Study of the Kinematic and Dynamic Characteristics of a Wormgear Transmission for Helicopter Applications

D.C. Sun and Qin Yuan
State University of New York at Binghamton
Binghamton, New York

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited
# TABLE OF CONTENTS

Abstract  
Nomenclature  

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Background and Objectives of the Study</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Wormgear Geometry</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Selection of Wormgear Type</td>
<td>2</td>
</tr>
<tr>
<td>2.2</td>
<td>Layout of Wormgear Dimensions</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Contact Analysis</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Method of Enveloping</td>
<td>5</td>
</tr>
<tr>
<td>3.2</td>
<td>Coordinate Systems and Operators</td>
<td>6</td>
</tr>
<tr>
<td>3.3</td>
<td>Worm Surface</td>
<td>9</td>
</tr>
<tr>
<td>3.4</td>
<td>Gear Surface</td>
<td>12</td>
</tr>
<tr>
<td>3.5</td>
<td>Tooth Domains</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>Properties Needed for Lubrication Studies</td>
<td>17</td>
</tr>
<tr>
<td>3.7</td>
<td>Results and Discussion</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Load Sharing Analysis</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Hypothesis on Load Sharing</td>
<td>23</td>
</tr>
<tr>
<td>4.2</td>
<td>Normal Forces on Gear Teeth</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Tangential Forces on the Worm</td>
<td>26</td>
</tr>
<tr>
<td>4.4</td>
<td>Results and Discussion</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>Description of Computer Programs</td>
<td>28</td>
</tr>
<tr>
<td>5.1</td>
<td>Graphics Program</td>
<td>28</td>
</tr>
<tr>
<td>5.2</td>
<td>Analyses Program</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>Summary</td>
<td>38</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>Table 1</td>
<td></td>
<td>41</td>
</tr>
<tr>
<td>Figures</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>Appendix 1</td>
<td>Graphics Program</td>
<td>90</td>
</tr>
<tr>
<td>Appendix 2</td>
<td>Listing of the Analyses Program</td>
<td>101</td>
</tr>
<tr>
<td>Appendix 3</td>
<td>The principal Relative Curvature</td>
<td>131</td>
</tr>
</tbody>
</table>
Abstract

The first phase of the study of the performance of a wormgear transmission is reported. In this phase the work included the selection of a double-enveloping wormgear type, and its dimensions, suitable for use in helicopter transmissions; the 3-D graphics representation of the selected wormgear using the I-DEAS software; the analysis of the kinematics of meshing; the analysis of load sharing among the meshing teeth; and the implementation of the analyses in a computer program. The report describes the analyses, their results, and the use of the computer programs.
1 Background and Objectives of the Study

The wormgear is attractive for use in helicopter transmissions because it can accomplish two functions in one stage, viz. transmitting torque from the horizontal engine shaft to the vertical rotor shaft, and simultaneously realizing the required speed ratio. Thus, the use of the wormgear can result in compact transmission design with significant weight saving. However, the wormgear has a serious drawback. Because worm and gear teeth are in pure sliding contact, the sliding friction causes excessive power loss and renders the efficiency of power transmission low. Recently, Chaiko (1990) advanced the concept of using hydrostatic lubrication to remove the sliding friction. He considered a wormgear transmission of realistic dimensions and under realistic operating condition, and found that the efficiency of the transmission could reach 99.14%, and its weight could be only 41% that of the baseline transmission. These figures convincingly demonstrate that the hydrostatically lubricated wormgear can be a candidate of the advanced helicopter transmissions. To further investigate the performance of this type of transmission, an analytical tool capable of generating quantitative data for a wide range of operating conditions is needed. Hence a research proposal was submitted, and a grant was subsequently awarded (NAG3-1316), to develop such a tool.

The project consists of three years of study with the following objectives:

1. Year 1 is to be devoted to the selection of a double-enveloping wormgear type suitable for use in helicopter transmissions, the analysis of the kinematics of meshing, and the analysis of load sharing among the meshing teeth.

2. Year 2 is to be devoted to the selection of a hydrostatic lubrication configuration on the gear surface, and a comprehensive analysis of the hybrid (hydrostatic and hydrodynamic) lubrication at the contact based
on the surface geometry and running conditions obtained in Year 1.

3. Year 3 is to be devoted to the analyses of accessory bearings in the wormgear transmission, and the assembly of the obtained information in a computer program package for the design of this type of transmission. The present report describes the work done in Year 1.

2 Wormgear Geometry

2.1 Selection of Wormgear Type

Wormgears are used to transmit power between two non-parallel non-intersecting shafts. The angle between the shafts is usually 90 deg. There are two common types of wormgears: single-enveloping and double-enveloping. Fig. 2.1 shows a single-enveloping wormgear where the gear envelops the worm. The double-enveloping wormgear is shown in Fig. 2.2, which is like the single-enveloping wormgear except that the worm also envelops the gear.

The double-enveloping wormgear has two contact lines on the mating surface during most of the meshing period. In comparison, the single-enveloping wormgear has only one contact line. Hence, the double-enveloping wormgear is expected to have superior contact properties. The usual way to manufacture a wormgear is first milling a worm, then cutting the gear with a gear hob which has the same shape as the worm. The worm and gear hob are usually ground for high precision applications. The grinding wheel can be a disc (Shen, et al., 1983), Fig. 2.3, or a cone (Simon, 1989), Fig. 2.4. The worm grinding process is called the first enveloping process; and the gear cutting process is called the second enveloping process.

In the first enveloping process the grinding wheel actually rotates around a base circle, with angular velocity $\psi_2$, while the worm rotates about
its own axis with angular velocity $\dot{\psi}_1$. The distance between the worm axis and
the base circle axis is denoted as $a_0$, and the ratio $\dot{\psi}_1$ to $\dot{\psi}_2$ is denoted as $i_0$.
In the case of the disc grinding wheel, the grinding plane may be parallel to
the base circle axis or inclined to it at an angle $\beta$. The center distance $a$
and speed ratio $i$ of the wormgear are actually the respective quantities used
in the second enveloping process. If $a=a_0$ and $i=i_0$, the wormgear is said to
be of the conventional form; otherwise it is said to be of the modified form.

In helicopter applications the torque load and the speed ratio are high.
To accommodate such stringent operating conditions the double-enveloping
wormgear was selected. This is not only because the double-enveloping
wormgear has a larger contact region but also because its contact pattern is
favorable to the establishment of fluid film lubrication. Further, for ease
of manufacturing the disc grinding method was chosen. The analysis presented
below is based on the conventional form, but it can be easily extended to the
modified form.

2.2 Layout of Wormgear Dimensions

Wormgear dimensions are usually determined by following the gear
standards (e.g. AGMA, 1965), which were developed based on many years of
experience. The procedure typically starts with a known speed ratio and a
known center distance. For helicopter applications, the suggested
transmission power output is 4000 kW, the speed ratio is 110, and the center
distance is about 1500 mm. (Chaiko, 1990) These parameters are beyond the
range covered by the AGMA standards. Hence, while the standards were followed
as closely as possible, some modifications had to be made. Table 1 lists the
given and the calculated dimensions and parameters, along with the formulas
used for the calculation.

Common terminologies used to describe the double-enveloping wormgear are
shown in Fig. 2.5. Another term that should be introduced is the mid-plane, which refers to the plane passing through the worm axis and perpendicular to the gear axis. The pitch circle of the gear is measured in the mid-plane. The pitch circle of the worm is measured at the center of the worm. In the double-enveloping wormgear the worm circular pitch and the gear circular pitch are equal. The base circle is used to define the worm profile in the mid-plane. The normal pressure angle is always chosen to be 20 deg.

Several of the selected quantities in Table 1 should be explained. A right hand worm was selected. The pitch diameter of the worm \( d_i \) was chosen to be 300 mm because of strength and rigidity considerations. The number of meshing teeth \( N_p \) was chosen to be 10 instead of the usual 4 or 5 in order to reduce the load level on the meshing teeth. The choice of \( \beta = 5 \) deg was a compromise between acquiring a better contact condition and maintaining an acceptable tooth thickness at the top of the worm thread.

It should be noted that for the double-enveloping wormgear a different choice of dimensions can result in very different surface shape and contact pattern. The results shown below are based on the data of Table 1.

To better visualize the selected wormgear, a software named I-DEAS was used to generate its 3-D graphics, Figs. 2.6 to 2.9. Fig. 2.6 shows the engaged wormgear. The worm and gear are shown separately in Figs. 2.7 and 2.8. Fig. 2.9 depicts a close-up view of the gear teeth.

Details of the I-DEAS program that generated these graphics, and the procedure to run the program, will be described in Section 5.1.
3 Contact analysis

3.1 Method of Enveloping

The geometric and kinematic properties of the wormgear were analyzed with the method of enveloping. To describe the method, let us consider two rotating machine components, m and k, which are in meshing. Let us attach a coordinate system $S_m(x_m, y_m, z_m)$ to the machine component m; and likewise attach a coordinate system $S_k(x_k, y_k, z_k)$ to the machine component k. The two machine components are rotating with the angular coordinates $\psi_m$ and $\psi_k$ with respect to some stationary reference frame. Since the two components are in meshing, there must be some definite relationship between $\psi_m$ and $\psi_k$. Hence only one of them is independent, and either can serve as the running parameter. Let $\psi$ denote this running parameter. The surface of machine component m can be expressed as $f(x_m, y_m, z_m) = 0$ in $S_m$. By coordinate transformation, $f = 0$ becomes a family of surfaces, $F(x_k, y_k, z_k, \psi) = 0$ in $S_k$. Then the surface of the machine component k is the envelope of $F = 0$, viz.

$$F(x_k, y_k, z_k, \psi) = 0$$
$$\frac{\partial F}{\partial \psi} = 0$$

If $\psi_k/\psi_m = \text{constant}$, the surfaces of the two meshing components are said to be conjugate surfaces (Litvin, 1989).

While the idea was simple, the algebraic manipulations needed to carry it out were quite clumsy. Recently, Shen, et al. (1983) reformulated the idea using a set of operators. Because many of the operators involving the transformation of coordinates can be done once for all, the method has become simple as well as elegant. In the special case of conjugate surfaces, the method is even more convenient to use. This will be demonstrated below. The reader is referred to Shen, et al. (1983) for the details of the method.
3.2 Coordinate Systems and Operators

To carry out the analysis, four right hand coordinate systems, \( S_1, S_2, S_j \) and \( S_n \), are set up. See Fig. 3.1.

\( S_j \) and \( S_n \) are stationary systems. \( S_j(x_j, y_j, z_j) \) is located at the worm position. \( x_j \) is along the center line of the wormgear and points to the gear, and \( z_j \) axis is along the worm axis. \( S_n(x_n, y_n, z_n) \) is located at the gear position. \( x_n \) is along the center line but opposite to \( x_j \), and \( z_n \) is along the gear axis.

\( S_1 \) and \( S_2 \) are moving systems. \( S_1(x_1, y_1, z_1) \) rotates with the worm and its \( z_1 \) axis always coincides with the \( z_j \) axis. The worm rotation angle \( \psi_1 \) is the angle between \( x_1 \) and \( x_j \). \( S_2(x_2, y_2, z_2) \) rotates with the gear and its \( z_2 \) axis always coincides with the \( z_n \) axis. (In the first enveloping process \( S_2 \) turns around the base circle with the grinding wheel.) The rotation angle \( \psi_2 \) is the angle between \( x_2 \) and \( x_n \). Thus, \( S_1 \) is fixed to the worm and \( S_2 \) is fixed to the gear; and the worm and gear rotate about \( S_j \) and \( S_n \), respectively. Initially \( S_1 \) coincides with \( S_j \), and \( S_2 \) coincides with \( S_n \).

Let \( A_\psi^{(km)} \) be the coordinate transformation matrix from \( S_m \) to \( S_k \), where the subscript \( ( \ )_\psi \) indicates that \( \psi \) is the running parameter of the coordinate transformation. If the running parameter is understood, the subscript is dropped to simplify the notation. In addition, if the transformation is between two stationary systems, then no running parameter is involved and there will be no subscript. The coordinates in \( S_m \) can be transformed into \( S_k \) as:

\[
\begin{bmatrix}
X_k \\
Y_k \\
Z_k \\
1
\end{bmatrix} = A_\psi^{(km)}
\begin{bmatrix}
X_m \\
Y_m \\
Z_m \\
1
\end{bmatrix}
\]  

(2)
and a function in \( S_m \) can be transformed into \( S_k \) as:

\[
A^{(km)} f(x_m, y_m, z_m) = f(A^{(km)} x_m, A^{(km)} y_m, A^{(km)} z_m)
\]  

(3)

The coordinate transformation matrices from \( S_j \) to \( S_1 \), from \( S_2 \) to \( S_n \), and from \( S_n \) to \( S_j \) are given as:

\[
A^{(1j)} = \begin{bmatrix}
\cos \psi_1 & \sin \psi_1 & 0 & 0 \\
-sin \psi_1 & \cos \psi_1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(4)

\[
A^{(n2)} = \begin{bmatrix}
\cos \psi_2 & -\sin \psi_2 & 0 & 0 \\
\sin \psi_2 & \cos \psi_2 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(5)

\[
A^{(jn)} = \begin{bmatrix}
-1 & 0 & 0 & a_0 \\
0 & 0 & -1 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(6)

The transformation matrix from \( S_2 \) to \( S_1 \) may be written as \( A^{(12)} = A^{(1j)} A^{(jn)} A^{(n2)} \), or:

\[
A^{(12)} = \begin{bmatrix}
-cos \psi_1 \cos \psi_2 & cos \psi_1 \sin \psi_2 & -sin \psi_1 & a_0 \cdot \cos \psi_1 \\
\sin \psi_1 \cos \psi_2 & -\sin \psi_1 \sin \psi_2 & -\cos \psi_1 & -a_0 \cdot \sin \psi_1 \\
-sin \psi_2 & -\cos \psi_2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(7)

Note that the relation \( A^{(mk)} = (A^{(km)})^{-1} \) holds.

In addition to the coordinate transformation matrix, two other operators are defined. One is the differentiation operator:
\[
D_{\psi} = \frac{\partial}{\partial \psi}
\]

\[
D_{\psi}^2 = D_{\psi}D_{\psi} = \frac{\partial^2}{\partial \psi^2}
\]

The other is the similar differentiation operator:

\[
D_{\psi}^{(1)} = A_{\psi}^{(12)} D_{\psi} A_{\psi}^{(21)}
\]

\[
D_{\psi}^{(2)} = A_{\psi}^{(21)} D_{\psi} A_{\psi}^{(12)}
\]

In the above formulas, the subscript \((\ )_{\psi}\) may be dropped if no confusion can arise.

Several needed properties of these operators are given below:

\[
D_{A}^{(km)} f(x, y, z, \psi)
\]

\[
= A^{(km)} Df + (A^{(km)} f_y) \frac{\partial}{\partial x} A^{(km)} x + (A^{(km)} f_z) \frac{\partial}{\partial y} A^{(km)} y + (A^{(km)} f_{\psi}) \frac{\partial}{\partial z} A^{(km)} z
\]

\[
D_{A}^{(ij)} f = A^{(ij)} D^{(j)} D^{(i)} f
\]

\[
A^{(ij)} Df = D^{(i)} A^{(ij)} f
\]

\[
D^{(j)} f(x, y, z, \psi) = Df + f_{xj} D^{(j)} x + f_{yi} D^{(j)} y + f_{zi} D^{(j)} z
\]

where

\[
f_{xj} = \frac{\partial f}{\partial x}, \quad f_{yi} = \frac{\partial f}{\partial y}, \quad f_{zi} = \frac{\partial f}{\partial z}
\]

In the above formulas, \(k\) and \(m\) can be 1, 2, \(j\) and \(n\); but \(i\) and \(j\) can only be 1 and 2.

To distinguish between the first and the second enveloping processes, \(\psi\) is used as the running parameter for the first enveloping process and \(\theta\) is used for the second enveloping process. The order of operations with respect to different running parameters may be interchanged, e.g.
\[ D_\theta A_\psi^{(2n)} = A_\psi^{(2n)} D_\theta \]  

(14)

The derivation of the above and other properties of the operators can be found in Shen, et al. (1983).

It should be noted that different symbols, \( i_0 \) and \( i \), are used to denote the respective speed ratios in the first and the second enveloping processes, i.e.

\[
\begin{align*}
  i_0 &= \frac{\Psi_1}{\Psi_2} = \frac{\psi_1}{\psi_2} \\
  i &= \frac{\theta_1}{\theta_2} = \frac{\theta_1}{\theta_2}
\end{align*}
\]  

As said before, if \( i_0 = i \) and \( a_0 = a \) the wormgear is of the conventional form; otherwise it is of the modified form.

3.3 Worm Surface

The worm surface is formed by a grinding wheel in the first enveloping process. The surface of the grinding wheel is called the original generating surface. For a disc grinding wheel this original generating surface is a plane. For a cone grinding wheel the original generating surface is a line on the cone.

The original generating surface can be expressed in the \( S_2 \) system as \( f(x_2, y_2, z_2) = 0 \). In the first enveloping process \( S_2 \) rotates about \( z_2 \) with the rotation angle \( \psi_2 \), and \( S_1 \) rotates about \( z_1 \) with the rotation angle \( \psi_1 \). As a result of rotation, the original generating surface forms a one-parameter family in \( S_1 \), \( F(x_1, y_1, z_1, \psi) = A_\psi^{(12)} f(x_2, y_2, z_2) = 0 \). Because \( \psi_1/\psi_2 = \text{constant} \), the running parameter \( \psi \) can be either \( \psi_1 \) or \( \psi_2 \). Then, the envelope of the family
of surfaces is the worm surface, which is given as $F=0$ and $D_fF=0$.

Considering only the case of the selected wormgear, the original generating surface is a plane in $S_2$, which is:

$$f(x_2, y_2, z_2) = y_2 - z_2 \tan \beta + r_b = 0 \quad (16)$$

After coordinate transformation into $S_1$, the one-parameter family of planes becomes:

$$F = \Delta^{(12)} f = x_1 (\cos \psi_1 \sin \psi_2 + \tan \beta \sin \psi_1)$$
$$+ y_1 (-\sin \psi_1 \sin \psi_2 + \tan \beta \cos \psi_1)$$
$$- z_1 \cos \psi_2 - (a_0 \sin \psi_2 - r_b) = 0 \quad (17)$$

Then, the worm surface may be represented by the following two equations:

$$F = x_1 (\cos \psi_1 \sin \psi_2 + \tan \beta \sin \psi_1) + y_1 (-\sin \psi_1 \sin \psi_2 + \tan \beta \cos \psi_1)$$
$$- z_1 \cos \psi_2 - (a_0 \sin \psi_2 - r_b) = 0$$
$$\Delta \psi F = x_1 (\cos \psi_1 \cos \psi_2 + i_0 \tan \beta \cos \psi_1 - i_0 \sin \psi_1 \sin \psi_2)$$
$$+ y_1 (-\sin \psi_1 \cos \psi_2 - i_0 \tan \beta \sin \psi_1 - i_0 \cos \psi_1 \sin \psi_2) + z_2 \sin \psi_2 - a_0 \cos \psi_2 = 0 \quad (18)$$

Using the relation $\psi_1/\psi_2 = i_0$ and eliminating $\psi_1$ and $\psi_2$ from Eqs. 18, an explicit equation, which is nonlinear, of the worm surface can be obtained.

For a given value of $\psi_1$, Eqs. 18 represent the equation of a straight line. Recall that the first equation is the equation of the original generating plane expressed at the position $\psi_1$. Hence the straight line is on both the generating plane and the enveloping surface. This straight line is the contact line between the worm and the disc grinding wheel at the position $\psi_1$. It can also be seen that the worm surface is the set of all the straight contact lines. Fig. 3.2 depicts this situation.
According to differential geometry, if the planes in a one-parameter family are not parallel and there is not a point common to all the planes, then the envelope of this family is the tangent surface of its edge curve (Kreyszig, 1959). In Fig. 3.2 the curve labelled $\Gamma$ is the edge curve. The edge curve is given by these equations:

\[ F = x_1(\cos \psi_1 \sin \psi_2 + \tan \beta \sin \psi_1 + y_1(-\sin \psi_1 \sin \psi_2 + \tan \beta \cos \psi_1 - z_1 \cos \psi_2 - (a_0 \sin \psi_2 - r_h)) = 0 \]

\[ D_F F = x_1(\cos \psi_1 \cos \psi_2 + i_0 \tan \beta \cos \psi_1 - i_0 \sin \psi_1 \sin \psi_2) + y_1(-\sin \psi_1 \cos \psi_2 - i_0 \tan \beta \sin \psi_1 - i_0 \cos \psi_1 \sin \psi_2) + z_1 \sin \psi_2 - a_0 \cos \psi_2 = 0 \]

\[ D_F D_F F = x_1(i_0^2 \cos \psi_1 \sin \psi_2 + i_0^2 \tan \beta \sin \psi_1 + 2i_0 \sin \psi_1 \cos \psi_2 + \cos \psi_1 \sin \psi_2) + y_1(-i_0^2 \sin \psi_1 \sin \psi_2 + i_0^2 \tan \beta \cos \psi_1 + 2i_0 \cos \psi_1 \cos \psi_2 - \sin \psi_1 \sin \psi_2) - z_1 \cos \psi_2 - a_0 \sin \psi_2 = 0 \]
The edge curve divides the envelope surface into two branches, of which one is the worm surface. If the worm thread domain determined by the selected dimensions intersects the edge curve, the surface of the disc grinding wheel would extend over the edge curve. The crossed-over portion of the grinding surface would undercut the worm thread. Undercutting the thread would decrease the length of contact lines, decrease the root thickness of the worm thread, and consequently decrease the load carrying capacity of the thread. Hence, wormgear dimensions should be chosen properly so that the worm thread domain is away from the edge curve.

3.4 Gear Surface

The same procedure applies to the second enveloping process. In this process the gear hob has the same shape as the worm surface. Hence, the generating surface is given by Eqs. 18 in $S_1$. As the result of rotation, this generating surface forms a one-parameter family of surfaces in $S_2$. The running parameter for this process can be either $\theta_1$ or $\theta_2$, while $\psi$ becomes a geometric parameter already in existence in the worm surface equation. The procedure to obtain the gear surface may be as follows: By eliminating $\psi$ from Eqs. 18, one obtains an explicit equation of the worm surface. This equation may be written as:

$$F^* (x_1, y_1, z_1) = 0$$

(20)

The one-parameter family of surfaces in $S_2$ is obtained by coordinate transformation:

$$f^{(2)} (x_2, y_2, z_2, \theta) = A_0^{(21)} F^* = 0$$

(21)

The envelope of the family:

$$f^{(2)} = 0$$

$$D_h f^{(2)} = 0$$

(22)

is then the gear surface.
But it is not easy to obtain \( F^* = 0 \). Neither is it necessary. Assume that \( \psi = \psi(x_1, y_1, z_1) \) is obtained by solving the second equation of Eqs. 18, \( D_\psi F = 0 \). Substituting it into the first equation of Eqs. 18, \( F = 0 \), yields the explicit worm surface equation

\[
F^* = F(x_1, y_1, z_1, \psi(x_1, y_1, z_1)) = 0
\]  

(23)

Then, the equation \( D_\theta F^{(2)} = D_\theta A_\theta^{(21)} F^* = 0 \) can be written as:

\[
D_\theta A_\theta^{(21)} F^* = A_\theta^{(21)} D_\theta F^* + (A_\theta^{(21)} F^{*x_1}) D_\theta A_\theta^{(21)} x_1 + (A_\theta^{(21)} F^{*y_1}) D_\theta A_\theta^{(21)} y_1 + (A_\theta^{(21)} F^{*z_1}) D_\theta A_\theta^{(21)} z_1 = 0
\]  

(24)

But from Eq. 23,

\[
\begin{align*}
F^{*x_1} &= F_{x_1} + F_\psi x_1 \\
F^{*y_1} &= F_{y_1} + F_\psi y_1 \\
F^{*z_1} &= F_{z_1} + F_\psi z_1
\end{align*}
\]  

(25)

Then, since \( D_\psi F = 0 \) and Eq. 23 does not contain \( \theta \), Eq. 24 reduces to:

\[
D_\theta A_\theta^{(21)} F^* = (A_\theta^{(21)} F_{x_1}) D_\theta A_\theta^{(21)} x_1 + (A_\theta^{(21)} F_{y_1}) D_\theta A_\theta^{(21)} y_1 + (A_\theta^{(21)} F_{z_1}) D_\theta A_\theta^{(21)} z_1 = D_\theta A_\theta^{(21)} F = 0
\]  

(26)

which, on multiplification by \( A_\theta^{(21)} A_\theta^{(12)} \), becomes \( A_\theta^{(21)} D_\theta^{(11)} F = 0 \). Thus, the gear surface is given by:

\[
\begin{align*}
A_\theta^{(21)} F &= 0 \\
A_\theta^{(21)} D_\theta F &= 0 \\
A_\theta^{(21)} D_\theta^{(11)} F &= 0
\end{align*}
\]  

(27)

which can be written as:

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
y_2 \\
z_2 \\
1
\end{bmatrix} = 0
\]  

(28)

where
\[ a_{11} = -\sin \psi_2 \cos \theta_2 \cos (\psi_1 - \theta_1) - \tan \beta \cos \theta_2 \sin (\psi_1 - \theta_1) + \cos \psi_2 \sin \theta_2 \]
\[ a_{12} = \sin \psi_2 \sin \theta_2 \cos (\psi_1 - \theta_2) + \tan \beta \sin \theta_2 \sin (\psi_1 - \theta_2) + \cos \psi_2 \cos \theta_2 \]
\[ a_{13} = \sin \psi_2 \sin (\psi_1 - \theta_1) - \tan \beta \cos (\psi_1 - \theta_1) \]
\[ a_{14} = a_3 [\sin \psi_2 \cos (\psi_1 - \theta_1) + \tan \beta \sin (\psi_1 - \theta_1)] - a_0 \sin \psi_2 + \tau_p \]  \hspace{1cm} (29)

\[ a_{21} = \sin \psi_2 \cos \theta_2 \sin (\psi_1 - \theta_1) - \tan \beta \cos \theta_2 \cos (\psi_1 - \theta_1) \]
\[ - [\sin \psi_2 \sin \theta_2 + \cos \psi_2 \cos \theta_2 \cos (\psi_1 - \theta_1)] / i_0 \]
\[ a_{22} = -\sin \psi_2 \sin \theta_2 \sin (\psi_1 - \theta_1) + \tan \beta \sin \theta_2 \cos (\psi_1 - \theta_1) \]
\[ - [\sin \psi_2 \cos \theta_2 - \cos \psi_2 \sin \theta_2 \cos (\psi_1 - \theta_1)] / i_0 \]
\[ a_{23} = \sin \psi_2 \cos (\psi_1 - \theta_1) + \tan \beta \sin (\psi_1 - \theta_1) + [\cos \psi_2 \sin (\psi_1 - \theta_1)] / i_0 \]
\[ a_{24} = -a_3 [\sin \psi_2 \sin (\psi_1 - \theta_1) - \tan \beta \cos (\psi_1 - \theta_1) - \cos \psi_2 \cos (\psi_1 - \theta_1) / i_0] - a_0 \cos \psi_2 / i_0 \]  \hspace{1cm} (30)

\[ a_{31} = a_{12} - i \cdot [\sin \psi_2 \cos \theta_2 \sin (\psi_1 - \theta_1) - \tan \beta \cos \theta_2 \cos (\psi_1 - \theta_1)] \]
\[ a_{32} = -a_{11} + i \cdot [\sin \psi_2 \sin \theta_2 \sin (\psi_1 - \theta_1) - \tan \beta \sin \theta_2 \cos (\psi_1 - \theta_1)] \]
\[ a_{33} = -i \cdot [\sin \psi_2 \cos (\psi_1 - \theta_1) + \tan \beta \sin (\psi_1 - \theta_1)] \]
\[ a_{34} = a \cdot i \cdot [\sin \psi_2 \sin (\psi_1 - \theta_1) - \tan \beta \cos (\psi_1 - \theta_1)] \]  \hspace{1cm} (31)

In Eqs. 27 there are three equations and five unknowns. Using the relations \( \psi_1 / \psi_2 = i_0 \) and \( \theta_1 / \theta_2 = -i \) the explicit form of the envelope can be obtained by eliminating \( \psi \) and \( \theta \). Eqs. 27, along with the relation limiting the gear tooth domain, then provide the shape of the gear tooth.

Consider the case of the modified form. For a given value of \( \theta \), Eqs. 27
produce a curve. The curve is the contact line between the worm and the gear at this \( \theta \) position. The curve goes over the gear surface twice as shown in Fig. 3.3. For fixed values of \( \psi \) and \( \theta \), Eqs. 27 represent a point on the contact line.

Consider the case of the conventional form. For a given \( \theta \), if one chooses \( \psi=\theta \), the third and the second equations of Eqs. 27 are identical. Then Eqs. 27 reproduce a straight contact line of the first enveloping process. This kind of contact lines is called the first enveloping contact lines (the solid lines in Fig. 3.4). The set of all these straight lines reproduce a part of the original generating plane. For a given \( \theta \), let \( \theta=\psi \), Eqs. 27 give a curved contact line, called the second enveloping contact line (the dashed lines in Fig. 3.4). The set of all the curved contact lines form the curved part of the gear surface. Thus, in the case of the conventional form, the gear surface consists of a planar region and a curved region. For a certain range of \( \theta \) values these two regions overlap each other. The overlapped region is shown in Fig. 3.4 where the dashed lines and the solid lines cross. This situation is impossible to occur in reality. From a view of the \((y_2,z_2)\) plane it is clear that the curved region is farther away from the solid interior than the planar region. Hence, the curved region is fictitious, which is actually hobbed off during manufacturing. Analytically the principal relative curvature is used to sort out the fictitious part (Shen, et al, 1983). The relative curvature is the difference of curvatures of the worm and gear surfaces at the contact point. Because the gear tooth envelops the worm thread, at the contact point the curvature of the gear tooth must not be greater than that of the worm thread (absolute values). Hence, if the relative curvature is negative, the produced gear surface is not real. The derivation of the expression for the principal relative curvature is given in Appendix 3.

From the above description it can be seen that, in the double-enveloping
wormgear, the worm and gear are in line contact and there are usually two
contact lines on each pair of the meshing teeth.

3.5 Tooth domains

After the equations of the worm and gear surfaces are obtained, the
tooth domains need be specified to determine what portions of these surfaces
are the actual worm thread and gear tooth. Table 1 can be used to set up the
following geometrical limitations on the worm and gear surfaces. The worm and
gear tooth domains are shown in Fig. 3.5. Several new symbols that appear in
the following equations are also defined in this figure.

The maximum and minimum radii of the worm thread are:

\[ r_{\text{max}} = a - \sqrt{(d_{t1}/2+c)^2 - z_1^2} \]
\[ r_{\text{min}} = a - \sqrt{(d_{g1}/2+c)^2 - z_1^2} \]
\[ r_v = \sqrt{x_v^2 + y_v^2} \]
\[ r_{\text{min}} \leq r_v \leq r_{\text{max}} \]  \hspace{1cm} (32)

where \( r_v \) is the radius of any point on the worm surface.

The maximum and minimum radii of the gear tooth are:
\[ R_{\text{max}} = a - \sqrt{(d_1/2)^2 - (b_2/2)^2} \]

\[ r_{c_{\text{max}}} = \min \{ (a + \Delta a) - \sqrt{h_{\text{min}}^2 - (z_2/2)^2}, R_{\text{max}} \} \]

\[ r_{c_{\text{min}}} = (a + \Delta a) - \sqrt{h_{\text{max}}^2 - (z_2/2)^2} \]

\[ r_{c_0} = \sqrt{x_2^2 + y_2^2} \]

\[ r_{c_{\text{min}}} \leq r_{c_0} \leq r_{c_{\text{max}}} \] (33)

where \( r_{c_0} \) is the radius of any point on the gear surface, and

\[ h_{\text{max}} = a - \sqrt{(d_2/2)^2 - (b_1/2)^2} \]

\[ h_{\text{min}} = a - \sqrt{(d_2/2)^2 - (b_1/2)^2} \] (34)

\[ \Delta a = (h_{\text{max}} + d_2/2) - a \]

3.6 Properties Needed for Lubrication Studies

To carry out the lubrication analysis several geometric and kinematic properties are needed. These include the shape of the two contacting surfaces, the relative velocity between the two surfaces at a contact point, and the normal forces acting on the meshing teeth, etc.

3.6.1 Surfaces in Stationary Systems

Eqs. 18 and Eqs. 27 are obtained in the moving coordinate systems, where the worm and the gear appear stationary. To see the moving worm and gear surfaces, these equations must be expressed in the stationary systems. At a given instance, or for a given gear position \( \theta \), there are two ways to obtain the worm and gear surfaces in a stationary coordinate system. One way is
calculating the coordinates in the moving system first, then transforming these points into the stationary system. The other way is transforming the equations into the stationary system first. In either way the coordinate transformation matrices are specialized to the given position. To distinguish this position from the running parameter \( \theta \), it is denoted as \( \theta^* \). Thus the worm surface in \( S_j \) is given as:

\[
A_{\theta^*}^{(21)} F = 0 \\
A_{\theta^*}^{(21)} D_{\theta^*} F = 0
\]

(35)

the gear surface in \( S_n \) is given as:

\[
A_{\theta^*}^{(n2)} A_{\theta}^{(21)} F = 0 \\
A_{\theta^*}^{(n2)} A_{\theta}^{(21)} D_{\theta} F = 0 \\
A_{\theta^*}^{(n2)} A_{\theta}^{(21)} D_{\theta}^{(1)} F = 0
\]

(36)

and the gear surface in \( S_j \) is given as:

\[
A^{(jn)} A_{\theta^*}^{(n2)} A_{\theta}^{(21)} F = 0 \\
A^{(jn)} A_{\theta^*}^{(n2)} A_{\theta}^{(21)} D_{\theta} F = 0 \\
A^{(jn)} A_{\theta^*}^{(n2)} A_{\theta}^{(21)} D_{\theta}^{(1)} F = 0
\]

(37)

3.6.2 The Relative Velocity at a Contact Point

In Eqs. 35 to 37 if one sets \( \theta = \theta^* \), the contact lines expressed in the stationary systems are obtained for the given \( \theta^* \) position. Then the relative velocity at a contact point can be calculated, which is the difference of the worm velocity and the gear velocity at this contact point. In \( S_j \) the relative velocity is given as:
\[ V_z = V_1 - V_2 \]
\[ V_1 = y_j \omega_1 \mathbf{i} + x_j \omega_1 \mathbf{j} \]
\[ V_2 = -y_n \omega_2 (-\mathbf{i}) + x_n \omega_2 (-\mathbf{k}) \]  

(38)

where \( x_j, y_j, z_j \) are the coordinates of the contact point in \( S_j \); \( x_n, y_n, z_n \) are the corresponding coordinates in \( S_n \); \( \mathbf{i}, \mathbf{j} \) and \( \mathbf{k} \) are the unit vectors in \( S_j \).

To facilitate further development, let us introduce the term "center point". A center point is the point of intersection between a tooth surface and the gear pitch circle. In the meshing zone there is a gear center point on every gear tooth. Correspondingly there is a worm center point on the worm coil in meshing with this gear tooth. The worm center point is in general slightly different than the gear center point, unless the center point is a contact point. In the analysis below, the normal and friction forces between a pair of meshing teeth are taken as acting at the worm center point, which for brevity will be referred to simply as the center point.

In case that the center point is a contact point, the relative velocity is:

\[ V_z = \sqrt{V_1^2 + V_2^2} = \sqrt{(x_j \omega_1)^2 + (d_2 \omega_2/2)^2} \]
\[ y_j = 0, \quad x_n^2 + y_n^2 = (d_2/2)^2 \]  

(39)

3.6.3 Normal Direction of the Worm Surface

Because the worm surface is formed by the original generating plane, the normal direction at a point of the worm surface is the same as the normal direction of the original generating plane. Further, since the worm surface and the original generating plane contact in a straight line, all the points on the contact line have the same normal direction. In \( S_1 \) this normal
direction is:

\[ n_x = \cos \psi_1 \sin \psi_2 + \tan \beta \sin \psi_1 \]
\[ n_y = -\sin \psi_1 \sin \psi_2 + \tan \beta \cos \psi_1 \]
\[ n_z = -\cos \psi_2 \] (40)

Transforming Eqs. 40 into \( S_2 \), the normal direction of the gear surface at the contact point can also be obtained. Note that since the normal direction of the worm is outward from the worm body, the normal direction of the gear is then inward to the gear body. The normal direction can be further transformed into a stationary coordinate system. Thus, for the given position \( \theta^* \), the normal direction in \( S_3 \) is:

\[ n_{x'} = \cos (\psi_1 - \theta_1^*) \sin \psi_2 + \tan \beta \sin (\psi_1 - \theta_1^*) \]
\[ n_{y'} = -\sin (\psi_1 - \theta_1^*) \sin \psi_2 + \tan \beta \cos (\psi_1 - \theta_1^*) \] (41)
\[ n_{z'} = -\cos \psi_2 \]

3.6.4 Clearance Function

Once the worm and gear surfaces are found, the clearance between the two mating surfaces can be determined. This information is of vital importance in the analysis of fluid film lubrication. This clearance is defined to be the gap between the two surfaces measured in the direction of the common normal on the contact line. Clearly, as the distance from the contact line increases, the gap so measured becomes less representative of the true distance between the two surfaces. But this is immaterial, because as the gap becomes large, the effects of fluid film lubrication also become negligible. The clearance function can be obtained numerically as follows: (1) Choose the grid system on the gear tooth and obtain the coordinates \((x_2, y_2, z_2)\) of the grid points. (2) Calculate the normal direction at a contact point and obtain the equation
of the straight line passing through a grid point and in this normal
direction. (3) Find the point of intersection between this straight line and
the worm surface. The gap at the grid point is then the distance between the
grid point and the point of intersection. It is convenient to carry out this
calculation in the lubrication analysis, where the grid system has to be
selected first.

3.7 Results and Discussion

Sample results were computed for the selected wormgear. Fig. 3.6 shows
the wormgear position at $\theta^* = 0$, where ten pairs of teeth are in meshing,
beginning from gear tooth No.2 and ending with gear tooth No.11. The contact
lines on several of these teeth are shown in Figs. 3.7 as viewed in the $(x_n,
z_n)$ plane. In each of these figures the circles are on the gear tooth, whose
domain is shown by the solid lines; the stars are on that portion of the worm
thread that overlaps the gear tooth. Note also that the unit of the
coordinates in these figures is millimeter, and the teeth are shown in their
true proportion. On gear tooth No.2, Fig. 3.7(a), the two contact lines are
far from the mid-plane ($z_n = 0$). These contact lines move toward the mid-plane
as the tooth number increases. See Figs. 3.7(a) to 3.7(d). On gear tooth
No.10, Fig. 3.7(e), the second enveloping contact line crosses over the first
enveloping contact line. The crossed-over part is nonexistent, as is
explained previously. On gear tooth No.11, Fig. 3.7(f), the entire second
enveloping contact line falls below the first enveloping contact line, and
becomes non-existent. Thus, toward the end of the meshing zone, the total
length of the contact lines diminishes. From these figures it can also be
seen that there are two contact lines on the majority of the teeth in the
meshing zone; these contact lines are located rather symmetrically on the two
sides of the mid-plane and they run across the tooth face more or less in the
direction of the $x_2$ axis.
The total length of the contact lines on each meshing tooth is plotted in Fig. 3.8. The situation corresponding to the position $\theta^{*}=0$ is shown in Fig. 3.8(a). When the gear rotates one degree, i.e. $\theta^{*}=1$ deg, tooth No.11 moves out of, while tooth No.1 has not moved into, the meshing zone. Hence there are only nine pairs of teeth in meshing. The distribution of the contact line lengths in this situation is shown in Fig. 3.8(b). When the gear rotates one more degree, i.e. $\theta^{*}=2$ deg, tooth No.1 has moved into, while tooth No.10 still remains in, the meshing zone. There are again ten pairs of teeth in meshing. The corresponding distribution of the contact line lengths is shown in Fig. 3.8(c). Since the angle between two neighboring teeth is $360/N_2 = 3.273$ deg, if the gear further rotates 1.27 deg, the situation of Fig. 3.8(a) re-appears.

The variation of the contact line length on one gear tooth as it moves through the meshing zone is shown in Fig. 3.9, which is essentially the three graphs of Figs. 3.8 combined.

The radial position of the edge curve on the worm surface is shown in Fig. 3.10. For comparison, the radial position of the hour-glass-shaped worm pitch cylinder is plotted as the dashed curve. Since the tooth depth is about 20 mm, it can be seen that the edge curve is located far below the worm thread. Hence there is no problem of undercutting the worm thread.

From the above results one can identify the following merits of the selected wormgear:

1. On the majority of the meshing teeth there are two contact lines. This means that the contact region on a tooth of the double-enveloping wormgear is larger than that of the single-enveloping wormgear, which only possesses one contact line on a meshing tooth.

2. The contact lines run across the tooth face more or less in the direction of the $x_2$ axis. Since the relative velocity of the meshing
surfaces is nearly perpendicular to the \( x_2 \) axis, the two directions meet at a large angle. This feature is favorable to the establishment of fluid film lubrication at the contact.

3. On the majority of the meshing teeth, the pattern of the contact lines is nearly symmetrical with respect to the center point. This feature allows a major simplification to be made in the load sharing analysis.

4. With the chosen gear dimensions there is no problem of undercutting the worm thread.

4 Load Sharing Analysis

The methodology of determining how the load is distributed among meshing teeth is not well developed. Rational analyses are rare and complicated. For spur and helical gears, the load sharing analyses are usually based on the variation of tooth stiffness along contact lines (Elkholy, 1989). In the practical design of these gears, the approach is to analyze an idealized meshing model, then apply various factors to modify the results. To estimate the magnitudes of these load distribution factors, the common approach is to assume that gear teeth behave like a series of thin slices that deflect independently (Steward, 1989). It is doubtful that these methods are suitable to the double-enveloping wormgear, because its contact pattern is vastly different than those of the spur and helical gears. Hence, for the double-enveloping wormgear, there does not exist a theory of load sharing. Neither are there adequate experimental data to guide the analysis and design. In the following, we shall base our analysis of load sharing on a hypothesis.

4.1 Hypothesis on Load Sharing

At a contact point the forces between the meshing surfaces consist of a normal component, that acts in the direction of the common normal, and a
friction component. The latter acts in the direction of the relative velocity and has a magnitude equal to the normal component multiplied by the Coulomb friction coefficient. Since the direction of the normal and that of the relative velocity vary with position, these forces vary from contact point to contact point not only in magnitude but also in direction. However, the contact analysis in Section 3 reveals that the contact line pattern on each tooth is nearly symmetrical with respect to the center point. In addition, because of the large speed ratio (and consequently the small lead angle) of the selected wormgear, the variation of the normal direction on a tooth surface is small. These two features allow the following simplifications to be made:

1. The resultant normal force on a tooth is proportional to the contact line length on the tooth. It acts at the center point and is in the normal direction of the center point.
2. The resultant friction force on a tooth acts at the center point and is in the direction of the relative velocity at the center point.

The above simplifications may be considered as a hypothesis of load sharing, whose validity should be verified later when experimental data or results of a deformation analysis become available.

4.2 Normal Forces on Gear Teeth

The coordinates \((x_{mp}, z_{mp})\) in \(S_j\) and the parameter \(\psi_{mp}\) of the worm center point for the \(m\)th pair of meshing teeth can be found by solving Eqs. 35 and the following set of equations describing the gear pitch circle:

\[
(a-x_j)^2 + z_j^2 = \left( \frac{d_j}{2} \right)^2
\]

\[y_j = 0\]  

(42)

The normal direction at the center point can be calculated from Eqs. 41, where \(\psi_{mp}\) is to be used for \(\psi\), and \(\theta^\ast\) is the given position.
From Fig. 4.1, where friction is not shown, the torque contributed by the normal force on the gear tooth of the mth meshing pair is:

\[ T_{c_m} = F_{c_m}^n r_{c_m} \]

\[ r_{c_m} = [ \left( a - x_p \right) \frac{|n_{jx}|}{n_y} - z_p \frac{|n_{jx}|}{n_y} ]_m \]

\[ n_y = \sqrt{n_{jx}^2 + n_{jy}^2 + n_{js}^2} \]

where \( r_{c_m} \) is an effective radius; \( n_{jx}, n_{jy}, n_{js} \) are as given in Eqs. 41; \([ ]_m\) denotes the mth meshing pair.

By hypothesis, \( F_{c_m}^n \) is proportional to the length of contact lines. Besides, \( T_G = \sum T_{c_k} \). Hence the formula:

\[ F_{c_m}^n = \frac{T_G L_m}{\sum_{k=1}^{N_p'} L_k r_{c_k}} \]

where

- \( F_{c_m}^n \) is the normal force acting on the gear tooth of the mth meshing pair,
- \( T_G \) is the required output torque from the gear shaft,
- \( L_m \) is the length of contact lines on the mth pair of the meshing teeth, and
- \( N_p' \) is the actual number of meshing pairs.

The relative motion between the worm and gear is pure sliding. Hence, friction plays an important role in the performance of the wormgear (Shigley and Mitchell, 1983). Let \( \mu \) be the Coulomb friction coefficient, then the magnitude of friction is \( \mu F_{c_m}^n \), and its direction on the gear is the same as the direction of the relative velocity \( v_1 - v_2 \).

At the center point of the mth pair the worm velocity is parallel to the
(x_j,y_j) plane and equal to x_m^o \omega_1, the gear velocity is in the (x_j,z_j) plane and equal to d_2 \omega_2/2. In S_j the direction of the relative velocity is given by Eqs. 39 and is shown in Fig. 4.2.

Let \( \text{F}_{\text{fom}}^n \) be the normal force on the gear tooth of the mth meshing pair when friction is present. The torque due to the friction force, \( \mu \text{F}_{\text{fom}}^n \), is:

\[
T_{\text{fm}} = - \frac{\text{F}_{\text{fom}}^n}{d_2} \rho_{\text{fom}}
\]

\[
\rho_{\text{fom}} = \left[ \frac{\mu \omega_2 d_2}{\sqrt{(x_p \omega_1)^2 + (d_2 \omega_2/2)^2}} \right]_{\text{m}} \tag{45}
\]

Then, \( \text{F}_{\text{fom}}^n \) is found to be:

\[
\text{F}_{\text{fom}}^n = \frac{T_C L_\text{m}}{\sum_{k=1}^{k' x_j} L_k (\rho_{\text{ek}} - \rho_{\text{efk}})} \tag{46}
\]

4.3 Tangential Forces on the Worm

The force acting on the worm can be decomposed into the radial, tangential, and axial components, viz. \( \text{F}_{\text{bm}} x_j^1, \text{F}_{\text{bm}} y_j^1, \) and \( \text{F}_{\text{bm}} z_j^1 \) (Shigley and Mitchell, 1983). See Fig. 4.1. The tangential component balances the input worm torque, which in turn determines the input power. The ratio of the input power without friction to that with friction, averaged over an interval where the gear rotates 3.273 deg, determines the efficiency of the wormgear.

The tangential force on the worm for the mth pair of the meshing teeth is:

\[ \text{without friction} \]

\[
\text{F}_{\text{bm}} x_j^1 = - \text{F}_{\text{fom}}^n \left( \frac{|y_j^1|}{n_\text{W}} \right)_{\text{m}} \tag{47}
\]

26
with friction

\[ F_{fkm}^{\gamma_2} = F_{fcm}^n \left[ \frac{\mu x_p \omega_1}{\sqrt{(x_p \omega_1)^2 + (d_2 \omega_2 / 2)^2}} + \frac{|n_{fy}|}{n_w} \right]_m \]  

(48)

Since the worm torque equals the sum of the torques contributed from the meshing teeth, it is:

without friction

\[ T_N = \sum_{m=1}^{n_p} F_{km}^{\gamma_2} x_{mp} \]  

(49)

with friction

\[ T_{fw} = \sum_{m=1}^{n_p} F_{fkm}^{\gamma_2} x_{mp} \]  

(50)

4.4 Results and Discussion

Load sharing was computed for a power output of 4000 kW and a rotation speed of 130 r/min of the gear shaft. To account for the effect of friction at the worm and gear contact, a Coulomb friction coefficient value of 0.03 was used in the computation. The results are shown in Figs. 4.3 to 4.5. In these figures, the stars represent the case with friction and the circles the case without friction.

Figs. 4.3(a) to 4.3(c) present the normal forces acting on the meshing gear teeth for the three wormgear positions, viz. \( \theta^* = 0, 1 \) and 2 deg. These figures correspond to the cases of Figs. 3.8(a) to 3.8(c). The variation of the normal force on one gear tooth as it moves through the meshing zone is shown in Fig. 4.4, which is essentially the three graphs of Figs. 4.3 combined. It is seen that the difference between including and not including friction cannot be detected. This implies that, for a given gear output power, the lubricant film at the worm and gear contact must withstand almost
the same normal force whether friction is present or not.

The tangential forces acting on the worm thread corresponding to the three worm gear positions of $\theta^* = 0, 1$ and 2 deg are shown in Figs. 4.5(a) to 4.5(c). Here, including or not including friction makes a big difference. This is because for the selected worm gear the direction of the relative velocity, which is the direction of friction, is very close to that of the tangential force on the worm. Thus, a large portion of the input torque must be spent to overcome the frictional torque. Hence, effective lubrication at the contact can drastically reduce the input power required of the worm.

5 Description of Computer Programs

Two computer programs were developed in this work, the graphics program and the analyses program. The graphics program is for creating the 3-D graphics of the selected worm gear. The analyses program is for implementing the geometric, kinematic and load sharing analyses.

5.1 Graphics Program

The graphics program is named NASA.PRG and is written in I-DEAS programming language, IDEAL.

5.1.1 I-DEAS Software

I-DEAS (Integrated Design Engineering Analysis System) is a widely used commercial software (SDRC, 1991). It has three program families: solid modeling, finite element and advanced technique. Solid modeling is used to geometrically create an object. The tasks in the solid modeling family include construction geometry, object modeling, mechanism design, drawing
Construction geometry is used to create profiles, wireframes and skin groups. A profile is defined by a boundary that lies in a plane. The boundary can consist of lines, arcs, circles and splines, which can be defined by points, centers and radii. In the boundary holes can be defined.

Object modeling is used to design mechanical parts which are modeled as objects. There are several ways to create an object: (1) Use primitives such as blocks, cylinders, cones and spheres. (2) Extrude or revolve a profile into a solid. (3) Create a group of profiles and then generate a solid through the group. This task also provides jointing, cutting and assembling options. Once two objects are created, a third object can be created from them by using these options. The created object can be displayed with options such as scale, hidden line, shade display, etc. The created object can also be rotated, moved and oriented to the required position; and saved and printed.

I-DEAS can be run interactively by selecting from the menu on the screen using the mouse, or by inputing from the keyboard or through a program.

I-DEAS has a high level programming capability using the language IDEAL. The commands can be written in a program file which is then executed. The programming capability permits arithmetic expressions, variables, mathematic functions, and algebraic operations; it also provides commands for branching and do-looping, as well as controlling the input and output.

5.1.2 Description of the Graphics Program

The process of creating a double-enveloping wormgear by using I-DEAS is similar to that of manufacturing the wormgear. This is carried out as
follows:

First a worm blank and a disc grinding wheel are created. Once these two objects are created, the worm can be generated by cutting the worm blank with the grinding wheel. In the worm creating process the motion of the worm blank and the grinding wheel is similar to that of the first enveloping process.

Then the gear blank is created. The gear is created by cutting the gear blank with the previously created worm. The motion of the gear blank and the worm is similar to that of the second enveloping process.

The motions are simulated by rotating the blanks and cutters about their respective axes. The rotation is divided into incremental steps. At each step the cutting operation takes place. The incremental step determines how smooth the generated worm and gear surfaces are. As a result there is a difference between the I-DEAS generated surface and the manufactured surface. Because of the computer storage limitation, the steps can not be chosen too small.

The structure of the graphics program is shown by the flow chart in Fig. 5.1, which consists of the following steps:

1. Input wormgear dimensions and parameters which are needed in its manufacture.

2. Input the incremental stepsizes ('dtw' for generating the worm and 'dtg' for generating the gear) and the total number of steps in the cutting processes ('wn' for generating the worm and 'gn' for generating the gear).

3. Construct the grinding wheels for generating the worm. Four are constructed and jointed in sequence to reduce the CPU time, i.e. four coils of the worm thread are created for one revolution of the worm
blank (dtw*wn=360 deg).

4. Create the worm and gear blanks.

5. Create the worm. When the process is completed, the created worm is saved under the name 'worm'.

6. Create the gear. When the process is completed, the created gear is saved under the name 'gear'.

7. Display the created worm and gear.

The program is listed in Appendix 1. The program works interactively and ends at the object modeling screen. This allows the user to see the created objects again before exiting I-DEAS.

The procedure to execute the program is as follows: (1) Enter I-DEAS and come to the construction geometry task screen. (2) Run program in the construction geometry screen. More detailed instructions are given in Appendix 1.

Because the worm thread of the double-enveloping wormgear is asymmetric about its center, the number of steps needed to create the whole worm is larger than that for creating other machine parts, such as the spur gear or the cylindrical worm. More coils of the worm thread requires more steps. Every step increases the number of surfaces on the worm blank. The larger the number of surfaces, the more CPU time and computer disc space are needed to calculate and save the data. Once the capacity of the computer or the I-DEAS space is full, the screen stops, or the program crashes. This is why there are only four coils of worm thread shown in Fig. 2.7. The program was executed on an IBM 5080 terminal at SUNY-Binghamton.

From the above description it can be seen that (1) the computer simulation is helpful to visualize the selected wormgear, and (2) the computer simulation can not replace the analysis, because the numerical data of the
generated surfaces are not accurate enough for the projected lubrication analysis.

5.2 Analyses Program

5.2.1 Overall view of the Program

The analyses program is named NASA.FOR and is written in FORTRAN. This program consists of a main program and several subroutines. The main program is the driver program of all the subroutines and may be tailored and adapted to the final design package. The subroutines fall in two categories: those driven by the main program (DIMEN, WORM, WORML, GEAR, CONTACT, PITCH and LOAD) and those driven by other subroutines (EQW, EQWL, EQG, FIRST, SECOND, WORMT, ROOT and NORMAL). Most of the logical judgements are done in the subroutines called by the main program. The evaluations of analytical expressions are done in the subroutines called by other subroutines.

The structure of the analyses program is shown by the flow chart in Fig. 5.2, which consists of the following steps:
1. Calculate the wormgear dimensions (DIMEN).
2. Calculate the coordinates in $S_1$ of the worm surface and the curvatures of the worm surface (WORM and WORML).
3. Calculate the coordinates in $S_2$ of the points on the gear surface along the contact line and the curvatures of the gear surface (GEAR).
4. Calculate the contact lines in $S_n$ on every pair of the meshing teeth for a given position (CONTACT).
5. Calculate the coordinates in $S_3$ and the kinematic properties of the worm center points for a given position (PITCH).
6. Calculate the load distribution on the meshing teeth for a given position (LOAD).
Input of this program, under the 'parameter' command, are at the beginning of the main program and are explained there. In the main program the given position $\theta^*$ is named 'tn2', which should be the input for the subroutines CONTACT, PITCH and LOAD.

Input to a subroutine are explained at the beginning of the subroutine. The output to be used by other subroutines are described in the subroutine statements. The output not to be used by other subroutines are stored in data files (DIMEN, WORM, WORML, GEAR, CONTG, PITCH and LOAD).

Error flags would be shown on the screen such as 'dimension is not big enough', 'the stepsize is too large', 'the worm surface intersects the edge curve', etc.

5.2.2 Detailed Description of the Program

Main Program

The input under the 'parameters' command include the center distances $a_0$ and $a$, the speed ratios $i_0$ and $i$, the number of worm thread $N_t$, the number of meshing teeth $N_p$, the worm pitch diameter $d_1$, the apex angle $\beta$, the rotation speed of the worm, the output power of the wormgear, and the friction coefficient at the contact.

The step sizes and the dimension arrays are input under the 'data' command. The input step sizes are: 'df' (for the running parameter $\psi_1$), 'dt' (for the running parameter $\theta_2$), 'dx' (in the x axis), and 'dz' (in the z axis). There are four one-dimensional common arrays with dimension $nn=600$.

The given position $\theta^*$ is named 'tn2'. The wormgear position of $\theta^*=0$ is shown in Fig. 3.6. 'tn2' is not to be larger than the angle between two
neighboring gear teeth. The two angles 't20' and 't21' specify the interval of the running parameter \( \theta_2 \), which is chosen to be slightly larger than the meshing zone.

The convergence criteria for root search are specified by 'ep' (for angles) and 'epp' (for distances). A logical judgement for the number of meshing teeth is set up in the main program. This is because the actual number of meshing teeth obtained from subroutines CONTACT and PITCH should agree, otherwise the results cannot be used in the subroutine LOAD. A flow chart for the main program is shown in Fig. 5.3.

Subroutine DIMEN

This subroutine must be run before any other subroutine, because it provides the wormgear parameters and dimensions. The formulas used for the calculation can be found in AGMA (1965), Dudley (1984), and Shigley and Mitchell (1983).

The meshing zone is specified by 'start' and 'end', which are the angles where a gear tooth runs into and out of the meshing zone respectively. 'f20' and 'f21' are the two angles specifying the interval of the running parameter \( \psi_2 \). 'tao' is the angle between two neighboring gear teeth. The half gear face angle, named 'dlt', is used to specify the \( \psi_1 \) interval in the calculation of the gear surface. 'reps' is an angle for determining the position of the original generating plane that grinds the other side of the worm thread, Fig. 5.4.

Subroutine WORM and WORML

These subroutines implement the analysis described in Sections 3.3 and 3.5. The two sides of the worm thread are calculated by WORM and WORML,
respectively.

Three logical judgements are set up. The first checks if the worm surface intersects the edge curve. The second checks which branch of the envelope surface is the real worm surface, Fig. 3.2. The third examines the tooth domain.

Subroutines EQW and EQWL are called by subroutines WORM and WORML, respectively. A flow chart for WORM (and WORML) is shown in Fig. 5.5.

Subroutine GEAR

This subroutine implements the analysis described in Sections 3.4 and 3.5. Calculations for both the conventional form and the modified form are provided.

A $\theta$ value, named 't1', is first given. Then $\psi$, named 'f1', is varied near the $\theta$ value to calculate the coordinates of the points on the gear surface. This is because if $\psi$ is far away from $\theta$, it is also off the tooth domain. The range of $\psi$ variation is twice the gear face angle. This range is much less than the range of the running parameter $\psi$ in the subroutine WORM. Hence, the stepsize $\Delta\psi_1$, 'df', used here is a tenth of the stepsize given in the main program.

The logical judgements set up in this subroutine consist of the tooth domain examinations and sorting out the fictitious points. A flow chart for GEAR is shown in Fig. 5.6.

Subroutine CONTACT

This subroutine implements the analysis of Section 3.6.1. Calculations
for both the conventional form and the modified form are provided. At a given position 'tn2', the contact lines on all the meshing teeth are calculated. The angles 'start' and 'end' are used in this subroutine to find out how many pairs of meshing teeth actually exist in the meshing zone.

Similar to the case of GEAR, $\psi$ is varied near the $\theta$ value. To obtain accurately the length of the contact lines, as many points are needed as is practicable. Hence the stepsizes of $\psi$ and $\zeta$ are chosen to be $df/100$ and $dz/10$, where 'df' and 'dz' are the input stepsizes from the main program.

Four temporary one-dimensional arrays are used to save the coordinates of the points on the same contact line. 'nn' gives the dimension of these arrays, which is controlled by the main program. If the saved points exceed 'nn', an error flag would appear on the computer screen. The number 'nc' (output name 'ncc') records how many pairs of meshing teeth exist in the meshing zone for the given position, which should be the same as that obtained from the subroutine PITCH.

Output of the contact line length is stored in the array 'CONTC', which has dimension $N_p$ and is the input of the subroutine LOAD. A flow chart for CONTACT is shown in Fig. 5.7.

Subroutine PITCH

This subroutine implements the analysis of Sections 3.4 and 3.6.2. At the given position 'tn2' (named 'fn2' here), the worm center point on each pair of the meshing teeth are obtained by first finding the points on the worm profile in the mid-plane, i.e. $y_j = 0$, then sorting out the center point on the profile by using the subroutine ROOT. The angles 'start' and 'end' are used to check if a worm coil falls in the meshing zone.
To obtain accurately the center point coordinates while still saving the CPU time, the stepsize is chosen to be \( df/100 \), where \( 'df' \) is the input stepsize from the main program. The range of \( \psi \) variation is 90 deg around the given position \( 'fn2' \). The index \( mw = 1 \) would mean that the worm profile intersects the edge curve. The coordinates of the center points obtained are in \( S_j \). These coordinates and the normal directions at the center points are stored in one-dimensional arrays with the maximum dimension \( N_p \), and are input for the subroutine LOAD.

The number \( 'nc' \) (output name \( 'ncp' \)) records how many pairs of meshing teeth exist in the meshing zone for the given position, which should be the same as that obtained from the subroutine CONTACT. A flow chart for PITCH is shown in Fig. 5.8.

Subroutine LOAD

This subroutine implements the analysis of Sections 4. Before this subroutine is run, the subroutines PITCH and CONTACT must be run so as to provide the input for this subroutine.

Subroutine ROOT

This subroutine searches the root in a given interval where only one root exists. The root searching is based on the idea of bisection (Press, 1986), but three sections are used instead to reduce the CPU time. Here the root is a point at which the radius of the worm is equal to the radius of the worm pitch cylinder at the same axial position. The criteria used to search the center point are the difference of the two radii or the angles between the two grid points.
Summary

Based on the consideration of achieving better contact properties (longer contact lines and a larger angle between the contact line and the relative sliding velocity), a double-enveloping wormgear was selected. The worm was generated by a disk grinding wheel. The center distance and the speed ratio were respectively the same in the first and the second enveloping processes (conventional form).

Based on the suggested speed ratio and center distance for helicopter applications, the other dimensions and geometric parameters of the wormgear were laid out basically in accordance with the AGMA gear standards. However, some of these quantities were modified as a result of compromising between conflicting requirements. It is felt that for a novel application like ours that is not covered by past experiences, some deviations from the common practice would be inevitable.

3-D graphics representation of the selected wormgear was developed using the I-DEAS software. The software works like a cutting tool generating a surface from a blank, but in discrete steps. Because of the computer storage limitation, the steps could not be made too small. As a result, the numerical data of the generated surfaces were not accurate enough for the projected lubrication analysis. However, the graphics representation does convey a clear notion of the proportion of the selected wormgear.

The kinematics of meshing was analyzed with the method of enveloping. The method is simple and elegant, and can be used to construct conjugate surfaces in other contacting mechanisms as well. Using this method the shape of the worm and gear surfaces, the contact lines, the normal directions, the relative normal curvatures, and the relative sliding velocities at the contact points were obtained. With these results the clearance function between the
meshing surfaces can also be obtained, which is needed for the projected lubrication analysis.

Based on the contact line pattern revealed by the kinematics analysis, a hypothesis of load sharing was made. Using this hypothesis the variation of the normal contact force acting on a gear tooth as it ran through the meshing zone, as well as the distribution of the normal forces on all the meshing gear teeth at any instant, were obtained. This information is needed in the projected lubrication analysis. The required torque output of the worm was likewise obtained. The results clearly demonstrated the advantage and need of implementing effective lubrication at the contact.

A computer program, written in FORTRAN, was developed to carry out the computations. The structure of the computer program and its usage are described in detail.
References


Kreyszig, E., 1959, Differential Geometry, University of Toronto Press.


Table 1 Dimensions of the Selected Wormgear

<table>
<thead>
<tr>
<th>TERM</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>FORMULA BASED</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed ratio</td>
<td>i</td>
<td>110</td>
<td></td>
<td>given</td>
</tr>
<tr>
<td>normal pressure angle</td>
<td>α_n</td>
<td>20°</td>
<td></td>
<td>Dudley, 1984 p. 3.71</td>
</tr>
<tr>
<td>center distance</td>
<td>a</td>
<td>880 mm</td>
<td></td>
<td>selected</td>
</tr>
<tr>
<td>number of worm threads</td>
<td>N_1</td>
<td>1</td>
<td></td>
<td>selected</td>
</tr>
<tr>
<td>number of gear teeth</td>
<td>N_2</td>
<td>110</td>
<td>N_2 = N_1*i</td>
<td></td>
</tr>
<tr>
<td>pitch diameter of worm</td>
<td>d_1</td>
<td>300 mm</td>
<td>d_1 = a^{0.875/k_3} (E) (k_3 = 1.7-2.2)</td>
<td>Dudley, 1984 p. 3.70</td>
</tr>
<tr>
<td>pitch diameter of gear</td>
<td>d_2</td>
<td>1460 mm</td>
<td>d_2 = 2*a - d_1</td>
<td></td>
</tr>
<tr>
<td>axial pitch</td>
<td>p_x</td>
<td>41.6975 mm</td>
<td>p_x = \pi*d_2/N_2</td>
<td>Dudley, 1984 Table 3.32</td>
</tr>
<tr>
<td>worm lead angle</td>
<td>γ</td>
<td>2.533 deg</td>
<td>\tan γ = p_x*N_1/π/d_1</td>
<td>Dudley, 1984 Eq. 3.38</td>
</tr>
<tr>
<td>normal circular pitch</td>
<td>p_o</td>
<td>41.6568 mm</td>
<td>p_o = p_x*cos γ</td>
<td>Dudley, 1984 Table 3.32</td>
</tr>
<tr>
<td>axial pressure angle</td>
<td>α_x</td>
<td>20.018 deg</td>
<td>\tan α_x = tanα_n/cos γ</td>
<td>Dudley, 1984 Eq. 3.40</td>
</tr>
<tr>
<td></td>
<td>Δα</td>
<td>0.818 deg</td>
<td>Δα = p_x/2/d_2</td>
<td>Dudley, 1984 p. 3.72</td>
</tr>
<tr>
<td>base circle diameter</td>
<td>d_b</td>
<td>519.3182 mm</td>
<td>d_b = d_2*sin(α_x + Δα)</td>
<td>Dudley, 1984 Eq. 3.41</td>
</tr>
<tr>
<td>module</td>
<td>x_m</td>
<td>13.2727 mm/ tooth</td>
<td>x_m = p_x/π</td>
<td>Dudley, 1984 Table 3.32</td>
</tr>
<tr>
<td>number of meshing teeth</td>
<td>N_p</td>
<td>10</td>
<td></td>
<td>selected</td>
</tr>
<tr>
<td>half angle of meshing</td>
<td>ψ_a</td>
<td>15.627 deg</td>
<td>ψ_a = \pi*(N_p*0.45)/N_2</td>
<td>Shen, 1983 Table 10.1</td>
</tr>
<tr>
<td>start angle of meshing</td>
<td>ψ_f</td>
<td>5.209 deg</td>
<td>ψ_f = \sin^{-1}(d_b/d_2) - ψ_a</td>
<td>Shen, 1983 Table 10.1</td>
</tr>
<tr>
<td>thickness of worm thread</td>
<td>S_1</td>
<td>18.7639 mm</td>
<td>S_1 = 0.45*p_x</td>
<td>Dudley, 1984 p. 3.73</td>
</tr>
<tr>
<td>thickness of gear tooth</td>
<td>S_2</td>
<td>22.9336 mm</td>
<td>S_2 = 0.55*p_x</td>
<td>Dudley, 1984 P. 3.73</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Formula</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------</td>
<td>----------------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>whole depth</td>
<td>$b_t$</td>
<td>20.8284 mm</td>
<td>Dudley, 1984 Table 3.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_t = 0.5 \times p_n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>working depth</td>
<td>$b_k$</td>
<td>18.7455 mm</td>
<td>Dudley, 1984 Table 3.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_k = 0.45 \times p_n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addendum</td>
<td>$b_a$</td>
<td>9.3728 mm</td>
<td>Dudley, 1984 Table 3.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_a = 0.225 \times p_n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>clearance</td>
<td>$c$</td>
<td>2.0829 mm</td>
<td>Shen, 1983 Table 10.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$c = b_t - b_k$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>throat diameter of worm*</td>
<td>$d_{a1}$</td>
<td>318.7455 mm</td>
<td>Dudley, 1984 Eq. 3.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{a1} = d_1 + 2 \times b_a$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>root diameter of worm*</td>
<td>$d_{r1}$</td>
<td>277.0888 mm</td>
<td>Dudley, 1984 Table 3.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{r1} = d_{a1} - 2 \times b_t$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>throat diameter of gear</td>
<td>$d_{a2}$</td>
<td>1478.7455 mm</td>
<td>Dudley, 1984 Table 3.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{a2} = d_{a2} + 2 \times b_a$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>root diameter of gear</td>
<td>$d_{r2}$</td>
<td>1437.0888 mm</td>
<td>Shen, 1983 Table 10.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{r2} = d_{a2} - 2 \times b_t$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>face width of worm</td>
<td>$b_1$</td>
<td>393 mm</td>
<td>Shen, 1983 Table 10.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_1 = d_2 \times \sin \psi_a$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>face width of gear</td>
<td>$b_2$</td>
<td>249 mm</td>
<td>Shen, 1983 Table 10.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2 = (0.9 - 1.0) \times d_{r1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apex angle of generating plane</td>
<td>$\beta$</td>
<td>5 deg</td>
<td>selected</td>
<td></td>
</tr>
</tbody>
</table>

* At the center of worm.
Figures

Fig. 2.1 Single-enveloping wormgear (cylindrical wormgear)
Fig. 2.2 Double-enveloping wormgear
Fig. 2.3 Worm ground by a disc grinding wheel
Fig. 2.4 Worm ground by a cone grinding wheel
Fig. 2.5 Double-enveloping wormgear layout and nomenclature
Fig. 2.6 Selected wormgear
Fig. 2.7 Selected worm
Fig. 2.8 Selected gear
Fig. 2.9 Close-up view of the selected gear teeth
Fig. 3.1 Coordinate systems
Fig. 3.2 Envelope surface (worm) and its edge curve
Fig. 3.3 Contact lines on a gear tooth of the modified form
Fig. 3.4 Contact lines on a gear tooth of the conventional form
Fig. 3.5 Worm and gear tooth domains
Fig. 3.6 Wormgear position at $\theta^* = 0$
Fig. 3.7 Contact lines on the meshing gear teeth
Fig. 3.8 Length of contact lines on the meshing teeth
   (a) position of $\theta^* = 0$ deg
   (b) position of $\theta^* = 1$ deg
   (c) position of $\theta^* = 2$ deg
Fig. 3.9 Variation of contact line length on a gear tooth
Fig. 3.10 Radial position of the edge curve
Fig. 4.1 Normal force and its components
Fig. 4.2 Friction force and its components
Fig. 4.3 Normal forces on the gear teeth
   (a) position of $\theta^* = 0$ deg
   (b) position of $\theta^* = 1$ deg
   (c) position of $\theta^* = 2$ deg
Fig. 4.4 Variation of normal force on a gear tooth
Fig. 4.5 Tangential forces on the worm
   (a) position of $\theta^* = 0$ deg
   (b) position of $\theta^* = 1$ deg
   (c) position of $\theta^* = 2$ deg
Fig. 5.1 Structure of graphics program
Fig. 5.2  Structure of analyses program
Fig. 5.3  Flow chart of main program
Fig. 5.4  Two sides of the worm thread
Fig. 5.5  Flow chart of subroutine WORM (also WORML)
Fig. 5.6  Flow chart of subroutine GEAR
Fig. 5.7  Flow chart of subroutine CONTACT
Fig. 5.8  Flow chart of subroutine PITCH
Fig. 2.1 Single-enveloping wormgear (cylindrical wormgear)
Fig. 2.2 Double-enveloping wormgear
Fig. 2.3 Worm ground by a disc grinding wheel
Fig. 2.4 Worm ground by a cone grinding wheel
Fig. 2.5 Double-enveloping wormgear layout and nomenclature
Fig. 2.6 Selected wormgear
Fig. 2.7  Selected worm
Fig. 2.8 Selected gear
Fig. 2.9 Close-up view of the selected gear teeth
Fig. 3.1 Coordinate systems
Fig. 3.2 Envelope surface (worm) and its edge curve
Fig. 3.3 Contact lines on a gear tooth of the modified form
Fig. 3.4 Contact lines on a gear tooth of the conventional form
Fig. 3.6 Wormgear position at $\theta^* = 0$
Fig. 3.7  Contact lines on the meshing gear teeth (a)
Fig. 3.7 Contact lines on the meshing gear teeth (b)
Fig. 3.7 Contact lines on the meshing gear teeth (c)
Fig. 3.7  Contact lines on the meshing gear teeth (d)
Fig. 3.7 Contact lines on the meshing gear teeth (e)
Fig. 3.7  Contact lines on the meshing gear teeth (f)
Fig. 3.8 Length of contact lines on the meshing teeth
(a) position of $\theta^* = 0$ deg
position $\theta^* = 1$ (deg)

Fig. 3.8  Length of contact lines on the meshing teeth
(b) position of $\theta^* = 1$ deg
position $\theta^* = 2$ (deg)

Fig. 3.8 Length of contact lines on the meshing teeth
(c) position of $\theta^* = 2$ deg
Fig. 3.9 Variation of contact line length on a gear tooth
Fig. 3.10 Radial position of the edge curve
Fig. 4.1 Normal force and its components
Fig. 4.2 Friction force and its components
position $\theta^* = 0$ (deg)

- * --- * with friction
- o --- o --- without friction

**Tooth Position $\theta_t$ (deg)**

**Fig. 4.3** Normal forces on the gear teeth
(a) position of $\theta^* = 0$ deg
position $\theta^* = 1$ (deg)

Fig. 4.3 Normal forces on the gear teeth
(b) position of $\theta^*=1$ deg
position $\theta^* = 2$ (deg)

- with friction
- without friction

Fig. 4.3 Normal forces on the gear teeth
(c) position of $\theta^* = 2$ deg
Fig. 4.4 Variation of normal force on a gear tooth
position $\theta^* = 0 \text{ (deg)}$

- **** with friction
- ooo o without friction

Fig. 4.5 Tangential forces on the worm
(a) position of $\theta^* = 0 \text{ deg}$
Fig. 4.5  Tangential forces on the worm 
(b)  position of $\theta^* = 1$ deg
Fig. 4.5  Tangential forces on the worm
(c) position of $\theta^* = 2$ deg
Fig. 5.1 Structure of graphics program
Fig. 5.2 Structure of analyses program
Input

Call DIMEN
   calculate the wormgear dimension

Call WORM and WORML
   calculate the coordinates of the worm surface

Call GEAR
   calculate the coordinates of the gear surface

Call CONTACT
   calculate the coordinates and the length of the contact lines.

Call PITCH
   calculate the coordinates of the center points of the worm surface.

$N_p = N_p^*$?

no

Pause

Call LOAD
   calculate the normal load and the tangential component.

Stop

Fig. 5.3 Flow chart of main program
Fig. 5.4  Two sides of the worm thread
Fig. 5.5 Flow chart of subroutine WORM (also WORML)
GEAR

\[ m = \frac{(t20-t21)}{dt} \]
\[ n = 4.0*\frac{dlt}{df} \]
\[ l = b2/dz \]

\[ j = 1, m \]

\[ j = 1, n \]

Call SECOND

Is the point fictitious?

Is the point in the tooth domain?

save the point coordinates

\[ j > n ? \]

\[ a_0 = a, i_0 = i ? \]

\[ k = 1, l \]

\[ \psi = \theta \]

Call FIRST

Is the point fictitious?

Is the point in the tooth domain?

save the point coordinates

\[ k > 1 ? \]

\[ i > m ? \]

Return

Fig. 5.6 Flow chart of subroutine GEAR
Fig. 5.7 Flow chart of subroutine CONTACT
Fig. 5.7 (continued)
Fig. 5.8  Flow chart of subroutine PITCH
Call NORMAL
calculate normal direction
and relative velocity.

\[ \text{nc} = \text{nc} + 1 \]

\[ \text{i > m ?} \]

\text{no} \quad D

\text{yes}

Return

Fig. 5.8 (continued)
Appendix 1 Graphics Program (NASA.PRG)

The procedure to execute 'NASA.PRG'

(1) From the SUNY-Binghamton operating system prompt (assumed here to be $):
    $ ready: type 'runideas' <return>
    $ jump to IBM screen (use Alt-jump)
    $ user i.d.: type 'ideas2'
    $ password: type 'no'
    $ jump back to BINGVMB screen type 'none' (use Alt-jump)
    $ disk space: type a number such as '15000'
    (waiting for temporary disk to be created)
    $ user i.d.: type your own i.d. number and read password
    $ terminal type: type 'i5080'
    $ jump screen (use Alt-jump)
    (waiting for I-DEAS screen to appear)

(2) Under the I-DEAS screen, from the I-DEAS operating prompt (assumed here to be #):
    Enter model file name (none)# <return>
    New model file? (yes)# <return>
    Enter model description (none)# <return>
    Enter system of units (SI)# <return>
    Enter operation mode (Normal_Operation)# <return>

(3) Then choose which "family" of I-DEAS you want. Notice that choices are in the form of a 'menu'. So you can pick a choice with the mouse or type under the I-DEAS operating prompt (assumed here to be #):
    Select menu# SO <return>
    (waiting for Solid Modeling Screen to appear)
    Select menu# CG <return>
Select menu# MF <return>
Select menu# PR <return>
Select menu# R <return>
File name# NASA <return>

Then the program is run interactively. All the results will be seen from the screen.

(4) The created object can be observed again before exiting I-DEAS system. The procedure to observe the created object is:

Select menu# MA <return>
Select menu# G <return>
Enter object name# type object name (worm or gear)
(the created object appears on the screen)

Once the created object is obtained, choose from the menu options such as orient, view, display and AU (auto scale) to observe the object.

(5) When the observation is completed, put the created object away from the screen by:

Select menu# / <return>
Select menu# MA <return>
Select menu# PU <return>

(6) When all the observations are completed, if needed the created object can be saved by:

Select menu# / <return>
Select menu# MA <return>
Select menu# WR <return>
Select menu# U <return>
Enter the file name# type the name you choose <return>
(7) Exit I-DEAS screen by:
   Select menu# / <return>
   Select menu# type 'exit' <return>
Program Listing

C:  This is an idea program to construct a worm gear set by the
C:  principle of envelop. The constructing process is same as
C:  the manufacturing process exactly. To cut four worm threads
C:  the 'wn' must be equal to 360/dtw and 'gn' must be 4*360/dtg.
C:
C:  The basic parameters provided are:
C:  a0------center distance of cutting
C:  db------base circle diameter
C:  b1------face width of worm
C:  b2------face width of gear
C:  d2------pitch diameter of gear
C:  da2------thread diameter of gear
C:  df2------root diameter of gear
C:  df1------root diameter of worm
C:  ratio--speed ratio of worm gear
C:  n2------number of gear teeth
C:  dt1----angle of worm turning in each cut
C:  dt2----angle of gear turning in each cut
C:  wn------times of worm cutting
C:  gn------times of gear cutting
C:  bta------apex angle of generating plane
C:  psif----angle of meshing start
C:  reps--angle of two cutting surfaces

C:  ** 'reps' is calculated by the following formular:
C:  reps=180-(2*[90-arcsin(db/d2)]+s2/r2), here
C:  s2=0.55*pi*d2/n2 i.e. s2/r2=1.1*180/n2 (deg)

C:  "/
C:  basic parameters input =

K:  #echo list
K:  #a0=880.0
K:  #n2=110
K:  #ratio=110
K:  #db=519.3182
K:  #bta=5.00
K:  #d1=300.0
K:  #d2=1460.0
K:  #da1=318.7455
K:  #da2=1478.7455
K:  #df1=277.0888
K:  #df2=1437.0888
K:  #b1=393.0
K:  #b2=249.0
K:  #rb=db/2.0
K:  #r1=d1/2.0
K:  #r2=d2/2.0
K:  #ral=da1/2.0
K:  #ra2=da2/2.0
K:  #rf1=df1/2.0
K:  #rf2=df2/2.0
K:  #dh=2.0829

93
K : #psif=5.209
K : #reps=39.8724
K : #aa=a0-sqrt((rf2+dh)**2-b1*b1/4.0)
K : #bb=sqrt(ra2*ra2-rb*rb)
K : #cc=a0-sqrt(rl*rl-b2*b2/4.0)
K : #dd=a0-sqrt(ra2*ra2-b1*b1/4.0)
K : #ee=a0-ra2
K : #dtw=8.00
K : #dtg=15.0
K : #wn=5
K : #gn=3

- creat 2-D profile of worm blank -

K : "CR"
K : "PR"
K : "K"
K : 0,r2
K : a0-cc,r2
K : "F"
K : a0-cc,b1/2.0
K : "F"
K : aa,b1/2.0
K : "F"
K : a0-rf2-dh,0
K : aa,-b1/2.0
K : "F"
K : a0-cc,-b1/2.0
K : "F"
K : a0-cc,-r2
K : "F"
K : 0,-r2
K : "F"
K : "C"
K : "/
K : "MA"
K : "PR"
K : "STO"
K : pl
K : "/"
K : "DE"
K : "F"
K : "Y"
K : "/"

- creat 2-D profile of worm root blank -

K : "CR"
K : "PR"
K : "K"
K : 0,b1/2
K : dd,b1/2
K : "F"
K : ee,0
K : dd,-b1/2.0
K : "F"
K : 0,-b1/2.0
K : "F"
= create 3-D gear blank =

= create a 3-D worm cutter =
K : 0,0,rb/tan(bta)
K : "P"
K : "T"
K : "K"
K : rb*cos(reps),-rb*sin(reps),0
K : rb/cos(reps),0,0
K : 0,0,rb/tan(bta)
K : "N"
K : "/
K : "OR"
K : "MO"
K : "D"
K : 0,1,0
K : a0
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,psif
K : "/
K : "MA"
K : "STO"
K : c2

= create four connected 3-D worm cutters =

C C C C
C : "/
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,360/n2
K : "/
K : "MA"
K : "G"
K : c2
K : "/
K : "CO"
K : "J"
K : "/
K : "MA"
K : "STO"
K : c2
K : "/
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,2*360/n2
K : "/
K : "MA"
K : "G"
K : c2
K : "/
K : "CO"
K : "J"
K : "P"
= creat 3-D worm blank =
K : "STO"
K : w1
K : "G"
K : c2

----------------------
= 3-D worm cutting =
----------------------
K : "/"
K : "CO"
K : "CU"
K : "/"
K : #time=1
K : #start:
K : t1=time*dtw
K : t2=t1/ratio
K : "OR"
K : "R"
K : -dtw,0,0
K : "/"
K : "MA"
K : "G"
K : c2
K : "/"
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,t2
K : "/"
K : "CO"
K : "CU"
K : "/"
K : #time=time+1
K : #if(time lt wn) then goto start
K : "/"
K : "OR"
K : "R"
K : "(time-1)*dt1,0,0"
K : "/"
K : "MA"
K : "STO"
K : worm
K : "MO"
K : "DE"
K : "Y"
K : "CR"
K : "REV"
K : p2
K : "/"
K : "OR"
K : "R"
3-D gear cutting
## view worm

```

```

## view gear

```

```
Appendix 2 Listing of the Analyses Program (NASA.FOR)

This is a part of a developing software package which is used to analyse the double-enveloping wormgear fluid film lubrication. This part includes:

1. calculation of the wormgear dimensions;
2. calculation of the coordinates of worm and gear surfaces in S1 and S2 systems;
3. calculation of the contact lines on all pairs of meshing teeth for a given position;
4. calculation of the coordinates and relative velocity at the worm pitch points on all pairs of meshing teeth for a given position;
5. calculation of load sharing among all meshing teeth for a given position.

To run this program the input parameters are:

- a0: center distance of the first enveloping (mm)
- a2: center distance of wormgear (mm)
- bta: apex angle of the generating plane (deg)
- d1: pitch diameter of worm (mm)
- df1: S1 running step size in the first enveloping (deg)
- dt2: S2 running step size in the second enveloping (deg)
- dx: interval along x axis (mm)
- dz: interval along z axis (mm)
- ep, epp: convergence criterion for root searching
- nl: number of worm thread
- np: number of meshing teeth
- nn: number of dimension
- rf: normal pressure angle (deg)
- ratio: speed ratio of the first enveloping
- ratio2: speed ratio of wormgear
- power: output power from gear shaft (kw)
- rpm: input speed from worm shaft (rpm)
- tn2: given gear position
- uf: friction coefficient

The variables in the main program are:

- t20: running angle of the second enveloping start (rad)
- t21: running angle of the second enveloping end (rad)

Subroutines included:

1. Called by MAIN Program

   - DIMEN: develop the dimensions of wormgear.
   - WORM: develop the worm surface in S1. Here the worm is ground by a disc grinding wheel.
   - WORML: develop the worm surface on the other side, in S1.
   - GEAR: develop the gear surface in S2.
   - CONTACT: calculate the length of contact lines on each pair of meshing teeth at the given position.
   - PITCH: finding the pitch points on the worm surface, the
normal directions and relative velocities on each pair of meshing teeth for a given position. 
calculate the normal force on each pair of meshing teeth and the torque sharing among the worm meshing teeth.

(2) Called by Subroutine

EQG the gear surface equation in S2

EQWL the worm surface equation in S1
FIRST calculation of the first enveloping surface
NORMAL calculation of the properties of the worm pitch points
ROOT finding the pitch point
SECOND calculation of the second enveloping surface
WORMT calculation of the worm profile on mid-plane in Sj

Outputs of the program are written in the data files under the names of the subroutines. Units used in the output data are:
rad for angle
mm for length
newton for force.

PROGRAM MAIN

implicit double precision(a-h,o-z)
parameter(bta=5.0,a0=880.00,a2=880.00,ratio=110.,ratio2=110.,
& rfn=90, nl=1, np=10, dl=300.0, epp=1.0e-5, ep=1.0e-3,
& power=4000, rpm=14280, uf=0.04, us=0.0)
dimension contc(np), tp2(np), ep2(np), pxj(np), pzj(np),
& xpn(np), ypn(np), zpn(np), vx(np), vy(np), vz(np),
& wtor(np), wtorf(np), fnr(np), fonn(np),
& eexa(nn), exbb(nn), ecx(nn), exdd(nn)
data df1/10.0/, dt2/1.0/, dx/5.0/, dz/5.0/, tn2/0.0/

c open(1, file=''/dien dat')
open(2, file=''/worm dat')
open(3, file=''/wormal dat')
open(4, file=''/gear dat')
open(5, file=''/contc dat')
open(7, file=''/normal dat')
open(8, file=''/pitch dat')
open(9, file=''/load dat')

c pi=4.d0*datan(1.d0)
bta=bta*pi/180.d0
tb=datan(bt)
dt=dt*pi/180.d0
df=df*pi/180.d0
rfnr=rfn*pi/180.d0

c { calculation of the dimensions }

call dimen(a0,a2,ratio2,nl,np,dl,rfnr,bt,
& b1,b2,cc,ddd,dlf,f20,f21,
& rb,rl,r2,ra1,ra2,rf1,rf2,rmax2,
& reps,tta,cmnx,cmn,star,end)

c print*, ' DIMEN done'
calculation of the worm surfaces along the contact line

```fortran
  call worm(a0, a2, ratio, rb, bt, tb,
            f20, f21, b1, ra2, rf2, cc, df, dz)

  call worml(a0, a2, ratio, rb, bt, tb,
             f20, f21, b1, ra2, rf2, cc, df, dz, reps)
  print*, 'WORM and WORML done'
```

calculation of the gear surfaces along contact lines

```fortran
  t20 = f20
  t21 = f21
  call gear(a0, a2, ratio, ratio2, rb, bt, b1, b2, rmax2, cmax, cmin,
            r2, ra2, rf2, r1, ral, rfl, r1, cc, ddd, b1, b2,
            f20, f21, df, t20, t21, dt, dlt, dz)
  print*, 'GEAR done'
```

calculation of the contact line at the given position 'tn2'

```fortran
  call contact(a0, a2, ratio, ratio2, np, rb, bt, dh, ddd,
               r2, ral, rfl, t2, ra2, rf2, rmax2, cmax, cmin, b1, b2,
               f20, f21, df, t20, t21, tt2, tn2, start, end, dlt, dz,
               contc, ncc, nn, exaa, exbb, excc, exdd)
  print*, 'CONTACT done'
```
calculate the characters of pitch points on the worm surface at the given position 'tn2'

```fortran
  call pitch(a0, a2, ratio, ratio2, np, rb, bt, cc, ddd,
             r1, ral, rfl, t2, ra2, rf2, rmax2, cmax, cmin, b1, b2,
             f20, f21, df, t20, t21, tt2, tn2, start, end, dlt, dz,
             nn, exaa, exbb, excc, exdd, ep, epp,
             tp2, fp2, pxj, pzj, xpn, ypn, zpn, vx, vy, vz, ncp)
  print*, 'PITCH done'
```
calculate the load distribution at the given position 'tn2'

```fortran
  if(ncc.ne.ncp) then
    print*, 'ncc.ne.ncp check parameters'
    print*, ncc, ncp
    endif
  call load(a2, ratio2, r2, np, power, rpm, uf, ncc, tp2, pxj,
            pzj, xpn, ypn, zpn, contc, wtor, wtorf, forn, fornf)
  print*, 'LOAD done'
```

stop
end

*************** END of MAIN PROGRAM ***********************

103
Subroutine DIMEN called by MAIN

This subroutine is used to develop the dimensions of a worm gear.

To run this subroutine the input parameters are:

- $a_0$: center distance of the first enveloping (	ext{mm})
- $a_2$: center distance of worm gear (	ext{mm})
- $bta$: apex angle of the generating plane (deg)
- $d1$: pitch diameter of worm (	ext{mm})
- $nl$: number of worm thread
- $nbao$: number of meshing teeth
- $ratio$: speed ratio of worm gear
- $rfn$: normal pressure angle (deg)

The output of the subroutine are:

- $b1$: face width of worm (	ext{mm})
- $b2$: face width of gear (	ext{mm})
- $cmax$: maximum thread radius of the gear hob (	ext{mm})
- $cmin$: minimum root radius of the gear hob (	ext{mm})
- $dh$: clearance (	ext{mm})
- $dlit$: half of gear face angle (rad)
- $dlta$: the change in center distance (mm)
- $end$: the angle of the meshing end (rad)
- $f20$: the angle of the first enveloping beginning (rad)
- $f21$: the angle of the first enveloping ended (rad)
- $rl$: pitch radius of worm (mm)
- $r2$: pitch radius of gear (mm)
- $ral$: throat radius of worm (mm)
- $ra2$: throat radius of gear (mm)
- $rb$: radius of base circle (mm)
- $rf1$: root radius of worm (mm)
- $rf2$: root radius of gear (mm)
- $reps$: the angle of grinding wheel turning to another side (rad)
- $rmax$: out-radius of the gear (mm)
- $start$: the angle of the meshing starting (rad)
- $tao$: angle between two neighboring gear teeth (rad)

The results which are not called by other subroutines are written in the data file under the name of 'dimen.dat'.

```
subroutine dimen(a0,a2,ratio,nl,nbao,d1,rfn,bta,
                   b1,b2,dlta,dlit,f20,f21,
                   rb,rl,r2,ral,ra2,rf1,rf2,rmax,
                   reps,tao,cmax,cmin,start,end)
  implicit double precision(a-h,o-z)
  format(2x,6f12.4)
  c
  c
  pitch diameter and radii
  n2=1dint(nl*ratio)
  d2=2.0*a2-d1
  rl=d1/2.0
  r2=d2/2.0
  c
  circle pitch, thickness, addendum, dedendum and clearance
```
\[ \pi = 4.0 \times \text{atan}(1.0) \]
\[ px = \pi \times d2/n2 \]
\[ s2 = 0.55 \times px \]
\[ sl = px - s2 \]
\[ xm = px/\pi \]
\[ gama = \text{atan}(px \times n1/(\pi \times d1)) \]
\[ rfx = \text{atan}(rfn)/\cos(gama) \]
\[ rfx = \text{atan}(rfx) \]
\[ drf = px/(2 \times d2) \]
\[ \begin{align*}
  pnx &= \pi \times \cos(gama) \\
  ba &= 225 \times pnx \\
  bk &= 0.45 \times pnx \\
  bt &= pnx/2.0 \\
  bf &= bt-ba \\
  dh &= bt-bk \\
\end{align*} \]
\[ \text{diameters and radius} \]
\[ \begin{align*}
  db &= d2 \times \text{dsin}(rfx+drf) \\
  dal &= d1+2.0 \times ba \\
  da2 &= d2+2.0 \times ba \\
  df1 &= d1+2 \times ba-2 \times bt \\
  df2 &= d2+2 \times ba-2 \times bt \\
  rb &= db/2.0 \\
  ral &= dal/2.0 \\
  ra2 &= da2/2.0 \\
  rfl &= df1/2.0 \\
  rf2 &= df2/2.0 \\
\end{align*} \]
\[ \begin{align*}
  rfi &= rfx+drf \\
  tao &= 2 \times \pi \times n2 \\
  psia &= 0.5 \times tao \times (nba - 0.45) \\
  start &= rfi - psia \\
  end &= start+2 \times psia \\
  bl &= \text{idint}(d2 \times \text{dsin}(psia)) \\
  b2 &= \text{idint}(.9 \times df1) \\
\end{align*} \]
\[ \begin{align*}
  \text{rgmax} &= a2 - d\text{sqrt}(r1 \times r1 - b2 \times b2/4.0) \\
  \text{cmax} &= a2 - d\text{sqrt}(rf2 \times rf2 - b1 \times b1/4.0) \\
  \text{cmin} &= a2 - d\text{sqrt}(ra2 \times ra2 - b1 \times b1/4.0) \\
  \text{dlta} &= \text{cmax}+rf2-a2 \\
  \text{reps} &= (2.0 \times rfi+2 \times s1/d2) \\
  f20 &= \text{start}+tao \\
  f21 &= \text{end}+tao \\
  dl1 &= \text{dasin}(df1/d1) \\
\end{align*} \]
\[ \text{table of the wormgear dimensions} \]
\[ \begin{align*}
  \text{write(1,*)} & \quad \text{'Dimensions of selected wormgear'} \\
  \text{write(1,*)} & \quad \text{'a0,a2,rfn,ratio,db'} \\
  \text{write(1,1)} & \quad a0,a2,rfn,ratio,db \\
  \text{write(1,*)} & \quad \text{'p2,s2,s1,bk,bt,cc'} \\
  \text{write(1,1)} & \quad px,s2,s1,bk,bt,dbh \\
  \text{write(1,*)} & \quad \text{'dal,df1,da2,df2,d1,d2'} \\
  \text{write(1,1)} & \quad dal,df1,da2,df2,d1,d2 \\
  \text{write(1,*)} & \quad \text{'ral,rfi,ra2,rf2,r1,r2'} \\
  \text{write(1,1)} & \quad ral,rfi,ra2,rf2,r1,r2 \\
  \text{write(1,*)} & \quad \text{'rfi,rfx,drf,rcrs,lambda,psia'} \\
  \text{write(1,1)} & \quad rfi,rfx,drf,rcrs,lambda,psia \\
  \text{write(1,*)} & \quad \text{'bta,f20,f21,start,end'} \\
\end{align*} \]
write(1,1) bta,f20,f21,start,end
write(1,*) 'dlt,b1,b2'
write(1,1) dlt,b1,b2
return
end

End of Subroutine DIMEN

Subroutine WORM called by MAIN

This subroutine is used to calculate the worm surface in S1, where
worm is ground by a disc. To drive this subroutine, the subroutine
DIMEN must be called first.

To run this subroutine the input parameters are:

- \( a_0 \) = center distance of the first enveloping (mm)
- \( a_2 \) = center distance of wormgear (mm)
- \( b_1 \) = face width of worm (mm)
- \( b_t \) = apex angle of the generating plane (rad)
- \( c_c \) = clearance (mm)
- \( d_f \) = step size of \( f_1 \)
- \( d_z \) = step size of \( z_1 \)
- \( f_{20} \) = the angle of the first enveloping beginning (rad)
- \( f_{21} \) = the angle of the second enveloping ended (rad)
- \( r_a \) = throat radius of gear (mm)
- \( r_{ratio} \) = speed ratio of wormgear
- \( r_b \) = radius of base circle (mm)
- \( r_{f2} \) = root radius of gear (mm)
- \( t_b \) = tan(bt)

The variables in WORM are:

- \( f_1 \) = the running parameter in the first enveloping (rad)
- \( f_2 \) = \( f_1/ratio \)
- \( z_1 \) = the coordinate in worm axis direction

The results which are not called by other subroutines are written
in the data file under the name of 'worm.dat'.

```
subroutine worm(a0,a2,ratio,rb,bt,tb,
& f20,f21,b1,ra2,rf2,cc,df,dz)
  implicit double precision(a-h,o-z)
  format(2x,6f10.4)
Title of WORM.dat file
write(2,*) 'The contact lines on the worm surfaces in S1'
write(2,*) 'f1 | f2 | z1 | x1 | y1 |
& curvel'
write(2,*) '--------------------------'
&----------'
precalculation
fi=f20*ratio
fe=f21*ratio
m=1dnint((fe-fi)/df)+1
```
\[ z_i = -b_1 / 2.0 \]
\[ l = \text{idint}(b_1 / dz) \]

Calculation of the worm surface coordinate in \( S_1 \)

\[ \text{do } 20 \ k = 1, 1 \]
\[ \text{do } 10 \ j = 1, m \]
\[ f_1 = f_1 + (j - 1) \ast df \]
\[ f_2 = f_1 / \text{ratio} \]
\[ \text{call eqw}(f_1, f_2, a_0, \text{ratio}, r_b, t_b, \]
\[ p, q, r, s, p_1, q_1, r_1, s_1, p_2, q_2, r_2, s_2) \]
\[ d_w = p \ast (q_2 \ast r_1 - q_1 \ast r_2) + q \ast (p_1 \ast r_2 - p_2 \ast r_1) + r \ast (p_2 \ast q_1 - p_1 \ast q_2) \]
\[ \text{if}(d_w \text{eq} 0.0) \ \text{goto } 10 \]
\[ f_z = -p \ast (q \ast (q_2 \ast s_1 - q_1 \ast s_2) + q \ast (p_1 \ast s_2 - p_2 \ast s_1) + s \ast (p_2 \ast q_1 - p_1 \ast q_2)) / d_w \]
\[ f_x = -q \ast (r \ast (r_2 \ast s_1 - s_2 \ast r_1) + r \ast (q_1 \ast s_2 - q_2 \ast s_1) + s \ast (q_2 \ast r_1 - r_2 \ast q_1)) / d_w \]
\[ f_y = -r \ast (s \ast (s_1 \ast s_2 \ast r_1) + s \ast (p_1 \ast s_2 - p_2 \ast s_1) + s \ast (p_2 \ast r_1 - r_2 \ast p_1)) / d_w \]
\[ \text{redge} = \text{dsqrt}(f_x \ast f_x + f_y \ast f_y) \]
\[ d_w = p_1 \ast q - p \ast q_1 \]
\[ x = ((q_1 \ast r - q \ast r_1) \ast z - q_1 \ast s_1 + q_1 \ast s) / d_w \]
\[ y = ((p_1 \ast r - r_1 \ast p) \ast z - p_1 \ast s_1 + p \ast s_1) / d_w \]

Judge if the point is in the tooth domain

\[ \text{if}(z \text{lt} f_z) \ \text{goto } 10 \]
\[ r_w = \text{dsqrt}(x \ast x + y \ast y) \]
\[ r_{\text{max}} = a_2 - \text{dsqrt}((r_2 + c_2) \ast 2 \ast z - z^2) \]
\[ r_{\text{min}} = a_2 - \text{dsqrt}((r_2 + c_2) \ast 2 \ast z - z^2) \]
\[ \text{if}(r_w \text{gt} r_{\text{max}} \text{or} r_w \text{lt} r_{\text{min}}) \ \text{goto } 10. \]

Justify the tooth undercut

\[ \text{if}(\text{redge} \text{gt} r_w) \ \text{then} \]
\[ \text{print*,'the worm surface intersects with the edge curve'} \]
\[ \text{print*,x,y,z} \]
\[ \text{endif} \]

Calculation of the worm surface curvature

\[ f_f = p_2 \ast x + q_2 \ast y + r_2 \ast z + s_2 \]
\[ r_r = r_r \]
\[ p_e_l = 1.0 + p \ast p / r_r \]
\[ p_f_l = p \ast q / r_r \]
\[ p_g_l = 1.0 + q \ast q / r_r \]
\[ d_w = f_f \ast r_2 \ast d \text{sqrt}(p \ast p + q \ast q + r \ast r) \]
\[ a_a = p_2 \ast r_1 - r_2 \ast p_1 \]
\[ b_b = q_2 \ast r_1 - r_2 \ast q_1 \]
\[ p_1 = a_a / d_w \]
\[ p_m_l = a_a - b_b / d_w \]
\[ p_n_l = b_b - b_b / d_w \]
\[ c_u_rv_e_l = (p_e_l \ast p_n_l - 2.0 \ast p_f_l \ast p_m_l + p_g_l \ast p_1) / (p_e_l \ast p_g_l - p_f_l \ast p_f_l) \]

\[ \text{write}(2,2) \ f_1,f_2,z,x,y,\text{curvel} \]
10 continue
20 continue
return
end

--- End of Subroutine WORM ---

107
Subroutine EQW called by WORM

This subroutine is used to calculate the coefficients of the worm surface equations.

To run this subroutine the input parameters are:

\[ a_0 \] center distance of the first enveloping (mm)

\[ b_t \] apex angle of the generating plane (rad)

\[ f_1 \] running parameter in the first enveloping (rad)

\[ f_2 \] \( f_1/ratio \)

\[ r_b \] radius of base circle (mm)

\[ t_b \] \( \tan(b_t) \)

The output in EQW are:

\[ p, q, r, s \] coefficients of the first equation in Eqs(19)

\[ p_1, q_1, r_1, s_1 \] coefficients of the second equation in Eqs(19)

\[ p_2, q_2, r_2, s_2 \] coefficients of the third equation in Eqs(19)

subroutine eqw(f1,f2,a0,ratio,rb,tb,
& p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)

implicit double precision(a-h,o-z)

cf2=dcos(f2)
sf2=dsin(f2)
cfl=dcos(f1)
sfl=dsin(f1)

Here is the equation (18)

\[ p = c_1 f_2 + t_b s_2 \]
\[ q = t_b c_1 - s_1 s_2 \]
\[ r = -c_2 \]
\[ s = -r_b - a_0 s_2 \]
\[ p_1 = -t_b c_1 s_1 + r_1 + c_2 \]
\[ q_1 = -t_b s_1 r_1 - c_2 \]
\[ r_1 = s_2 \]
\[ s_1 = a_0 c_2 \]
\[ p_2 = -t_b s_1 r_1 - c_2 \]
\[ q_2 = -t_b s_1 r_1 + c_2 \]
\[ r_2 = c_2 \]

return

END of Subroutine EQW

Subroutine WORML called by MAIN

This subroutine is used to calculate the worm surface in S1, where worm is ground by a disc. To drive this subroutine, the subroutine DIMEN must be called first.

To run this subroutine the input parameters are:
The variables in WORML are:

- f1: the running parameter in the first enveloping (rad)
- f2: f1/ratio
- z1: the coordinate in worm axis direction

The other results which are not called by other subroutines are written in the data file under the name of 'worml.dat'

```
subroutine worml(a0,a2,ratio,rh,rb,tb,
&    f20,f21,b1,ra2,rf2,cc,df,dz,reps)
    implicit double precision(a-h,o-z)
    format(2x,6f10.4)
Title of worml.dat file

    write(3,*) '   The contact lines on the other side worm surfaces'
    write(3,*) '----------------------------------------------'
    write(3,*) ' f1  |  f2  |  z1  |  x1  |  y1  |
&  curve1'
    write(3,*) '---------|---------|--------|--------|--------'
&---------
precalculation
    fi=f20*ratio
    fe=f21*ratio
    m=int((fe-fi)/df)+1
    zi=-b1/2.0
    l=int(b1/dz)
    do 20 i=1,1
        z=zi+(i-1)*dz
        do 10 j=1,m
            f1=fi+(j-1)*df
            f2=f1/ratio
            f2=f2-reps
            call eqwl(f1,f2,a0,ratio,rh,rb,
&    p,q,r,s,pl,q1,rl,s1,p2,q2,r2,s2)
              dw=p*(q2*rl-q1*r2)+q*(p1*r2-p2*rl)+r*(p2*q1-pl*q2)
    if (dw.eq.0) goto 10
```

109
fz=-(p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
fx=-(q*(r2*s1-s2*r1)+r*(q1*s2-q2*s1)+s*(q2*r1-r2*q1))/dw
fy=(p*(r2*s1-s2*r1)+r*(p1*s2-p2*s1)+s*(p2*r1-r2*p1))/dw
redge=dsqrt(fx*fx+fy*fy)

judge if the point is in the tooth domain

if(z.gt.fz) goto 10
dw=p1*q-p*q1
x=-(p1*r-r*q1)*z-q*s1+q1*s)/dw
y=-((pl*r-r*pl)*z-p*s1+p1*s)/dw
rw=dsqrt(x*x+y*y)
rm=2-dsqr((2+cc)**2-z*z)
min=2-2dsqr((2+cc)**2-z*z)
if(rw.gt.rmax.or.rw.lt.rmin) goto 10

c  c judge the tooth undercut

if(redge.gt.rw) then
  print*, 'this point of worm surface is on the edge curve'
  print*, x, y, z
endif

c  c calculation of the worm surface curvatur

ff=p2*x+q2*y+r2*z+s2
rr=r*r
pe1=1.0+p*p/rr
pe2=p2/p*rr
pg1=1.0+q2/rr
bw2=f2*rr*dsqrt(p2*p+q+rr)
ab=p2*r1-r*pl
bb=q*r1-r*q1
pl1=aa+aa/dw
pl1=aa*bb/dw
pl1=bb*bb/dw
curvel=(pe1*pl1-2.0*pe1*pl1+pe1*pl1)/(pe1*pg1-p1*pl1)

c  write(3,3) fl,f2+reps,z,x,y,curvel
10 continue
20 continue
return
end

-------------------------------- End of Subroutine WORML --------------------------------

-------------------------------- Subroutine EQWL called by WORML --------------------------------

This subroutine is used to calculate the coefficients of the worm surface.
To run this subroutine the input parameters are:
  a0  center distance of the first enveloping (mm)
  bt  apex angle of the generating plane (rad)
  f1  running parameter of the first enveloping (rad)
  f2  fl/ratio
  rb  radius of base circle (mm)
  tb  tan(bt)
The output in EQW are:

- \( p, q, r, s \) coefficients of the worm equation
- \( p_1, q_1, r_1, s_1 \) coefficients of the worm equation
- \( p_2, q_2, r_2, s_2 \) coefficients of the worm equation

---

```c
subroutine eqwl(f1,f2,a0,ratio,rb,tb,
 & p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
  implicit double precision(a-h,o-z)
  cf2=dcos(f2)
  sf2=dsin(f2)
  cf1=dcos(f1)
  sf1=dsin(f1)
  p=cf1*sf2+tb*sf1
  q=tb*cf1-sf1*sf2
  r=-cf2
  s=-rb-a0*sf2
  p1=-tb*cf1*ratio-sf1*sf2*ratio+cf1*cf2
  q1=-tb*sf1*ratio-sf1*cf2-cf1*sf2*ratio
  r1=sf2
  s1=-a0*cf2
  p2=-tb*sf1*ratio*ratio-cf1*sf2*ratio*ratio
  & -cf1*sf2-2*sf1*cf2*ratio
  q2=-tb*cf1*ratio*ratio+sf1*sf2*ratio*ratio
  & +sf1*sf2-2*cf1*cf2*ratio
  r2=cf2
  s2=a0*sf2
  return
end
```

---

End of Subroutine EQWL

---

Subroutine GEAR called by MAIN
This subroutine is used to calculate the gear surface in S2, where
the gear hob is ground by a disc. To drive this subroutine, the
subroutine DIMEN must be called first.

To run this subroutine the input parameters are:
- \( a_0 \) center distance of the first enveloping (mm)
- \( a_2 \) center distance of wormgear (mm)
- \( b_1 \) face width of worm (mm)
- \( b_2 \) face width of gear (mm)
- \( b_t \) apex angle of the generating plane (rad)
- \( c \) clearance (mm)
- \( c_{max} \) maximum thread radius of the gear hob (mm)
- \( c_{min} \) minimum root radius of the gear hob (mm)
- \( d_{dd} \) the change in center distance (mm)
- \( d_{lt} \) half of gear face angle (rad)
- \( d_f \) step size of \( f_1 \) (mm)
- \( d_z \) step size of \( z_2 \) (mm)
- \( f_{20} \) the angle of the first enveloping beginning (rad)
- \( f_{21} \) the angle of the second enveloping end (rad)
- \( r_1 \) pitch radius of worm (mm)
- \( r_2 \) pitch radius of gear (mm)
- \( r_{al} \) throat radius of worm (mm)
ra2  throat radius of gear (mm)
ratio speed ratio of wormgear
ratio2 speed ratio of wormgear
rb  radius of base circle (mm)
rf1  root radius of worm (mm)
rf2  root radius of gear (mm)
rmax2 out-radius of the gear (mm)
t20  running angle of the second enveloping start (rad)

t01  running angle of the second enveloping ended (rad)
tb  \tan(bt)

The variables in GEAR are:

t1  the running parameter in the second enveloping (rad)
t2  t1/ratio
z2  the coordinate in gear axis direction

The results which are not called by other subroutines are written in the data file under the name of 'gear.dat'.

-------------------------------------------------------------

subroutine gear(a0,a2,ratio,ratio2,rb,tb,tt,rf2,rbmax2,cmx,cmin,
& ra2,r2,r2,ra1,rf1,rl,cc,ddd,b1,b2,
& f20,f21,dt,t20,t21,dt,dl,t,dt,dl,t)
implicit double precision(a-h,o-z)
format(2x,6f12.4)

Title of Gear.dat file

write(4,*)' This is the point coordinates on gear surface:( when & t1=f1, the point is on the first contact line)'
write(4,*)'-----------------------------------------------------
&---------------------------------------------'
write(4,*)' t1  f1  x2  y2
& z2  r'
write(4,*)'-----------------------------------------------------
&---------------------------------------------'

l=idint(b2/dz)+1
m=idint((t21-t20)/dt)+1
df=dt/10.0
n=idint(4.d0*dt/dt)+1

precalculation

do 10 i=1,m
  t2=t20+(i-1)*dt
  t1=t2*ratio2

calculation of the curved part of the gear surface in S2

do 20 j=1,n
  f1=(t1-2.0*dt)+(j-1)*df1
  f2=f1/ratio
  if(f2.lt.f20.or.f2.gt.f21) goto 20
  call second(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,
& x1,y1,z1,x2,y2,z2,xn,yn,zn,
& curvel,curve2,curve,dw)

judge if the point is in the tooth domain

112
if(dw.eq.0) goto 20
if(curve.lt.0) goto 20
if(z2.gt.b2/2.or.z2.lt.-b2/2) goto 20
rg=dsqrt(x2*x2+y2*y2)
rmmin=(a2+d2d)-dsqrt(cmax**2-z2**2)
rmmax=(a2+d2d)-dsqrt(cmin**2-z2**2)
if (rmmax.gt.rmax2) then
rmax=rmax2
endif
if(rg.gt.rmax.or.rg.lt.rmin) goto 20
write(4,4)t1,f1,x2,y2,z2,rg
continue
if(a0.ne.a2.or.ratio.ne.ratio2) goto 10
calculation of the planar part of the gear surface in S2
(for the case of f1=t1 in the conventional form)
d0 30 k=1,1
z2=-b2/2+(k-1)*dz
f1=t1
f2=t2
if(f2.lt.f20.or.f2.gt.f2l) goto 30
 call first(t2,f2,a0,a2,ratio,ratio2,rb,tt,ttt,z2,
 &x1,y1,z1,y2,x2,xn,yn,zn,
 &curve1,curve2,curve,dw)
c judge if the point is in the tooth domain
if(dw.eq.0) goto 30
if(curve.lt.0) goto 30
rg=dsqrt(x2*x2+y2*y2)
rmmin=(a2+d2d)-dsqrt(cmax**2-z2**2)
rmmax=(a2+d2d)-dsqrt(cmin**2-z2**2)
if (rmmax.gt.rmax2) then
rmax=rmax2
endif
if(rg.gt.rmax.or.rg.lt.rmin) goto 30
write(4,4)t1,f1,x2,y2,z2,rg
continue
10 continue
return
end

---------- End of Subroutine GEAR ----------

Subroutine SECOND called by Subroutine GEAR

This subroutine is used to calculate the coefficients of the gear
surface equations Eq(28) in technical report.

To run this subroutine the input parameters are:
a0 center distance of the first enveloping (mm)
a2 center distance of wormgear (mm)
btr apex angle of the generating plane (rad)
f2 running parameter in the first enveloping (rad)
ratio  speed ratio of wormgear
ratio2  speed ratio of wormgear
rb   radius of base circle (mm)
t2   running parameter in the second enveloping (rad)
tb   tan(bt)

The output in SECOND are:

curve1  worm surface curvature at the contact point
curve2  gear surface curvature at the contact point
curve   relative curvature at the contact point
dw     index (dw=0 the point is a singular point)
x1,y1,z1 contact point coordinates in S1
x2,y2,z2 contact point coordinates in S2
x3,y3,z3 contact point coordinates in S3

subroutine second(t2,f2,a0,a2,ratio,ratio2,rb,tb,btr,
&  x1,y1,z1,x2,y2,z2,x3,y3,z3,
&  curve1,curve2,curve,dw)

implicit double precision(a-h,o-z)
calculate the coordinates of the point for given f1, t1

t1=t2*ratio2
f1=f2*ratio
call eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,rb,tb,btr,
&  p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
if (dw.eq.0) goto 10
x2=-(q*(x2*x1-x2*x1)+r*(q1*s2-q2*s1)+s*(q2*r1-r2*q1))/dw
y2=-(p*(x2*x1-x2*x1)+r*(p1*s2-p2*s1)+s*(p2*r1-r2*p1))/dw
z2=-(p*(q2*x1-q1*x2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
c
t1=dcos(t1)
st1=dsin(t1)
c2=dcos(t2)
st2=dsin(t2)
cf2=dcos(f2)
sf2=dsin(f2)
cf1=dcos(f1)
sf1=dsin(f1)
sft=dsin(f1-t1)
cft=dcos(f1-t1)

c Coordinate transformation

x1=-x2*ct1*ct2+y2*ct1*st2-z2*st1+a2*ct1
y1=-x2*st1*ct2-y2*st1*st2-z2*ct1-a2*st1
z1=-x2*st2-y2*ct2
x3=x2*ct2-y2*st2
y3=x2*st2+y2*ct2
zn=z2
c
c curvature of the gear surface
rr=r*r
pee=1+p*p/rr
pfn=p*q/rr
pg2=1+q*q/rr
b=p*r1-r*p1
d=q*r1-r*q1
c=p*r2-r*p2
e=q*r2-r*q2
c
rto=ratio*ratio
rto2=ratio2*ratio2
ffx=sf2*ct2*cft+tb*ct2*sft+2.0*cf2*ct2*sft/ratio
&+(sf2*ct2*cft-cf2*ct2)/rto
ffy=sf2*st2*cft+tb*st2*sft+2.0*cf2*st2*sft/ratio
&+(sf2*st2*cft+cf2*ct2)/rto
ffz=sf2*sft-tb*cft-2.0*cf2*cft/ratio+s2*sft/rto
ffd=a2*(sf2*cft+tb*sft+2.0*cf2*sft/ratio+s2*cft/rto)
&-a0*s2/rto
ff=ffx*x2-ffy*y2-ffz*z2-ffd
c
&+(sf2*ct2-cf2*st2*cft+ratio2*cf2*ct2*sft)/ratio
&+(sf2*st2*cf2*ct2*cft+ratio2*cf2*st2*sft)/ratio
ftz=ratio2*(sf2*sft-tb*cft-cf2*cft/ratio)
ftd=ratio2+a2*(sf2*cft+tb*sft+cf2*sft/ratio)
ft=ftx*x2+fty*y2+ftz*z2+ftd
c
ttx=-p+2.0*ratio2*(sf2*st2*sft-tb*st2*cft)
&+rto2*(sf2*ct2*cft+tb*ct2*sft)
tty=q+2.0*ratio2*(sf2*ct2*sft-tb*ct2*cft)
&-rto2*(sf2*st2*cft+tb*st2*sft)
ttd=rto2+a2*(sf2*cft+tb*sft)
tt=ttx*x2+ttt*y2+ttz*z2-ttd
c
g=ff**ttt*ff
cm=dcos(btr)/(g*rr)
pl2=cm*(b*b+ttttt-2*b*c*ff+c*cff)
plm2=cm*(b*d*tttt-2*b*c+e*ff)
plm2=cm*(d*tttt-2*d*e*ff)
curve2=(pe2*pl2-2*pf2*plm2+pf2*pl2)/(pe2*pf2*pf2)
nc
curvature of the worm surface
c
pel=1.0+p*p/rr
pf1=p*q/rr
pg1=1.0-q*q/rr
dw=ff**rr*sqrt(p*p+q*q+rr)
aa=p*r1-r*p1
bb=q*r1-r*q1
pl1=aa*aa/dw
pml=aa*bb/dw
pml=bb*bb/dw
curvel=(pel*pl1-2.0*pfl1+pml+plg1*pl11)/(pel*gpl1*gfl1)
c
relative curvature

curve=curve2-curve1

c
10 return
end
c
c--- End of Subroutine SECOND ---
Subroutine EQG called by Subroutine FIRST and SECOND

This subroutine is used to calculate the coefficients of the gear surface equations.

To run this subroutine the input parameters are:

- \( a_0 \)  \( \text{center distance of the first enveloping} (\text{mm}) \)
- \( a_2 \)  \( \text{center distance of the first enveloping} (\text{mm}) \)
- \( b_t \)  \( \text{apex angle of the generating plane} (\text{rad}) \)
- \( f_1 \)  \( \text{running parameter of the first enveloping} (\text{rad}) \)
- \( f_2 \)  \( f_1/\text{ratio} \)
- \( t_1 \)  \( \text{running parameter of the second enveloping} (\text{rad}) \)
- \( t_2 \)  \( t_1/\text{ratio} \)
- \( \text{ratio} \)  \( \text{speed ratio of wormgear} \)
- \( \text{ratio2} \)  \( \text{speed ratio of wormgear} \)
- \( r_b \)  \( \text{radius of base circle} (\text{mm}) \)
- \( \tan(b_t) \)  \( \text{tan} \)

The output in EQW are:

- \( p, q, r, s \)  \( \text{coefficients in Eqs(29)} \)
- \( p_1, q_1, r_1, s_1 \)  \( \text{coefficients in Eqs(30)} \)
- \( p_2, q_2, r_2, s_2 \)  \( \text{coefficients in Eqs(31)} \)

```
subroutine eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,tb,rb,btr,
               p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
implicit double precision(a-h,o-z)

ctl=dcos(t1)
st1=dsin(t1)
ct2=dcos(t2)
st2=dsin(t2)
cf2=dcos(f2)
sf2=dsin(f2)
cfl=dcos(f1)
sfl=dsin(f1)
sft=dsin(f1-t1)
cft=dcos(f1-t1)
```

Here is the Eqs(29)-(30)

```
p=-sf2*ct2*cft-tb*ct2*sft+cf2*st2
q=sf2*st2*cft-tb*st2*sft+cf2*ct2
r=sf2*sft-tb*cft
s=a2*(sf2*cft-tb*sft)-a0*sf2+rb
p1=sf2*ct2*sft-tb*ct2*cft-(sf2*st2+cf2*ct2*cft)/ratio
q1=-sf2*st2*sft+tb*st2*cft-(sf2*ct2-cf2*st2*cft)/ratio
r1=sf2*cft+tb*sft+cf2*sft/ratio
s1=-a2*(sf2*sft-tb*cft-cf2*cft/ratio)-a0*cf2/ratio
p2=q*ratio2*(sf2*ct2*sft-tb*ct2*cft)
q2=-p+ratio2*(sf2*st2*sft-tb*st2*cft)
r2=-ratio2*(sf2*cft+tb*sft)
s2=a2*ratio2*(sf2*sft-tb*cft)

return
end
```

End of Subroutine EQG
Subroutine FIRST called by GEAR

This subroutine is used to calculate the coefficients of the gear surface equations and curvatures.

To run this subroutine the input parameters are:

- a0  center distance of the first enveloping
- a2  center distance of wormgear
- btr apex angle of the generating plane
- f2  running parameter in the first enveloping
- ratio speed ratio of wormgear
- ratio2 speed ratio of wormgear
- rb  radius of base circle
- t2  running parameter in the second enveloping
- tan(bt)
- z2  coordinate of the point in S2

The output in FIRST are:

- curve1 worm surface curvature at the contact point
- curve2 gear surface curvature at the contact point
- curve relative curvature at the contact point
- dw index ( dw=0 the point is a singularity point )
- x1,y1,z1 contact point coordinates in S1
- x2,y2 contact point coordinates in S2
- xn,yn,zn contact point coordinates in Sn

---

subroutine first(t2,f2,a0,a2,ratio,ratio2,rb,tb,btr,z2, & x1,y1,z1,y2,x2,xn,yn,zn, & curve1,curve2,curve,dw)
implicit double precision(a-h,o-z)
calculation of the coordinates of the point
at the given position f1, t1
t1=t2*ratio2
f1=f2*ratio
call eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,rb,tb,btr, & p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2) & dw=p1*q-p*q1
if (dw.eq.0) goto 10
x2=((q1*r-q*r1)*z2-q*s1+q1*s)/dw
y2=-(p1*r-p*r1)*z2-p*s1+p1*s)/dw

cctl=dcos(t1)
sctl=dsint(t1)
ct2=dcos(t2)
st2=dsint(t2)
cf2=dcos(f2)
sf2=dsint(f2)
cfl=dcos(f1)
sf1=dsint(f1)
sft=dsint(f1-t1)
cft=dcos(f1-t1)

Coordinate transformation
c curvature of the gear surface
c
\[ r = r \times r \]
\[ p_e = 1 + p \times p / r \]
\[ p_f = p \times q / r \]
\[ p_g = 1 + q \times q / r \]
\[ b = p \times r_1 - r \times p_1 \]
\[ d = q \times r_1 - r \times q_1 \]
\[ c = b \]
\[ e = d \]
\[ t = r_0 \times t \times r_0 \times r_0 \]
\[ f = f_f \times x_2 + f_f \times y_2 - f_f \times z_2 - f_f d \]
\[ f_t = f_f \times x_2 - f_f \times y_2 + f_f \times z_2 + f_f d \]
\[ f_{tx} = f_f \times x_2 \times s_f t - t_2 \times s_f t \times t_2 \times s_f t \]
\[ f_{ty} = f_f \times y_2 \times s_f t + t_2 \times s_f t \times t_2 \times s_f t \]
\[ f_{tz} = f_f \times z_2 \times s_f t + t_2 \times s_f t \times t_2 \times s_f t \]
\[ g = f_f \times t_t - f_t \times f_t \]
\[ cm = d_c o(s/bt) / (g \times r) \]
\[ p_{l_2} = 0 \]
\[ p_{m_2} = 0 \]
\[ p_{n_2} = 0 \]
\[ c_{u} = c_{u} \times c_{u} \times c_{u} \times c_{u} \times c_{u} \]
\[ c_{v} = c_{v} \times c_{v} \times c_{v} \times c_{v} \times c_{v} \]
\[ c_{w} = c_{w} \times c_{w} \times c_{w} \times c_{w} \times c_{w} \]
\[ c_{x} = c_{x} \times c_{x} \times c_{x} \times c_{x} \times c_{x} \]
\[ c_{y} = c_{y} \times c_{y} \times c_{y} \times c_{y} \times c_{y} \]
\[ c_{z} = c_{z} \times c_{z} \times c_{z} \times c_{z} \times c_{z} \]
\[ c_{t} = c_{t} \times c_{t} \times c_{t} \times c_{t} \times c_{t} \]
\[ c_{u} \times c_{u} \times c_{u} \times c_{u} \times c_{u} \]
\[ c_{v} \times c_{v} \times c_{v} \times c_{v} \times c_{v} \]
\[ c_{w} \times c_{w} \times c_{w} \times c_{w} \times c_{w} \]
\[ c_{x} \times c_{x} \times c_{x} \times c_{x} \times c_{x} \]
\[ c_{y} \times c_{y} \times c_{y} \times c_{y} \times c_{y} \]
\[ c_{z} \times c_{z} \times c_{z} \times c_{z} \times c_{z} \]
\[ c_{t} \times c_{t} \times c_{t} \times c_{t} \times c_{t} \]

118
\[ pl1 = a_a \times a_a / d_w \]
\[ plm = a_a \times b_b / d_w \]
\[ pln = b_b \times b_b / d_w \]
\[ \text{curvel} = (p_e1 \times p_n1 - 2.0 \times p_f1 \times p_m1 + p_g1 \times p_l1) / (p_e1 \times p_g1 - p_f1 \times p_l1) \]

relative curvature

\[ \text{curve} = \text{curve2} - \text{curvel} \]

10 return
end

---END of Subroutine FIRST---

---

Subroutine CONTACT called by MAIN

This subroutine is used to calculate the real contact lines in Sn for the given position \( t_{n2} \). To drive this subroutine, the subroutine DIMEN must be called first and the given position must be given previously.

To run this subroutine the input parameters are:
- \( a_0 \): center distance of the first enveloping (mm)
- \( a_2 \): center distance of wormgear (mm)
- \( b_1 \): face width of worm (mm)
- \( b_2 \): face width of gear (mm)
- \( b_{tr} \): apex angle of the generating plane (rad)
- \( d_h \): clearance (mm)
- \( c_{max} \): maximum thread radius of the gear hob (mm)
- \( c_{min} \): minimum root radius of the gear hob (mm)
- \( d_{dd} \): the change in center distance (mm)
- \( d_{lt} \): half of gear face angle (rad)
- \( df \): step size of \( f_1 \)
- \( dz \): step size of \( z_2 \)
- \( f_{20} \): the angle of the first enveloping begining (rad)
- \( f_{21} \): the angle of the second enveloping ended (rad)
- \( m_p \): number of meshing teeth
- \( r_1 \): pitch radius of worm (mm)
- \( r_2 \): pitch radius of gear (mm)
- \( r_{a1} \): throat radius of worm (mm)
- \( r_{a2} \): throat radius of gear (mm)
- \( r_{ratio} \): speed ratio of wormgear
- \( r_{ratio2} \): speed ratio of wormgear
- \( r_b \): radius of base circle (mm)
- \( r_{f1} \): root radius of worm (mm)
- \( r_{f2} \): root radius of gear (mm)
- \( r_{max2} \): outradius of the gear (mm)
- \( s_{tart} \): the angle of the meshing starting (rad)
- \( t_{20} \): running angle of the second enveloping start (rad)
- \( t_{21} \): running angle of the second enveloping ended (rad)
- \( t_b \): \( \tan(b_t) \)
- \( t_{n2} \): the given position
- \( t_{t2} \): angle between two neighboring gear teeth (rad)

The output of the subroutine are:
- \( c_{contc} \): the length of contact lines of each meshing tooth
- \( n_c \): the number of real meshing pair
The variables in CONTACT are:

t1  the running parameter in the second enveloping (rad)
t2  t1/ratio
z2  the coordinate in gear axis direction

The common exchange dimensions are:

x,y,z save the coordinates of contact point on meshing pair
nn  maximum dimension number of the exchange dimension
rg  save the radius of contact point to gear axis

The results which are not called by other subroutines are written in the data file under the name of 'contc.dat'.

-------------------------------------------------

subroutine contact(a0,a2,ratio,ratio2,np,rb,tb,btr,dbh,ddd,
&   rl,ral,rf1,rf2,ra1,ra2,rfmax2,cmax,cmin,b1,b2,
&   f20,f21,df,t20,t21,tt2,tn2,start,end,dlt,dz,
&   contc,nc,nn,x,y,z,rg)
implicit double precision(a-h,o-z)
dimension x(nn),y(nn),z(nn),rg(nn),contc(nn)
5 format(2x,6f10.4)

write(5,*),' Title of contc.dat file
write(5,*),' the length of contact lines on each meshing teeth'
write(5,*),' '-------------------------------'
write(5,*),' tnl | first | second | total '
write(5,*)' '---------------------'

precaculation

pi=4.0*datan(1.0d0)
dfl=df/100.0
dz2=dz/10.0
nc=0
l=idint(b2/dz2)+1
m=idint(f21/tt2)+1
n=idint(4.0*dlt/dfl)+1

calculation of the coordinates of the contact point

do 10 i=1,m
   t2=tn2+(i-1)*tt2
   if(t2.lt.start.or.t2.gt.end) goto 10
   t1=ratio*t2
   ct1=dcos(t1)
   st1=dsin(t1)
   ct2=dcos(t2)
   st2=dsin(t2)
   ind=0

   do 20 j=1,n
      fl=(t1-2.0*dlt)+(j-1)*dfl
      f2=f1/ratio
      if(f2.lt.start.or.f2.gt.end) goto 20
      call second(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,
&       x1,y1,z1,x2,y2,z2,xn,yn,zn,
&
c
c judge if the point is in the tooth domain
if(dw.eq.0) goto 20
if(curve.lt.0) goto 20
c
if(z2.lt.-b2/2.0.or.z2.gt.b2/2.0) goto 20
rg2=dsqrt(x2*x2+y2*y2)
n2=(a2+d2)-dsqrt((cmax-dh)**2-z2*z2)
r2=(a2+d2)-dsqrt(cmin**2-z2*z2)
if(rm2.gt.rmax2) then
  rm2=rmax2
endif
if(rg2.gt.rm2.or.rg2.lt.rn2) goto 20
c
if(z1.gt.bl/2.0.or.z1.lt.-bl/2.0) goto 20
rw1=dsqrt(x1*x1+y1*y1)
rn1=a2-dsqrt((ra2+dh)**2-z1*z1)
r1=a2-dsqrt((rf2+dh)**2-z1*z1)
if(rw1.gt.rml.or.rw1.lt.rnn) goto 20
c
ind=ind+1
if(ind.gt.nn) then
  print*, ' dimensions nn is not big enough for contact'
  print*, ' increase nn or df in main program.'
  print*, ind
  pause
endif
x(ind)=xn
y(ind)=yn
z(ind)=z2
rg(ind)=rg2
20 continue
c
  calculation of the length of the second contact line
  if (ind.ne.0) then
    rcl2=dsqrt((x(ind)-x(1))**2+(y(ind)-y(1))**2
               & +(z(ind)-z(1))**2)
  endif
do 30 j=1,ind
  x(j)=0
  y(j)=0
  z(j)=0
30 continue

c
if(a0.ne.a2.or.ratio.ne.ratio2) goto 10

c calculation of the coordinates of the contact point
  (this is for the situation where f1=t1)
c
ind=0
do 60 k=1,1
  z2=-b2/2.0+(k-1)*dz2
  f1=t1
  f2=t2
  if(f2.lt.f20.or.f2.gt.f21) goto 60
  call first(t2,f2,a0,a2,ratio,ratio2,rb,tb,dt,z2,
     & xl,yl,zl,y2,x2,yn,zn,
     & curve1,curve2,curve,dw)
121
judge if the point is in the tooth domain

    if(dw.eq.0)    goto 60
    if(curve.1t.0) goto 60

if(zl.gt.bl/2.0.or.zl.lt.-bl/2.0) goto 60
    rw1=sqrt((x1*x1+y1*y1)
    rml=a2-sqrt((rf2+dh)*(rf2+dh)-zl*zl)
    rnl=a2-sqrt((ra2+dh)*(ra2+dh)-zl*zl)
    if(rw1.gt.rml.or.rw1.lt.rnl) goto 60

if(zl.b2/2.0.or.zl.b2/2.0) goto 60
    rg2=sqrt(x2*x2+y2*y2)
    r2=(a0+d2)-sqrt((cmax-dh)**2-z2*z2)
    if (rg2.gt.rmax2) then
        rg2=rmax2
    endif
    if(r2.gt.2.or.r2.gt.2) goto 60

    ind=ind+1
    if(ind.gt.nn) then
        print*, ' dimensions nn is not big enough for contact'
        print*, ' increase nn or dz in main program.'
        print*, ind
        pause
    endif
    x(ind)=xn
    y(ind)=yn
    z(ind)=z2
    rg(ind)=rg2
70 continue

calculation of the length of the second contact line

if (ind.ne.0) then
    rcl1=sqrt((x(ind)-x(1)**2+(y(ind)-y(1)**2
    +r(2(ind)-z(1)**2)
    endif

calculation of the sum of two contact lines on the meshing tooth

if(rcl1.eq.0.and.rcl2.eq.0) goto 10
    nc=nc+1
    contc(nc)=rcl1+rcl2
    write(5,5) t1,rcl1,rcl2,contc(nc)
    rcl1=0
    rcl2=0

do 70 j=1,ind
    x(j)=0
    y(j)=0
    z(j)=0
70 continue
10 continue

return
end

----------------------------------END of Subroutine CONTACT----------------------------------
Subroutine PITCH called by MAIN

This subroutine is used to calculate the coordinates of the worm pitch points, its normal direction and relative velocity at this point in Sj. All the calculations are for the given position tn2. To drive this subroutine DIMEN must be called first and the given position must be input previously.

To run this subroutine the input parameters are:

- a0: center distance of the first enveloping (mm)
- a2: center distance of wormgear (mm)
- b1: face width of worm (mm)
- b2: face width of gear (mm)
- btr: apex angle of the generating plane (rad)
- dh: clearance (mm)
- cmax: maximum throat radius of the gear hob (mm)
- cmin: minimum root radius of the gear hob (mm)
- ddd: the change in center distance (mm)
- dlt: half of gear face angle (rad)
- df: step size of f1
- dz: step size of z2
- end: the angle of the meshing ended (rad)
- ep,epp: convergence criterion for root searching
- fl: the angle of the first enveloping beginning (rad)
- f21: the angle of the second enveloping ended (rad)
- np: number of meshing teeth
- rl: pitch radius of worm (mm)
- r2: pitch radius of gear (mm)
- ra1: throat radius of worm (mm)
- ra2: throat radius of gear (mm)
- ratio: speed ratio of wormgear
- ratio2: speed ratio of wormgear
- rb: radius of base circle (mm)
- rf1: root radius of worm (mm)
- rf2: root radius of gear (mm)
- rmax2: outradius of the gear (mm)
- start: the angle of the meshing starting (rad)
- t20: running angle of the second enveloping start (rad)
- t21: running angle of the second enveloping ended (rad)
- tb: tan(bt)
- tn2: the given position
- tt2: angle between two neighboring gear teeth (rad)

The output of the subroutine are:
- nc: the number of real meshing teeth
- tp2: the position of each meshing pair
- fp2: the f2 value of pitch points
- px,j,pzj: the coordinates of pitch points in Sj
- xpn,ypn,zpn: the normal direction components of pitch point in Sj
- vx,vy,vz: the relative velocity components of pitch point in Sj

The variables in PITCH are:
- t1: the running parameter in the second enveloping (rad)
- t2: t1/ratio
z1 the coordinate in worm axis direction

The common exchange dimensions are:
ff save the fl values of the worm pitch points
xm, zm save the coordinates of the worm mid-plane profile
rr save the radius of worm pitch points
nn maximum dimension number of the exchange dimension

The results which are not called by other subroutines are written in the data file under the name of 'pitch.dat'.

```
subroutine pitch(a0, a2, ratio, ratio2, np, rb, tb, btr, dh, ddd,
& r1, ral, rfl, r2, ra2, rf2, rmax2, cmax, cmin, b1, b2,
& f20, f21, df, t20, t21, tt2, tn2, start, end, dlt, dz,
& mn, zm, zm, ff, rr, ep, epp,
& tp2, fp2, pxj, pxj, xp, ypn, zpn, vx, vy, vz, nc)
implicit double precision(a-h, o-z)
dimension xm(nn), zm(nn), ff(nn), rr(nn), tp2(np), fp2(np),
& pxj(np), pxj(np), xp(np), ypn(np), zpn(np),
& vx(np), vy(np), vz(np)
6 format(2x, 6fl2.4)
```

Title of contc.dat file

```
write(7,*) ' pitch point and normal direction ',
write(7,*) ' tnl | xj | zj | xpn |
& ypn | zpn'
write(7,*) ' pitch point and relative velocity direction',
write(7,*) ' tnl | xj | zj | rvx |
& rvy | rvz'
write(8,*) ' pitch point and normal direction ',
write(8,*) ' tnl | xj | zj | xpn |
& ypn | zpn'
write(8,*) ' pitch point and relative velocity direction',
write(8,*) ' tnl | xj | zj | rvx |
& rvy | rvz'
```

Precalculation

nc=0
yl=0.0
pi=4.0*datan(1.0)
1=m-idint(f21/tt2)+1
start=start*ratio
endl=end*ratio
dfl=df/100.0
ml=midint(pi/(2.0*dfl)+1

Worm profile on the mid-plane (yj=0)

do 50 k=1,m
ind=0
fn2=fn2+(k-1)*tt2
fnl=fn2*ratio2
if(fnl.lt.startl.or.fnl.gt.endl) goto 50

do 10 i=1,ml
fl=fnl-pi/4.0+(i-1)*dfl
call wormt(fl,fnl,y1,a0,ratio,rb,tb,btr,x1,z1,mw)
if(mw.eq.1) goto 10
if(xl.lt.0) goto 10
if(dabs(zl).gt.rb) goto 10

rwl=dsqrt(xl*xl+y1*y1)
rln=a0-dsqr((ra2+dh)*(ra2+dh)-z1*z1)
rml=a0-dsqr((rf2+dh)*(rf2+dh)-z1*z1)
if(rwl.gt.rml.or.rwl.lt.rln) goto 10

ind=ind+1
if(ind.gt.nn) then
    print*, 'the dimension is not big enough in PITCH'
    print*, 'increase nn in main'
    print*, ind
    pause
endif
xm(ind)=xl
zm(ind)=zl
rr(ind)=dsqr((a0-xl)*(a0-xl)+z1*z1)
ff(ind)=fl
continue 10

Find worm pitch points

jt=1
dr=dabs(rr(1)-r2)
do 20 j=2,ind
   if(dabs(rr(j)-r2).lt.dr) then
      dr=dabs(rr(j)-r2)
      jt=j
   endif
20 continue
if(ff(jt).gt.endl.or.ff(jt).lt.start1) goto 25

fl=ff(jt-1)
fr=ff(jt+1)
call root(fl,fr,fnl,y1,a0,ratio,rb,tb,btr,r2,epp,fl,xl,z1)
if(fl.gt.endl.and.fl.lt.start1) goto 25
rmin=dsqr((a0-xl)*(a0-xl)+z1*z1)

Direction of normal line and relative velocity at pitch point

call normal(fl,fnl,xl,y1,z1,a0,ratio,rb,tb,btr, &   px,py,pz,rx,ry,rz)
write(7,6) fnl,xl,z1,px,py,pz
write(8,6) fnl,xl,z1,rx,ry,rz

nc=nc+1
tp2(nc)=fn2
fp2(nc)=fl
pxj(nc)=xl
pzj(nc)=zl
xpn(nc)=px
ypn(nc)=py
zpn(nc)=pz
vx(nc)=rx
vy(nc)=ry
vz(nc)=rz
25 do 30 i=1,ind
   xm(i)=0
   zm(i)=0
   rr(i)=0
   ff(i)=0
30 continue

50 continue

return
end

----------- END of Subroutine PITCH -----------

Subroutine WORMT called by Subroutine PITCH and ROOT

This subroutine is used to calculate the coordinates of the worm
surface in Sj specially for yj=0.

To run this subroutine the input parameters are:
   a0  center distance of the first enveloping plane  (mm)
   btr  apex angle of the generating plane  (rad)
   ratio  speed ratio of the first enveloping  (mm)
   rb  radius of base circle  (rad)
   tan(bt)  the given position (tn2*ratio2)
   ft  running parameter of the first enveloping  (rad)

The output of the subroutine are:
   xm, zm  the coordinates of the worm profile
   mw  the index of the worm profile point on edge curve (mw=1)

subroutine wort(f,t,f1,y,a0,ratio,rb,tb,fbt,xm,zm,mw)
imPLICIT DOUBLE PRECISION(a-h,o-z)

   mw=0
   f2=ft/ratio
   f1=ft-fn1
   call eqw(f1,f2,a0,ratio,rb,tb,
     & p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
   dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
     if(dw.eq.0) goto 10
   fz=-p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
   dw=p*r1-p1*r
   xm= (y*(q1*r-q*r1)+(s1*r-s*r1))/dw
   zm= (y*(p1*q-p*q1)+(p1*s-p*s1))/dw
   if(zm.lt.fz) goto 10
   goto 20

10 mw=1

20 return
end

----------- END of Subroutine WORMT -----------

126
Subroutine ROOT called by Subroutine PITCH

This subroutine is used to search the pitch point from the worm profile in \( S_j \) specially \( y_j = 0 \). The method used here is called golden division.

To run this subroutine the input parameters are:
- \( a_0 \) center distance of the first enveloping \((\text{mm})\)
- \( b_{tr} \) apex angle of the generating plane \((\text{rad})\)
- \( r \) pitch radius of gear \((\text{mm})\)
- \( \text{ratio} \) speed ratio of the first enveloping \((\text{mm})\)
- \( \tan(b) \)
- \( \text{fnl} \) the given position \((\text{rad})\)
- \( \text{fi}, \text{fe} \) interval of \( \text{fi} \) for root searching \((\text{rad})\)
- \( \text{epp} \) convergence criterion for root searching \((\text{rad})\)

The output of the subroutine are:
- \( x_1, z_1 \) the coordinates of the worm pitch point in \( S_j \)
- \( \text{fl} \) the value of running parameter at pitch point

---

```fortran
subroutine root(fi,fe,fnl,y,a0,ratio,rb,\(\tan(b)\),fnl,fl,x1,z1)
  implicit double precision(a-h,o-z)
  call wormt(fi,fnl,y,a0,ratio,rb,\(\tan(b)\),xm,zm,mw)
  gi=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
  call wormt(fe,fnl,y,a0,ratio,rb,\(\tan(b)\),xm,zm,mw)
  ge=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
  df=fe-fi
  if(df.gt.epp) goto 150
  ft1=fi+.382*df
  ft2=fi+.618*df
  call wormt(ft1,fnl,y,a0,ratio,rb,\(\tan(b)\),xm,zm,mw)
  gl=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
  call wormt(ft2,fnl,y,a0,ratio,rb,\(\tan(b)\),xm,zm,mw)
  g2=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
  if(g1*gi.gt.0) goto 120
  fe=ft1
  ge=gl
  goto 110
120 if(g2*gi.gt.0) goto 130
  fi=ft1
  fe=ft2
  gi=gl
  ge=g2
  goto 110
130 fi=ft2
  gi=g2
  goto 110
150 fl=(fi+fe)/2.0
  call wormt(fl,fnl,y,a0,ratio,rb,\(\tan(b)\),xm,zm,mw)
  goto 170
160 mc=1
170 return
end
```

END of Subroutine ROOT

127
Subroutine NORMAL called by Subroutine PITCH

This subroutine is used to calculate the properties of the worm pitch point in Sj.

To run this subroutine the input parameters are:
- \( a_0 \) center distance of the first enveloping (mm)
- \( \theta_r \) apex angle of the generating plane (rad)
- \( f_1 \) the value of running parameter at pitch point
- \( \text{ratio} \) speed ratio of the first enveloping
- \( r_b \) radius of base circle (mm)
- \( \tan(b) \)
- \( x_l, z_l \) the coordinates of the worm pitch point in Sj

The output of the subroutine are:
- \( w_{nx}, w_{ny}, w_{nz} \) the normal direction components of pitch point in Sj
- \( x_w, y_w, z_w \) the relative velocity components of pitch point in Sj

```plaintext
subroutine normal(f1,fn1,x1,y1,z1,a0,ratio,r_b,theta_r,
& wnx,wny,wnz,xw,yw,zw)
implicit double precision(a-h,o-z)
f2=f1/ratio
cf2=dcos(f2)
sf2=dsin(f2)
cf1=dcos(f1-fn1)
sf1=dsin(f1-fn1)

wnx=cf1*sf2+tb*sf1
wny=tb*cf1-sf1*sf2
wnz=-cf2
rx=x1*(1-a0/dsqrt(x1*x1+y1*y1))
ry=y1*(1-a0/dsqrt(x1*x1+y1*y1))
rz=z1
xw=wny*rz-wnz*ry
yw=wnz*rx-wnx*rz
zw=wnx*ry-wny*rx
rlu=dsqrt(xw*xw+yw*yw)
rlf=dsqrt(xw*xw+yw*yw+zw*zw)
read=acos(rlu/rlf)
return
end
```

END of Subroutine ROOT

Subroutine LOAD called by MAIN

This subroutine is used to calculate load sharing among the meshing teeth. All the calculations are under the given position \( \text{tn2} \). To drive this subroutine the subroutines DIMEN, CONTACT and PITCH must be called first.
To run this subroutine the input parameters are:
- a2: center distance of wormgear (mm)
- nc: the number of real meshing pairs
- nmax: number of meshing teeth
- power: output power from gear shaft (kw)
- r2: pitch radius of gear (mm)
- ratio2: speed ratio of wormgear
- rpm: input speed from worm shaft (rpm)
- t2: the position of each meshing pair
- uf: friction coefficient
- x1,z1: the coordinates of pitch points in Sj
- pnx, pny, pnz: the normal direction components of pitch point in Sj
- cont: the length of contact lines on each meshing pair

The other results which are not called by other subroutines are written in the data file under the name of 'load.dat'

```fortran
subroutine load (a2,ratio2,r2,nmax,nc,t2,
    & x1,z1,pnx,pny,pnz,cont,wtor,wtorf,forn,fornf)
    implicit double precision(a-h,o-z)
    dimension t2(nmax),cont(nmax),x1(nmax),z1(nmax),
    & pnx(nmax),pny(nmax),pnz(nmax),
    & rei(10),refi(10),ww(10),wwf(10)
    format(2x,6f12.4)
write(9,*) 'normal load and worm torque distribution'
write(9,*) 'pair No. | normal load | worm | to
    &que'
write(9,*) 'pair No. | uf=0 | uf<>0 | uf=0 |
    &--------
pi=4.0*datan(1.0d0)
grpm=rpm/ratio2
torqeg=9.550*1.0e6*power/grpm
calculation of the sum of the contact lines
dd=0.0
ddf=0.0
do 20 i=1,nc
    pn=dsqrt(pnx(i)*pxn(i)+pny(i)*pny(i)+pnz(i)*pnz(i))
    xn=dabs(pnx(i)/pn)
    yn=dabs(pny(i)/pn)
    zn=dabs(pnz(i)/pn)
    rei(i)=zn*(a2-xl(i))-xn*z1(i)
    pv=dsqrt(r2*r2+(xl(i)*ratio2)**2)
    refi(i)=uf*r2*r2/pv
    ww(i)=yn
    wwf(i)=yn+uf*xl(i)*ratio2/pv
    dd=d+rei(i)*cont(i)
    ddf=df+((rei(i)-refi(i))*cont(i)
20 continue
```
calculate normal load and worm torque distribution (uf=0)

    do 30 j=1,nc
        gfor=torqeg*cont(j)/dd
        wtor=gfor*ww(j)*xl(j)

calculation of the normal load and worm torque distribution

with friction

    gforf=torqeg*cont(j)/ddf
    wtorf=gforf*wwf(j)*xl(j)
    write(9,9) t2(j)*ratio2,gfor,gforf,wtor,wtorf
30 continue

return
end

END of Subroutine LOAD
Appendix 3 The Principal Relative Curvatures

Assume that a surface $\Sigma_1$ is a simple surface. At a point $Q$ of $\Sigma_1$ there exists a normal $n$. The tangent plane of the surface $\Sigma_1$ at $Q$ is perpendicular to the normal $n$. Any plane containing the normal $n$ is perpendicular to the tangent plane. This plane is called a normal plane of the surface $\Sigma_1$ at point $Q$. The intersection of the normal plane and the tangent plane is a straight line. The direction of the straight line is called a tangent direction of surface $\Sigma_1$ at point $Q$. Therefore, at a point on $\Sigma_1$ there is only one normal direction and only one tangent plane, but there are infinitely many tangent directions and infinitely many normal planes.

At point $Q$ the intersection of surface $\Sigma_1$ and a normal plane gives a planar curve, correspondingly the intersection of the normal plane and the tangent plane gives a tangent direction. The curvature of the planar curve at point $Q$ is called the normal curvature of surface $\Sigma_1$ in this tangent direction. Let the point $Q$ be given by the position vector $r(u,v)$, where $u$, $v$ are two independent variables. Then the normal curvature is:

$$k_n = \frac{Ldu^2+2Mdudv+Ndv^2}{Edu^2+2Fduv+Gdv^2}$$  \hspace{1cm} (a.1)

where $E=r_u^2$, $F=r_u \cdot r_v$, $G=r_v^2$, $L=r_{uu} \cdot n$, $M=r_{uv} \cdot n$, $N=r_{vv} \cdot n$; and $n$ is the normal direction given by $r_u \times r_v | r_u \times r_v |$. If the position of $Q$ is given by $r(x,y,z)$ where $z=z(x,y)$ represents the equation of the surface, then $E$, $F$, $G$, $L$, $M$, $N$ can be written as:

$$E = 1 + (\frac{\partial z}{\partial x})^2, \quad F = \frac{\partial x}{\partial u} \frac{\partial z}{\partial u}, \quad G = 1 + (\frac{\partial z}{\partial y})^2$$

$$L = \frac{\partial^2 z}{\partial x^2}, \quad M = \frac{\partial x}{\partial u} \frac{\partial^2 y}{\partial u^2}, \quad N = \frac{\partial^2 z}{\partial y^2}$$  \hspace{1cm} (a.2)

where $DB = \sqrt{1 + (\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2}$

131
The sign of the normal curvature is determined by the normal direction. If the normal direction points to the center of curvature center, the normal curvature is positive, otherwise it is negative.

Along different tangent directions the normal curvatures of a surface are different. The principal curvatures are the extreme values of the normal curvatures. The corresponding tangent directions are called the principal directions (Kreyszig, 1959).

If two surfaces \( \Sigma_1 \) and \( \Sigma_2 \) are in meshing at a contact point \( Q \), the normals of the two surfaces at the contact point coincide. Along a tangent direction let the normal curvatures of the two surfaces be \( k_n^{(1)} \) and \( k_n^{(2)} \), then in this direction the relative curvature of two surfaces is defined as:

\[
K_r = k_n^{(1)} - k_n^{(2)}
\]  

(a.3)

The principal relative curvatures are the extreme values of the relative curvatures, the corresponding directions are the principal relative directions. If the two surfaces are simple surfaces, there are two principal relative curvatures and two principal relative directions.

If the two surfaces contact along a line, the direction of the contact line is one of the principal relative direction and the corresponding principal relative curvature is zero. The other principal relative direction is perpendicular to the direction of the contact line and the principal relative curvature is (Litvin, 1989; Shen, et al. 1983):

\[
K_r = \frac{E_1 N_1 - 2 F_1 M_1 + G_1 L_1}{E_1 G_1 - F_1^2} - \frac{E_2 N_2 - 2 F_2 M_2 + G_2 L_2}{E_2 G_2 - F_2^2}
\]  

(a.4)

In the second enveloping process the gear surface, Eqs. 27, is generated by the worm surface, Eqs. 18. In this process the two surfaces are in line contact. Let \( P = A_{\phi}^{(21)}F \), the equations of the gear surface can be written as:
\[ P = 0 \]
\[ D_{\psi} P = 0 \]  \hspace{2cm} (a.5)
\[ D_{\theta} P = 0 \]

By solving Eqs. 18, the relations of \( z_1(x_1, y_1, \psi(x_1, y_1)) \) can be found. A point on the worm surface can then be expressed as \( r_1(x_1, y_1, z_1(x_1, y_1)) \). Further, by solving Eqs. 27, the relations of \( z_2(x_2, y_2), \psi(x_2, y_2), \theta(x_2, y_2) \) can be found. A point on the gear surface can be written as \( r_2(x_2, y_2, z_2(x_2, y_2)) \). Then, by using Eq. a.4 the principal relative curvature is obtained. In this equation, one should make the following substitutions:

\[ E_1 = \frac{F_{x_1}^2 + F_{z_1}^2}{F_{z_1}^2}, \quad F_1 = \frac{F_{x_1} F_{y_1}}{F_{z_1}^2}, \quad G_1 = \frac{F_{y_1}^2 + F_{z_1}^2}{F_{z_1}^2} \]  \hspace{2cm} (a.6)

\[ L_1 = \frac{F_{x_1} F_{z_1}}{F_{x_1} F_{z_1}}, \quad M_1 = \frac{F_{x_1} F_{z_1}}{F_{x_1} F_{z_1}}, \quad N_1 = \frac{F_{y_1} F_{z_1}}{F_{y_1} F_{z_1}} \]  \hspace{2cm} (a.7)

\[ E_2 = \frac{F_{x_1}^2 + F_{z_1}^2}{F_{x_1}^2}, \quad F_2 = \frac{F_{x_1} F_{y_1}}{F_{x_1}^2}, \quad G_2 = \frac{F_{y_1}^2 + F_{z_1}^2}{F_{x_1}^2} \]  \hspace{2cm} (a.8)

\[ L_2 = m \left( b^2 P_{b,0} - 2bc P_{b,0} + c^2 P_{b,0} \right) \]
\[ M_2 = m \left( bd P_{b,0} - (be + cd)P_{b,0} + ce P_{b,0} \right) \]
\[ N_2 = m \left( d^2 P_{b,0} - 2de P_{b,0} + e^2 P_{b,0} \right) \]  \hspace{2cm} (a.9)

where

\[ DA = F_{x_1}^2 F_{y_1} \sqrt{F_{x_1}^2 + F_{y_1}^2 + F_{z_1}^2} \]  \hspace{2cm} (a.10)

133
\[ d = \begin{vmatrix} x_1 & x_2 \\ x_1^2 & x_1^3 \end{vmatrix}, \quad c = \begin{vmatrix} x_1 & x_2 \\ x_1^2 & x_1^3 \end{vmatrix} \]

\[ d = \begin{vmatrix} y_1 & y_2 \\ y_1^2 & y_1^3 \end{vmatrix}, \quad e = \begin{vmatrix} y_1 & y_2 \\ y_1^2 & y_1^3 \end{vmatrix} \] (a.11)

\[ g = \begin{vmatrix} v_1 & v_1^2 \\ v_1 & v_1^2 \end{vmatrix}, \quad m = \frac{\cos \beta}{gR_s} \]

\[ F_q = \frac{\partial F}{\partial q}, \quad F_{\varphi} = \frac{\partial^2 F}{\partial \varphi \partial r} \] (a.12)

\( (q, r = x_1, y_1, z_1, \psi_1) \)

\[ P_s = \frac{\partial P}{\partial s}, \quad P_{st} = \frac{\partial^2 P}{\partial s \partial t} \] (a.13)

\( (s, t = x_1, y_1, z_1, \psi_1, x_2, y_2, z_2, \Theta_2) \)
Study of the Kinematic and Dynamic Characteristics of a Wormgear Transmission for Helicopter Applications

D.C. Sun and Qin Yuan

State University of New York at Binghamton
Department of Mechanical and Industrial Engineering
Binghamton, New York 13902–6000

Vehicle Propulsion Directorate
U.S. Army Research Laboratory
Cleveland, Ohio 44135–3191
and
NASA Lewis Research Center
Cleveland, Ohio 44135–3191

Project Manager, David E. Brewe, Materials Division, organization code 5140, NASA Lewis Research Center, (216) 433–6067.

Unclassified - Unlimited
Subject Category 37

The first phase of the study of the performance of a wormgear transmission is reported. In this phase the work included the selection of a double-enveloping wormgear type, and its dimensions, suitable for use in helicopter transmissions; the 3-D graphics representation of the selected wormgear using the I-DEAS software; the analysis of the kinematics of meshing; the analysis of load sharing among the meshing teeth; and the implementation of the analyses in a computer program. The report describes the analyses, their results, and the use of the computer programs.

WU-505–62–0K–00
C–NAG3–1316
IL161102AH45

E–8633

NASA CR–195287
ARL–CR–78

Unclassified

Unclassified

Unclassified

139
A05

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. Z39–18
289–102