE HOB TIP RADIUS OF 0.0 MEANS A SHARP-TIPPED HOB


ENTER THE FOLLOWING DATA:

     DIAMETRAL PITCH, ADDENDUM FRACTION,
     DEDENDUM FRACTION, HOB TIP RADIUS,
     AND NUMBER OF TEETH
?
1  1.00  1.25  0.0  10



IS THIS GOING TO BE A STANDARD GEAR (Y OR N)?
?
y


ENTER THE CUTTING PRESSURE ANGLE (IN DEGREES)
?
20


 ***** THE GEAR IS BEING UNDERCUT!

   TO AVOID UNDERCUTTING THE GEAR TEETH
   INCREASE THE NUMBER OF TEETH TO   22

 WOULD YOU LIKE TO CHANGE THE NUMBER OF TEETH
 IN THE GEAR (Y OR N)?
?
n
```

```
THE INPUT DATA IS:

 PITCH                =      1.0000000
 ADDENDUM FRACTION    =      1.0000000
 DEDENDUM FRACTION    =      1.2500000
 HOB TIP RADIUS       =      0.0000000
 NUMBER OF TEETH      =     10.0000000
 CUTTING PITCH RADIUS =      5.0000000
 CUTTING PRESS ANGLE  =     20.0000000
 TOOTH THICKNESS      =      1.5707960

IS THE INPUT DATA CORRECT (Y OR N)?
?
y



ENTER THE NUMBER OF INTERVALS PER CURVE
     (4 MIN - 80 MAX FOR GKS)
     (4 MIN - 25 MAX FOR CADAM)
     (4 MIN - 15 MAX FOR MOVIE.BYU)
?
15


INTERSECTION POINT IS LOCATED AT:
 X =    0.80798441      Y =    4.68753719
   AT A RADIUS OF     4.75666142
   AND LOCATED IN      8 ITERATIONS



THERE ARE THREE BASIC TYPES OF GEAR BLANK DESIGNS.
WHICH TYPE OF GEAR BLANK WOULD YOU LIKE TO DESIGN?

                1)  PLAIN
                2)  WEBBED
                3)  SPOKED

                   CHOOSE 1,2 OR 3
?
1
```

```
>>> PLAIN


ENTER THE FACE WIDTH OF THE GEAR.
?
1.00


DOES THE GEAR HAVE A HUB PROTRUSION (Y OR N)?
?
y


ON BOTH SIDES OF THE GEAR OR ONLY ONE SIDE?
     CHOOSE "1" FOR ONLY ONE SIDE OR
            "2" FOR BOTH SIDES
?
2


HOW FAR DOES THE PROTRUSION EXTEND FROM THE
       FLUSH FACE OF THE GEAR?
?
0.75


ENTER THE BORE DIAMETER OF THE HOLE IN THE BLANK.
?
1.50


MESSAGE ==><CADCD >REL 20.0---> FILE I/O REQUESTED
MESSAGE ==>.CADFIL.DRAW GEARS            IS OVERFILD
```

APPENDIX H. USEFUL AGMA STANDARDS

Some AGMA Standards that may be found useful as reference material:

1. AGMA Information Sheet for "Strength of Spur, Helical, Herringbone, and Bevel Gear Teeth," AGMA 225.01 (1967), AGMA, Washington, D.C.

2. AGMA Information Sheet for "Surace Durability (Pitting) of Spur, Helical, Herringbone, and Bevel Gear Teeth," AGMA 210.01 (1965), AGMA, Washington, D.C.

3. AGMA Information Sheet for "Gear Nomenclature - Terms, Definitions, Symbols, and Abbreviations," AGMA 112.04 (1965), AGMA, Washington, D.C.

4. AGMA Information Sheet for "Gear Tooth Wear and Failure," AGMA 110.03 (1962), AGMA, Washington, D.C.

The vita has been removed from the scanned document