NASA
Contractor Report 4259

AVSCOM
Technical Report 89-C-014

Generation and Tooth Contact Analysis of Spiral Bevel Gears With Predesigned Parabolic Functions of Transmission Errors

Faydor L. Litvin and Hong-Tao Lee

GRANT NAG3-783
NOVEMBER 1989

NASA

US ARMY
AVIATION
SYSTEMS COMMAND
AVIATION R&T ACTIVITY

NASA
Contractor Report 4259

AVSCOM
Technical Report 89-C-014

Generation and Tooth Contact Analysis of Spiral Bevel Gears With Predesigned Parabolic Functions of Transmission Errors

Faydor L. Litvin and Hong-Tao Lee
University of Illinois at Chicago
Chicago, Illinois

Prepared for
Propulsion Directorate
USAARTA-AVSCOM and
NASA Lewis Research Center
under Grant NAG3-783

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1989

Contents

NOMENCLATURE

Index

Upper case English characters except "I" and digit numbers indicate surfaces.

Lower case English characters indicate coordinate systems.

c	tool surface ($c = G, P$)
w	work surface ($w = 1, 2$)
$\mathcal{F}$, $\mathcal{Q}$	tool surfaces or work surfaces
G	gear tool surface
P	pinion tool surface
1	pinion surface
2	gear surface
I	first principal
II	second principal

Matrix

$[A]$	3 by 4 symmetric augmented matrix which relates principal curvatures and directions for mating surfaces
$[B]$	4 by 1 matrix representing homogenous coordinates of point B
$[L_{ab}]$	3 by 3 matrix describing the transformation of vector from the S_b coordinate system to S_a coordinate system
$[M_{ab}]$	4 by 4 matrix describing the transformation of coordinates from the S_b coordinate system to S_a coordinate system
$[N]$	3 by 1 matrix representing components of normal vector $\vec{N}$

$[n]$	3 by 1 matrix representing components of unit normal vector $\vec{n}$
$[\omega]$	3 by 1 matrix representing components of angular velocity vector $\vec{\omega}$

Vector

$\vec{B}$	position vector of point B on a surface
$\vec{B}_u$	$\partial\vec{B}/\partial u$
$\vec{B}_v$	$\partial\vec{B}/\partial v$
$\vec{e}_{I}$, $\vec{e}_{II}$	unit vectors along the principal directions of the surface at the contact point
$\vec{\imath}$, $\vec{\jmath}$, $\vec{k}$	base vectors along axes X, Y, and Z, respectively
$\vec{N}$	normal vector of point B on a surface
$\vec{n}$	unit normal vector of point B on a surface
$\vec{n}_u$	$\partial\vec{n}/\partial u$
$\vec{n}_v$	$\partial\vec{n}/\partial v$
$\vec{V}^{(CW)}$	slide velocity of surfaces Σ_C and Σ_W
${}_{tr}\vec{V}$	transfer velocity
${}_{r}\vec{V}^{(1)}$, ${}_{r}\vec{V}^{(2)}$	velocity vectors of contact point in its motion over the pinion and gear surfaces, respectively
$\vec{\omega}$	angular velocity
$\vec{\omega}^{(\mathcal{F}\mathcal{Q})}$	relative angular velocity of surface $\mathcal{F}$ with respect to surface $\mathcal{Q}$
$\vec{\tau}$	tangent vector

English Upper Case

A	mean pitch cone distance

A_0, A_1, A_2	coefficient of a quadratic equation
$\mathcal{A}$, $\mathcal{B}$	auxiliary parameters
B	point on a surface
C^n	class of a function
E, F, G	auxiliary parameters for first fundamental form
E_m	machining offset
E^3	three-dimensional space
$\mathcal{F}$	zero function
I	first fundamental form
II	second fundamental form
$\mathcal{I}$	Interval
L	generating planar curve for a sphere
L, M, N	auxiliary parameters for second fundamental form
L_m	vector sum of machine center to back and sliding base
M	middle point on the gear surface
N	number of teeth
P	plane
R	radius of a circle
R_{c_x}, R_{c_z}	x and z coordinates, respectively, of the center of a circle in the S_c coordinate system
S	coordinate system
$\mathcal{T}$	the smaller absolute value of $\mathcal{A}$ and $\mathcal{B}$
$V_{c_I}^{(WC)}$	the projection of $\vec{V}^{(WC)}$ on the $\vec{e}_{c_I}$
$V_{c_{II}}^{(WC)}$	the projection of $\vec{V}^{(WC)}$ on the $\vec{e}_{c_{II}}$
W	point width
${}_rV_{2_I}^{(1)}$, ${}_rV_{2_{II}}^{(1)}$	the projections of vector ${}_r\vec{V}^{(1)}$ on vectors $\vec{e}_{2_I}$ and $\vec{e}_{2_{II}}$, respectively

X_{MCB}	machine center to back
X_{SB}	sliding base

English Lower Case

a	constant (Chapter 1)
a	semimajor axis of the contact ellipse (Chapter 3 and Appendix A)
a_{ij}	element of matrix $[A]$ ($i = 1,2,3$ $j = 1,2,3$)
b	constant (Chapter 1)
b	semiminor axis of the contact ellipse (Chapter 3 and Appendix A)
b_1, b_2	auxiliary variables
c	clearance
c_{11}, c_{12}, c_{13}	auxiliary variables
d_G	average diameter of gear cutter
d_1, d_2, d_3	auxiliary variables
f_1, f_2	auxiliary variables
$m_{\mathcal{F}Q}$	gear ratio
$m'_{\mathcal{F}Q}$	derivative of gear ratio with respective to ϕ_Q
q	cradle angle
r	tip radius of the cutter
s	radial setting
t	semimajor axis of the contact ellipse
t_1, t_2, t_4	auxiliary variables
u	surface coordinates of a cone surface
u_{11}, u_{12}, u_{21}	auxiliary variables

u_{22}, u_{31}, u_{32} auxiliary variables

Non-English Upper Case

Σ	surface
Γ	shaft angle
Υ	angle measured counterclockwise from the root to the tangent of the path on the gear surface
$\aleph$	ratio constant
$\Re$	open rectangle
$\triangle$	discriminant of an equation

Non-English Lower Case

α	orientation angle of ellipse
β	mean spiral angle
δ	dedendum angle
ϵ	specified tolerance value
γ	root angle
κ	principal curvature
κ_{Λ}	$\kappa_{2\Sigma} - \kappa_{1\Sigma}$
κ_{Σ}	$\kappa_{I} + \kappa_{II}$
κ_{Δ}	$\kappa_{I} - \kappa_{II}$
κ_n	normal curvature
κ_r	relative normal curvature of the mating surface

λ	surface coordinate of a surface of revolution
μ	pitch angle
ν_1, ν_2	angles formed between vectors ${}_r\vec{V}^{(1)}$ and $\vec{e}_{2_I}$, and ${}_r\vec{V}^{(2)}$ and $\vec{e}_{2_I}$, respectively
ω	angular velocity
ϕ_C	turn angle of the cradle when the work is being cut
ϕ_W	rotation angle of the work while it is being cut
ϕ'_W	rotation angle of one member while it is being in meshing with another member of a pair of gears
$\phi'_2(\phi'_1)$	transmission function, the rotation angle of the gear in terms of that of the pinion in a pair of meshing gears
$\breve{\phi}'_2(\phi'_1)$	transmission function of a pair conjugate gear
$\triangle\phi'_2(\phi'_1)$	transmission error function
$(\triangle\phi'_2)^{(1)}$	predesigned parabolic function of transmission errors
$(\triangle\phi'_2)^{(2)}$	linear function of transmission errors induced by misalignment
ψ'_1, $\triangle\psi'_2$	expressions of ϕ'_1 and $\triangle\phi'_2$ in a new coordinate system
ψ	blade angle
θ	surface coordinate of a cone surface and a surface of revolution
$\sigma_{\mathcal{F}\mathcal{Q}}$	angle measured counterclockwise from $\vec{e}_{\mathcal{F}_I}$ to $\vec{e}_{\mathcal{Q}_I}$
τ	auxiliary variable, $\theta \mp q \pm \phi_C$
ε	elastic approach
ϖ	angle formed by the tangent to the curvature and first principal curvature

SUMMARY

A new approach for determination of machine-tool settings for spiral bevel gears is proposed. The proposed settings provide a predesigned parabolic function of transmission errors and the desired location and orientation of the bearing contact. The predesigned parabolic function of transmission errors is able to absorb piece-wise linear functions of transmission errors that are caused by the gear misalignment and reduce the gear noise. The gears are face-milled by head cutters with conical surfaces or surfaces of revolution.

A computer program for simulation of meshing, bearing contact and determination of transmission errors for misaligned gear has been developed.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The most important criteria of quality of meshing and contact of gears are the low level of noise and the sufficient dimensions and location of the bearing contact. Sometimes these requirements are contradictory and can be achieved by a compromise in the process of gear synthesis. Such a method of synthesis for spiral bevel gears has been developed in this report.

Traditionally, Gleason's spiral bevel gears are designed and manufactured with non-conjugate tooth surfaces. By varying machine-tool settings the transmission errors can be of different forms, which included a piece-wise linear function, an "S" curve, and a parabolic function, symmetrical or otherwise. Only a parabolic function with gear lagging is prefered. The problem encountered is that it is very difficult to reduce the level of a parabolic function of transmission errors with gear lagging.

Litvin et al. [1] proposed a method for generation of spiral bevel gears with conjugate tooth surfaces. Ideally such conjugate pair provides zero transmission errors. In practice, spiral bevel gears are frequently required to operate under misalignment caused by mounting tolerances and deflections. Using the Tooth Contact Analysis (TCA) programs we have found that the conjugate spiral bevel gears cause lead functions of transmission errors — strong monotonous increasing or decreasing functions for a cycle of meshing. These functions may be considered as linear functions or almost linear functions (Figure 1). Due to gear misalignment the bearing contact can be shifted from the desired location even to the tooth edge. For this reason it is necessary to control also the location and dimensions of the bearing contact.

There is an opportunity to reach these goals if the gears will be designed as non-conjugate pairs that transform rotation with a predesigned parabolic function of transmission errors. Then, as it will be proven in the next section, a linear function of transmission errors will be absorbed and the sensitivity of the gears to misalignment will be reduced.

The determination of pinion machine-tool settings is based on the local synthesis of the gears proposed by Litvin [2, 3, 4]. The local synthesis must satisfy the following requirements:

1. The gear surfaces are in tangency at the chosen mean contact point.
2. The tangent to the path of contact has the prescribed direction at the mean contact point.
3. The contact ellipse for the tooth surfaces has the desired dimensions at the mean contact point.
4. The transmission function $\phi_2(\phi_1)$ has the prescribed value at the mean contact point and its second derivative is negative on gear convex side and positive on gear concave side. Here, ϕ_1 and ϕ_2 are the rotation angles of the pinion and gear while they are being cut, respectively.

Requirement 4 means that the function of transmission errors is a parabolic one with gear lagging within the neighborhood of the mean contact point.

Traditionally, a pair of Gleason's gears is generated by two cones. In some cases the pinion is generated by a surface of revolution instead of a cone surface to obtain better bearing contact and to avoid an edge contact. Both cases are investigated and the machine-tool settings are determined according to the local synthesis and predesigned function of transmission errors.

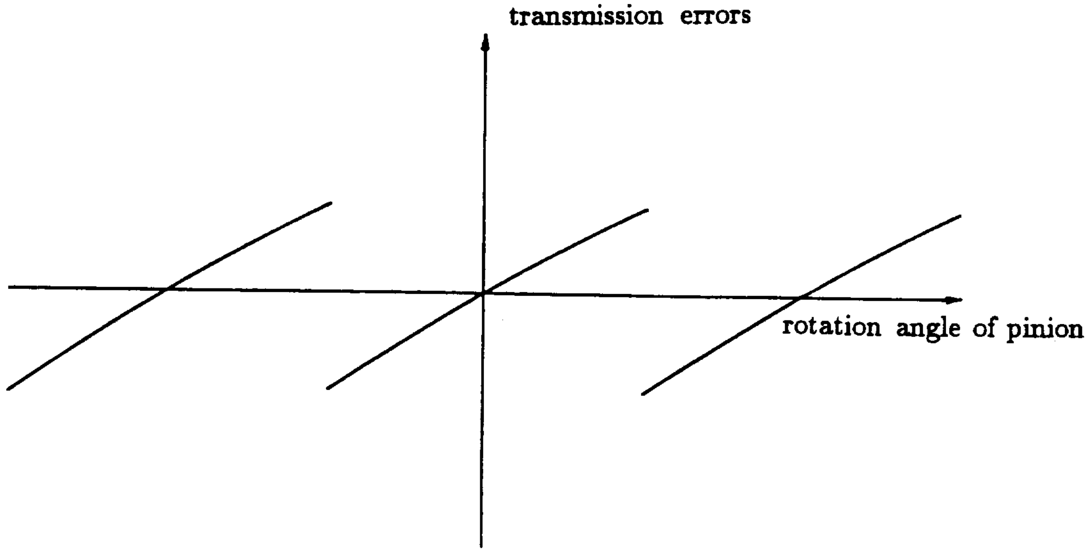


Figure 1: Transmission errors of conjugate gears caused by misalignment.

1.2 Transmission Errors And Its Compensation

In theory a pair of mating gears transforms rotation with a constant gear ratio

$$m_{21} = \frac{\omega_2}{\omega_1} = \frac{N_1}{N_2} \tag{1.1}$$

where ω_1 and ω_2 are the angular velocities of the gears

N_1 and N_2 are the numbers of teeth of pinion and gear, respectively

Therefore, the transmission function is expected to be linear for ideal gears, i.e.,

$$\breve{\phi}_2'(\phi_1') = \frac{N_1}{N_2}\phi_1' \tag{1.2}$$

However, the actual function $\phi_2'(\phi_1')$ is always different from $\breve{\phi}_2'(\phi_1')$ except at the mean contact point. The transmission errors are defined as the difference of theoretical and actual functions of transmission functions, i.e.,

$$\triangle\, \phi_2'(\phi_1') = \phi_2'(\phi_1') - \breve{\phi}_2'(\phi_1') = \phi_2'(\phi_1') - \frac{N_2}{N_1}\phi_1' \tag{1.3}$$

In general the transmission errors of gears may occur due to the following four reasons [5]:

1. The gears cannot exactly transform rotation described by equation (1.2) because of the method of their generation. Spiral bevel gears and hypoid gears that are generated by Gleason methods are good examples for this case.

2. The gear axes are misaligned or the gear shafts are deflected. Zhang, in his dissertation [5], has proved that the deflected gear shafts can be modeled as misaligned gear axes. Spur gears, helical gears, and conjugate spiral bevel gears are very sensitive to misalignment.

3. Heat treatment deviation of the real gear surface is one of the most important factors in surface distortion.

4. The elastic deformation of gear tooth surfaces under applied load.

Cases 1 and 2 among the above-mentioned are the main sources of transmission errors. They will be discussed later. The topics of 3 and 4 are beyond the scope of this report and will not be discussed.

For a pair of conjugate gears under misalignment, the investigation results in that the transmission function $\phi_2'(\phi_1')$ becomes a discontinuous piece-wise function that is linear or almost linear for each cycle of meshing as shown in Figure 2. The corresponding transmission errors determined by equation (1.3) are also an approximately piece-wise linear function as shown in Figure 3. Such functions cause a discontinuity in the regular tooth meshing and usually impact at the transfer point.

There is another type of function of transmission errors that is a piece-wise parabolic function as shown in Figure 4. This type of transmission errors does not cause a discontinuity of regular tooth meshing at transfer points. Gears with this type of transmission errors are not so sensitive to misalignment. This statement is based on an investigation into the interaction of a parabolic function with a linear function.

Consider that a pair of gears is predesigned with a parabolic function of transmission errors. This function may be represented by

$$(\triangle\phi_2')^{(1)} = a(\phi_1')^2 \tag{1.4}$$

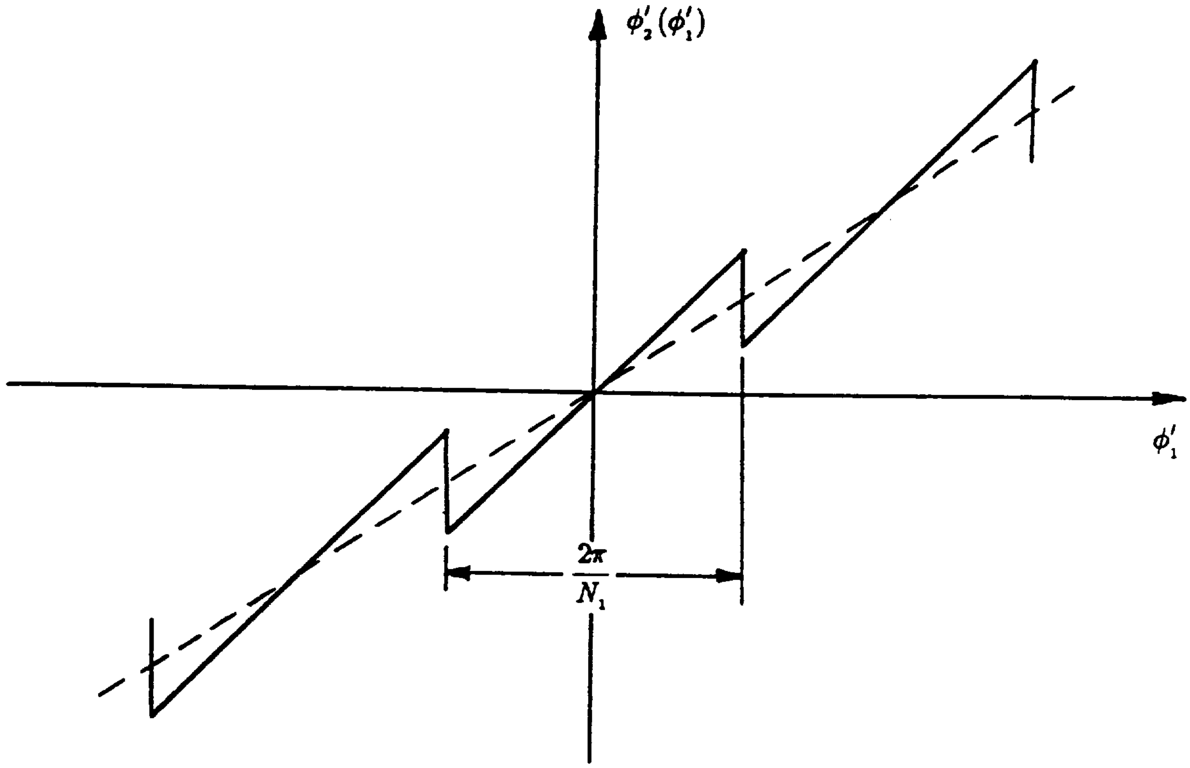


Figure 2: Transmission functions of gears under misalignment.

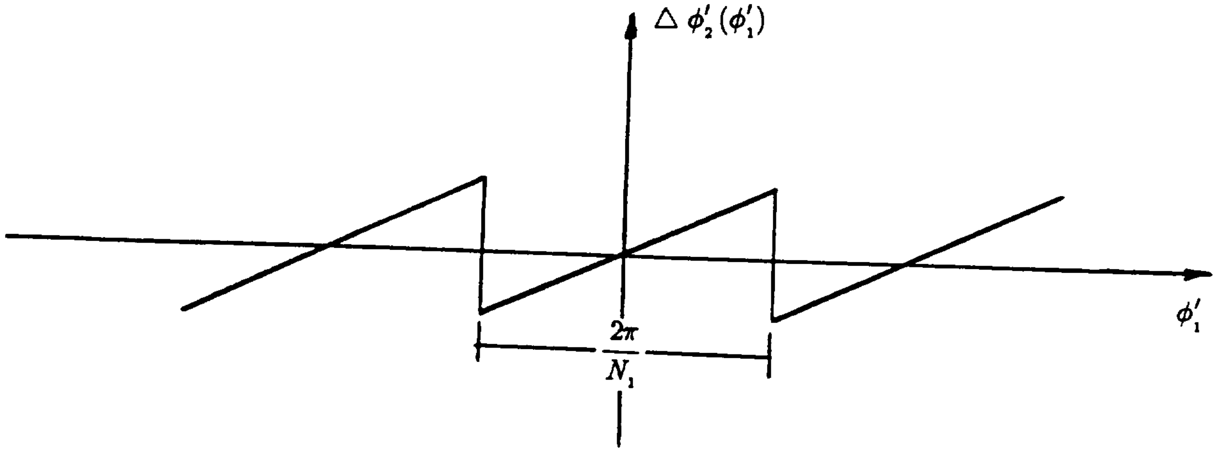


Figure 3: Transmission errors caused by gear Misalignment.

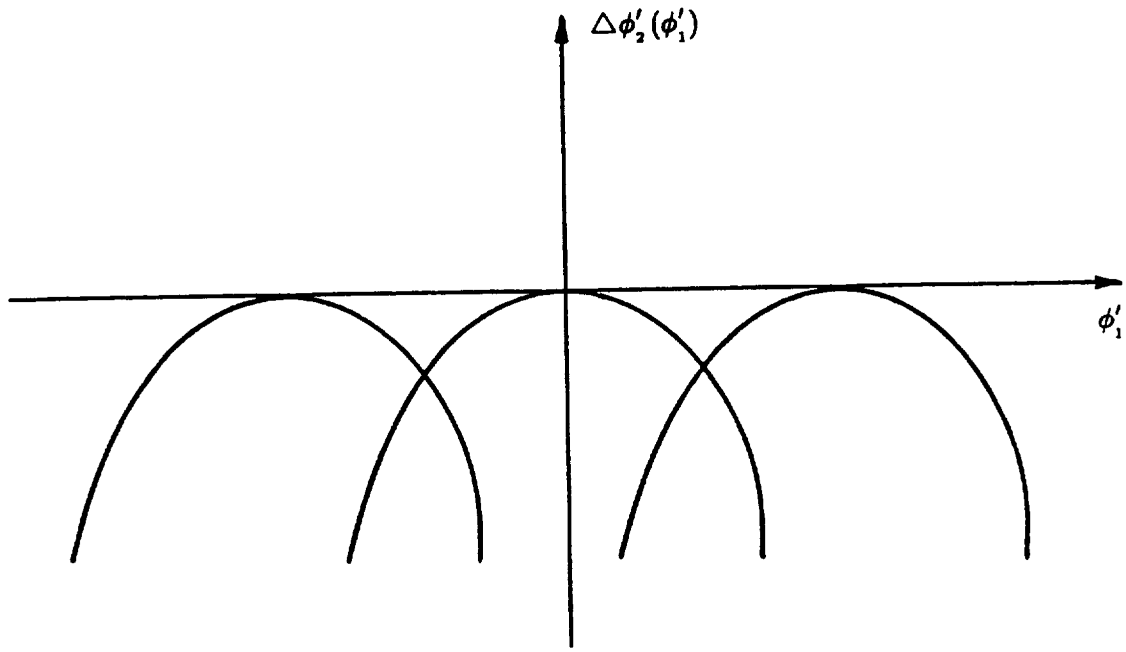


Figure 4: A piece-wise parabolic function of transmission errors.

The level of transmission errors is $a(2\pi/N_1)^2$.

Misalignment induces a linear function of transmission errors. It may be represented by

$$(\triangle\phi_2')^{(2)} = b\phi_1' \tag{1.5}$$

Since $(\triangle\phi_2')^{(1)}$ and $(\triangle\phi_2')^{(2)}$ are very small, the principle of superposition can be applied for the interaction of functions $(\triangle\phi_2')^{(1)}$ and $(\triangle\phi_2')^{(2)}$. Therefore, the resulting function is

$$\triangle\, \phi_2' = (\triangle\phi_2')^{(1)} + (\triangle\phi_2')^{(2)} = a(\phi_1')^2 + b\phi_1' \tag{1.6}$$

Equation (1.6) can be rewritten in a new coordinate system by (Figure 5)

$$\triangle\, \psi_2' = a(\psi_1')^2 \tag{1.7}$$

where

$$\triangle\, \psi_2' = \triangle\phi_2' + \frac{b^2}{4a} \qquad \psi_1' = \phi_1' + \frac{b}{2a} \tag{1.8}$$

From equation (1.7) we know that although the misalignment occurs, the resulting function of transmission errors represents the same parabolic function that has been translated with respect to the given parabolic function. This means that the predesigned parabolic function $(\triangle\phi_2')^{(1)}$ will absorb the linear function $(\triangle\phi_2')^{(2)}$ induced by misalignment. The level of transmission errors remains the same since the parabolic function of each tooth translates the same amount.

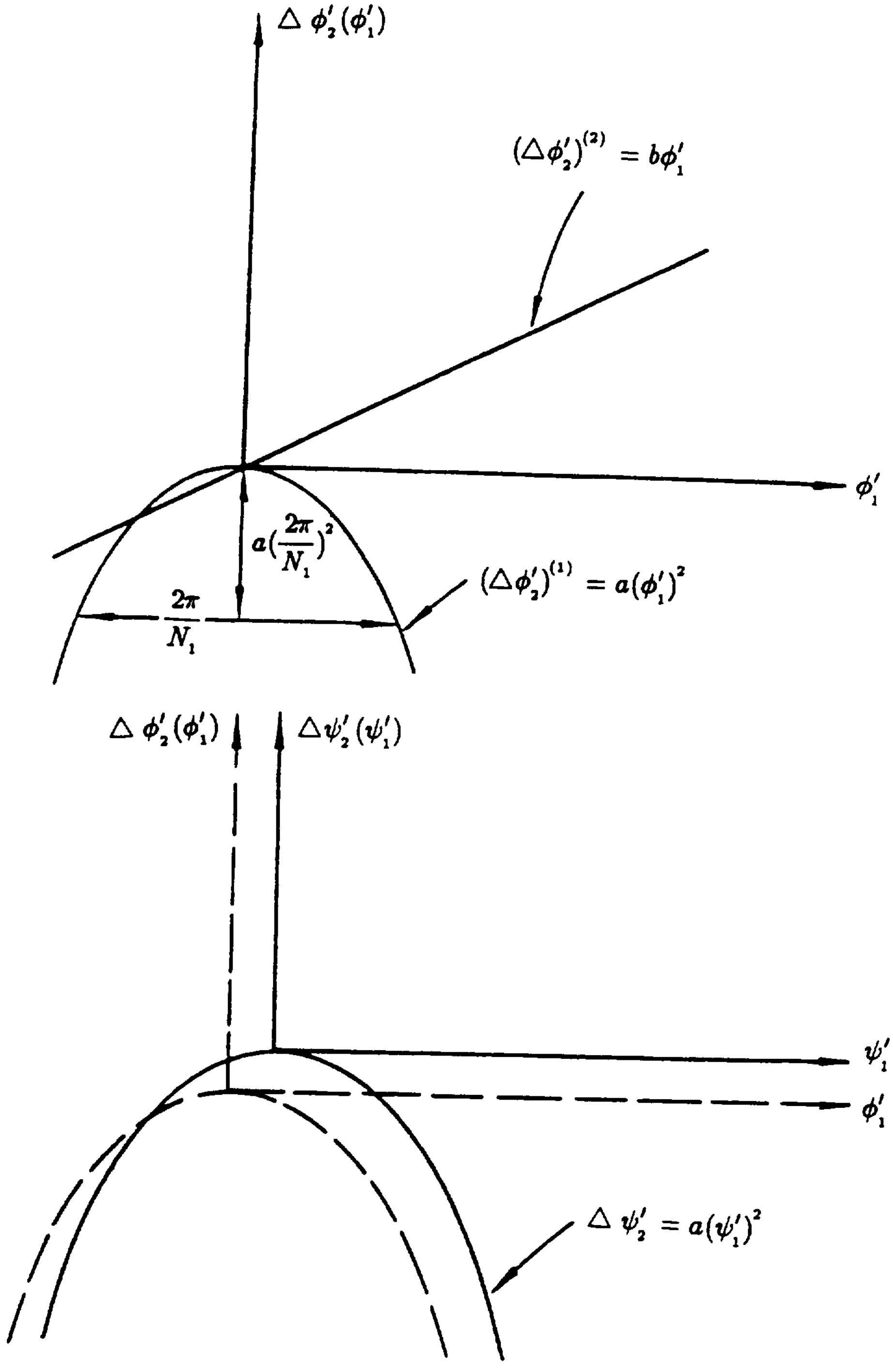


Figure 5: Interaction of parabolic and linear functions.

Misalignment changes the path of contact. The locations of transfer points are shifted to an edge. The amount of the shift is determined by $b/2a$. In general, the absolute value of b increases if the amount of misalignment increases. It is possible that an unfavorable ratio $b/2a$ may cause one of the transfer points to be off the tooth surface and that the function of transmission errors, $\triangle\psi_2'$, will become a discontinuous function for every cycle of meshing (Figure 6). To avoid this, the level of predesigned function of transmission error, or the absolute value of a, should be chosen with the expected level of transmission errors caused by misalignment.

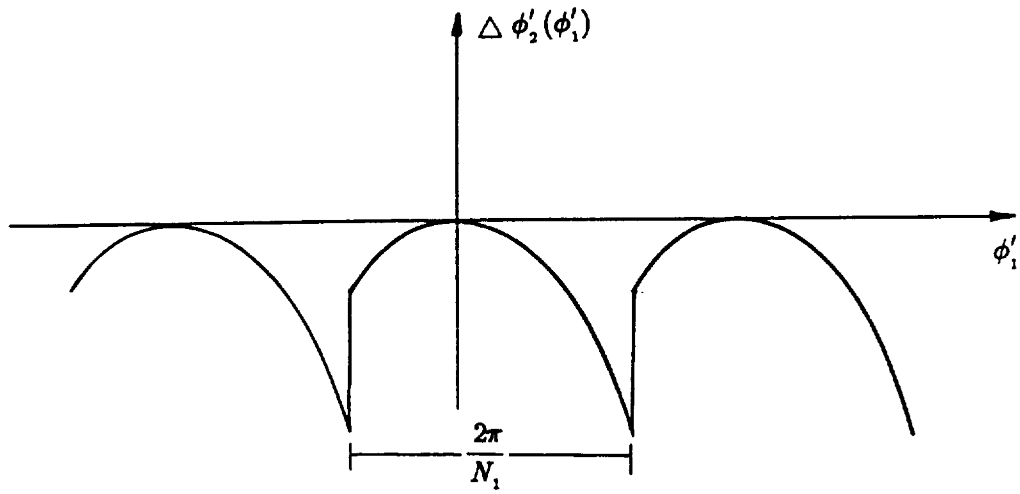


Figure 6: Discontinued parabolic function of transmission errors.

CHAPTER 2

GLEASON'S SPIRAL BEVEL GEARS

2.1 Gleason System

The Gleason Works, Rochester, New York, is one of the leading companies that produces equipment for manufacture of bevel and hypoid gears. William Gleason built the first machine in 1874 to cut bevel gears with straight teeth [6]. During the following years, the Gleason Works has developed a set of machines to generate spiral bevel gears. The basic construction (Figure 7) of a cutting machine consists of three major parts: the frame, the cradle, and the sliding base [7, 8].

When cutting starts, the work is plunged into the cutter. As the cutter rotates through the blank, a relative rolling motion is produced between the cradle and the work spindle to generate the tooth surface. While the cutter rolls out of engagement with the work, the cradle reverses rapidly, the sliding base on which the work is mounted is translated with respect to the cutter, and the work is indexed ahead for cutting the next tooth. This sequence of operations is repeated until the last tooth is cut.

In the process of cutting, the head-cutter rotates about its axis, and the axis generates in the cradle coordinate system a cylindrical surface. We may imagine that the cutter generates a tooth of crown gear as shown in Figure 8. Therefore, the cutting process corresponds to the motion of the gear rolling on a crown rack. The angular velocity of the head-cutter about its axis is not related with the generating motions and depends only on the desired velocity of cutting. This is

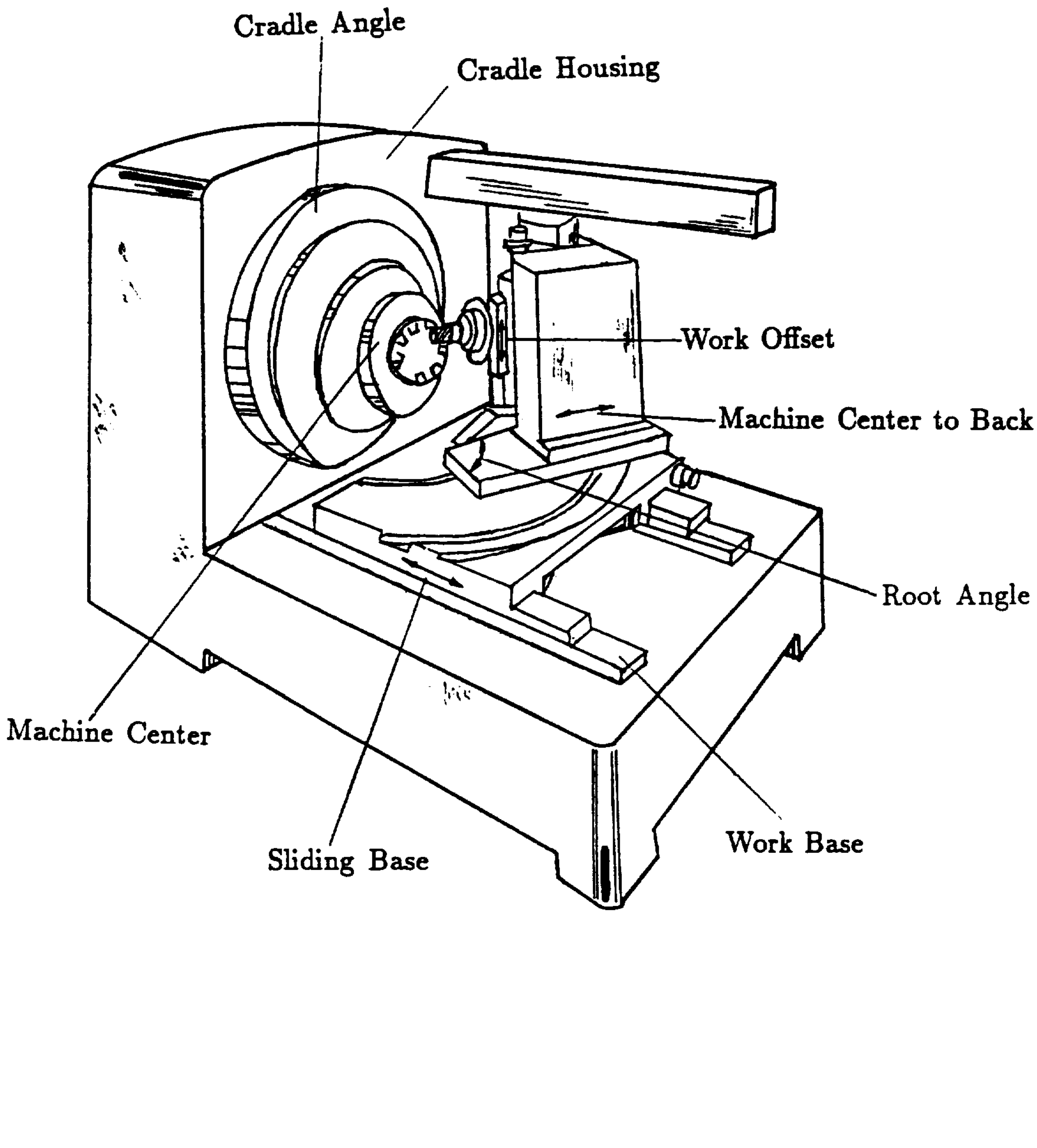


Figure 7: An isometric view for a gear generator.

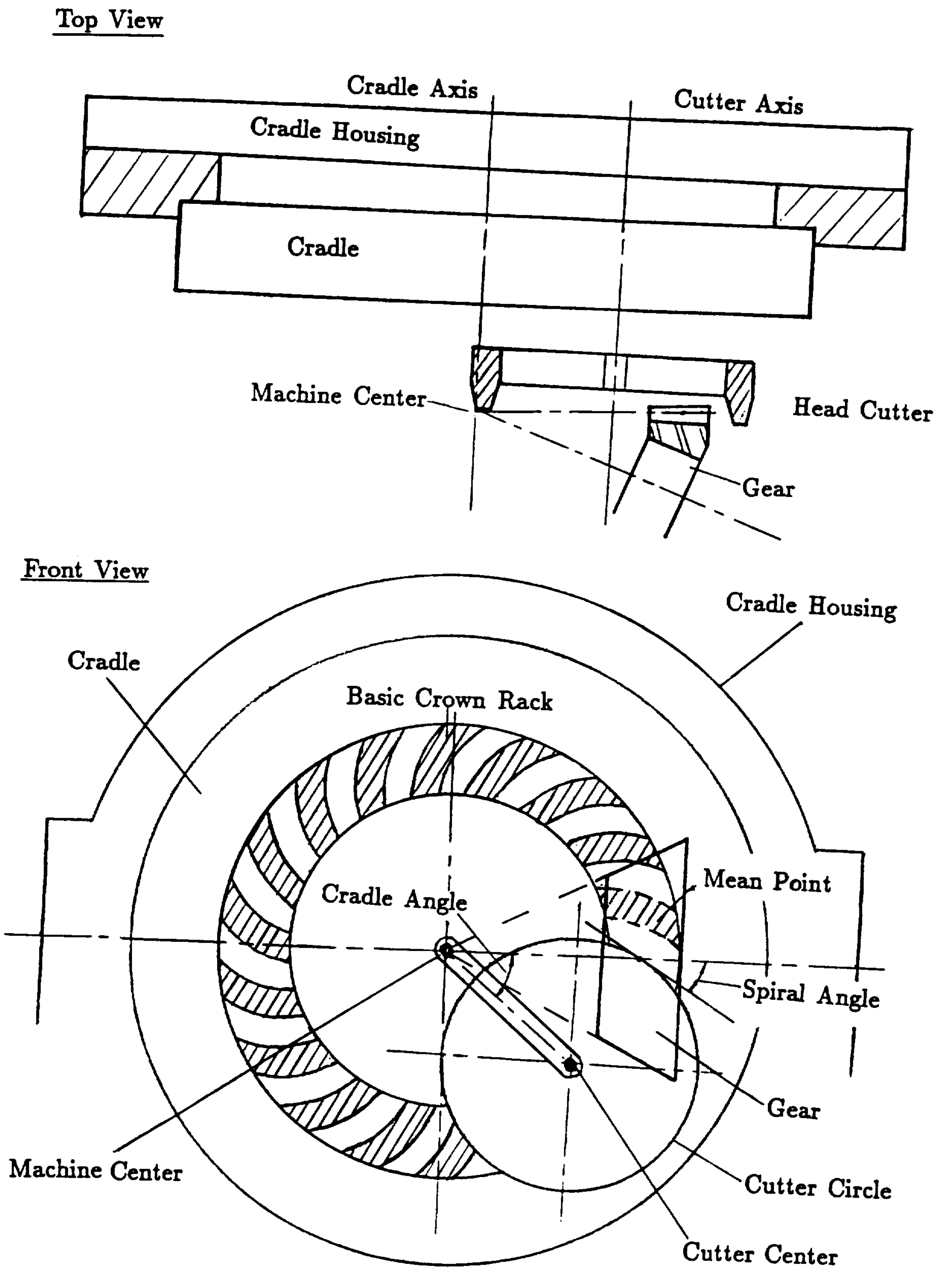


Figure 8: Cutting spiral gear teeth on the basic crown rack.

an important advantage of the Gleason methods of manufacture. Another advantage is that the same method for generation can be used as well for grinding. Grinding is essential for producing gears with hardened tooth surfaces and of high quality.

2.2 Head Cutters

Traditionally straight-sided blades have been applied in practice. The blades of the cutter generate cone surfaces while the cutter rotates its axis. Figure 9 shows these two cones. A current point B on the cone surface is represented in the coordinate system S_c as follows:

$$\vec{B}_c = \begin{bmatrix} B_{c_x} \\ B_{c_y} \\ B_{c_z} \\ 1 \end{bmatrix} = \begin{bmatrix} r\cot\psi - u\cos\psi \\ u\sin\psi\sin\theta \\ u\sin\psi\cos\theta \\ 1 \end{bmatrix} \tag{2.1}$$

where $u = \overline{AB}$ and θ are the surface coordinates, r is the tip radius of the cutter, and ψ is the blade angle. For the inside blade of the cutter, ψ is an acute angle. For the outside blade of the cutter, ψ is an obtuse angle.

Using equations (A.5) and (2.1) (provided $u\sin\psi \neq 0$), we obtain the equations of the unit normal to the cone surface.

$$\vec{n}_c = \begin{bmatrix} n_{c_x} \\ n_{c_y} \\ n_{c_z} \end{bmatrix} = \pm \begin{bmatrix} \sin\psi \\ \cos\psi\sin\theta \\ \cos\psi\cos\theta \end{bmatrix} \tag{2.2}$$

The total differential of vector $\vec{B}_c$ is

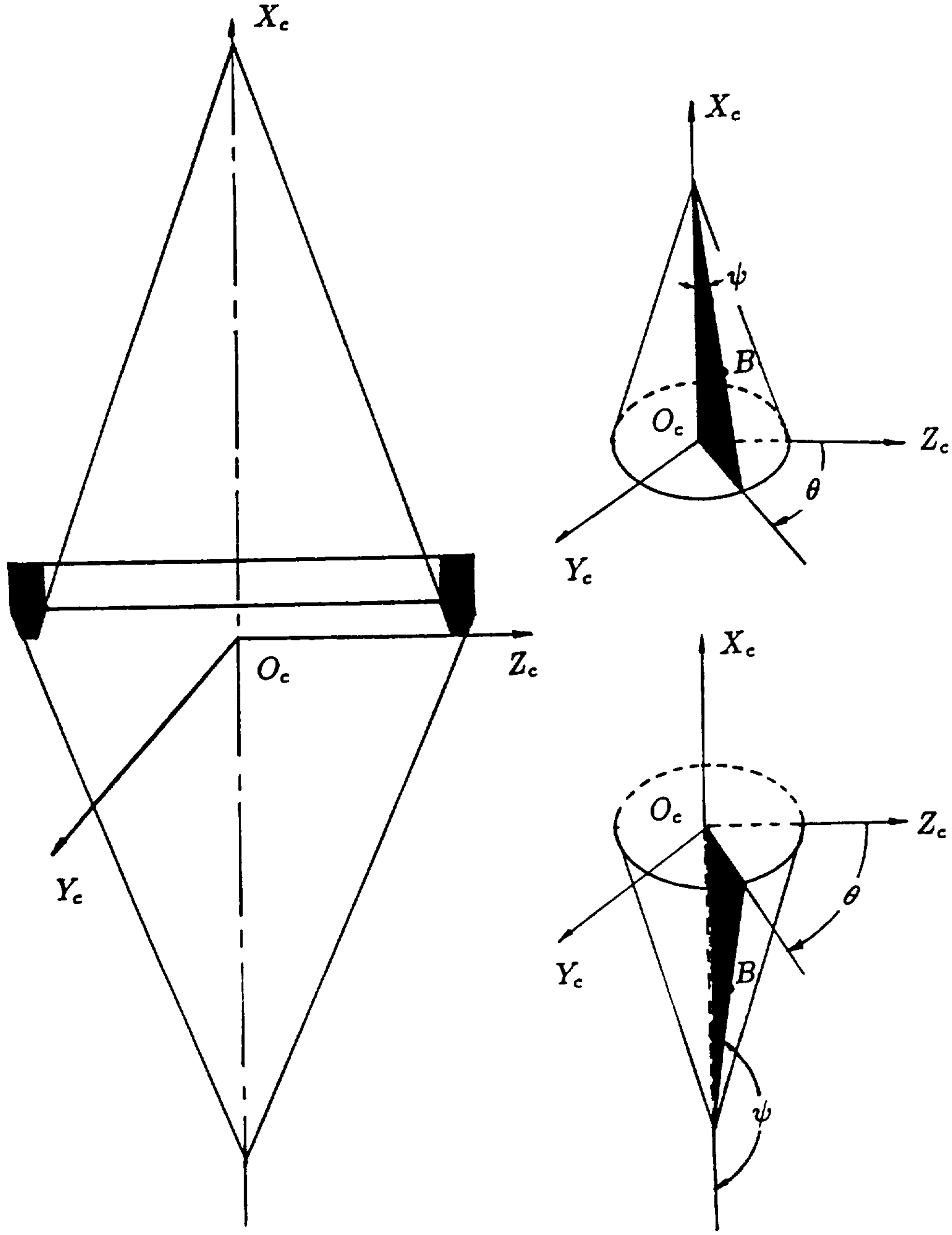


Figure 9: Generated cone surfaces of the head-cutter.

$$[dB_c] = \begin{bmatrix} -\cos\psi\, du \\ \sin\psi(\sin\theta\, du + u\cos\theta\, d\theta) \\ \sin\psi(\cos\theta\, du - u\sin\theta\, d\theta) \end{bmatrix} \tag{2.3}$$

The total differential of vector $\vec{n}_c$

$$[dn_c] = \pm \begin{bmatrix} 0 \\ \cos\psi\cos\theta\, d\theta \\ -\cos\psi\sin\theta\, d\theta \end{bmatrix} \tag{2.4}$$

Equations (A.26), (2.3), and (2.4) yield

$$\frac{0}{-\cos\psi\, du} = \frac{\pm\cos\psi\cos\theta\, d\theta}{\sin\psi(\sin\theta\, du + u\cos\theta\, d\theta)} = \frac{\mp\cos\psi\sin\theta\, d\theta}{\sin\psi(\cos\theta\, du - u\sin\theta\, d\theta)} = -\kappa_{I,II} \tag{2.5}$$

Equation (2.5) is satisfied if

$$du\, d\theta = 0 \tag{2.6}$$

One of the principal directions corresponds to $du = 0$; the other one to $d\theta = 0$. They can be represented by equations

$$\vec{e}_{I_c} = \frac{\dfrac{\partial \vec{B}_c}{\partial \theta}}{\left|\dfrac{\partial \vec{B}_c}{\partial \theta}\right|} \tag{2.7}$$

$$\vec{e}_{II_c} = \frac{\dfrac{\partial \vec{B}_c}{\partial u}}{\left|\dfrac{\partial \vec{B}_c}{\partial u}\right|} \tag{2.8}$$

Equations (2.3) and (2.7) yield

$$\vec{e}_{I_c} = \pm \begin{bmatrix} 0 \\ \cos\theta \\ -\sin\theta \end{bmatrix} \tag{2.9}$$

Plugging $du = 0$ into equations (2.5), we have

$$\kappa_{I} = \mp \frac{1}{u \tan\psi} \tag{2.10}$$

The sense of the principal curvature relies on the chosen direction of the normal.

Similarly, the unit vector of the second principal direction is

$$\vec{e}_{II_c} = \pm \begin{bmatrix} -\cos\psi \\ \sin\psi \sin\theta \\ \sin\psi \cos\theta \end{bmatrix} \tag{2.11}$$

The principal curvature is

$$\kappa_{II} = 0 \tag{2.12}$$

In addition to the cone surface, a tool provided by a surface of revolution is considered here. This surface of revolution is generated by an circular arc that rotates about the cutter axis. Such a surface can be applied as a grinding wheel or as a surface of a tool with curved blades.

Suppose the generating planar curve L (Figure 10) is an arc of a circle of radius R centered at point $(R_{c_x}, 0, R_{c_z})$. The spherical surface is generated by the circle in the rotational motion

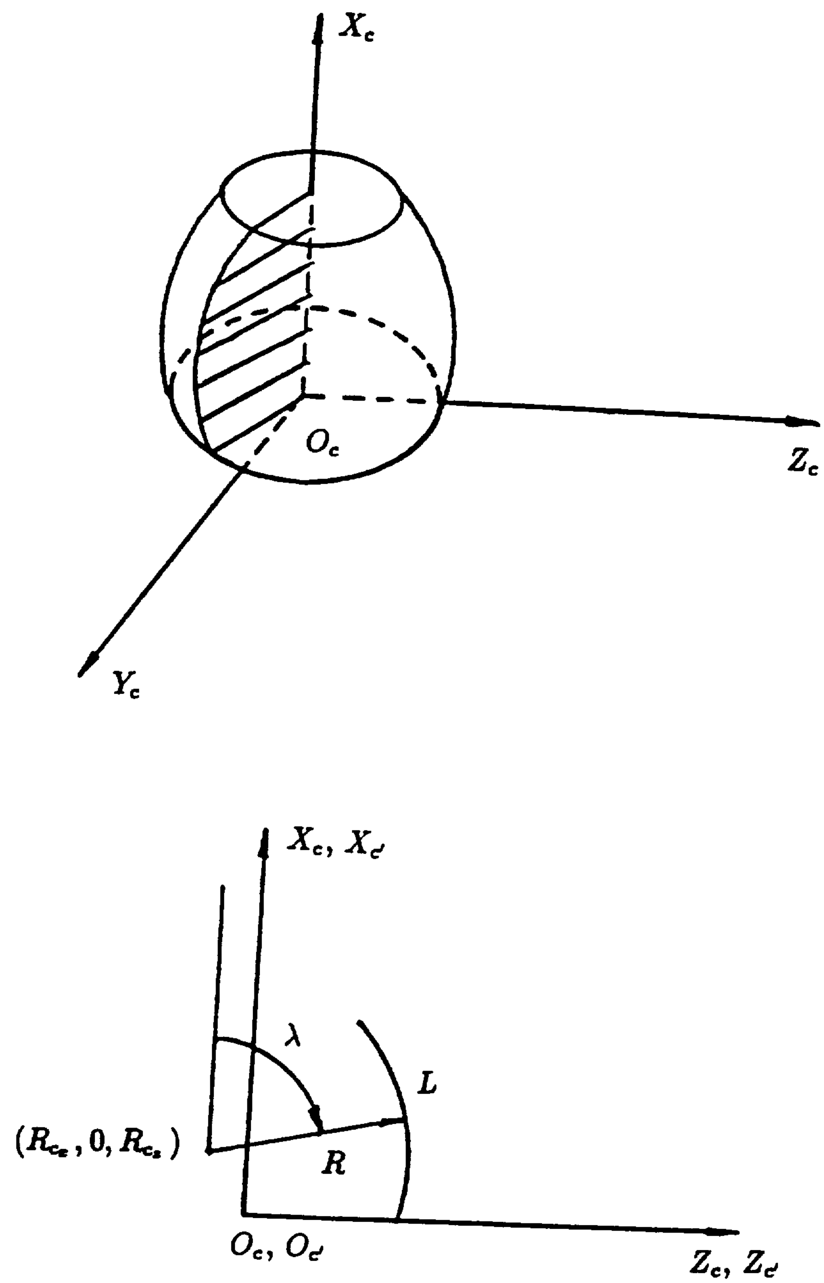


Figure 10: Generating arc circle for the curved edge of the head cutter.

about the Z_c-axis. Consider an auxiliary coordinate system $S_{c'}$ which is rigidly connected to the generating circle. Initially $S_{c'}$ and S_c coincide. The generating curve may be represented in the coordinate system $S_{c'}$ with the matrix equation

$$\begin{bmatrix} B_{c'_x} \\ B_{c'_y} \\ B_{c'_z} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{c_x} + R\cos\lambda \\ 0 \\ R_{c_z} + R\sin\lambda \\ 1 \end{bmatrix} \tag{2.13}$$

where λ is the varied parameter for planar curve L. The parameter λ lies within the following intervals:

$$\text{Inside blade} \begin{cases} 0 < \lambda < \pi/2, & \text{if } L \text{ is concave down;} \\ \pi < \lambda < 3\pi/2, & \text{if } L \text{ is concave up;} \end{cases}$$

$$\text{Outside blade} \begin{cases} 3\pi/2 < \lambda < 2\pi, & \text{if } L \text{ is concave down;} \\ \pi/2 < \lambda < \pi, & \text{if } L \text{ is concave up.} \end{cases}$$

The auxiliary coordinate system $S_{c'}$ rotates about the Z_c axis and the coordinate transformation from $S_{c'}$ to S_c is (Figure 11)

$$\vec{B}_c = \begin{bmatrix} B_{c_x} \\ B_{c_y} \\ B_{c_z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_{c'_x} \\ B_{c'_y} \\ B_{c'_z} \\ 1 \end{bmatrix} \tag{2.14}$$

Equations (2.13) and (2.14) yield

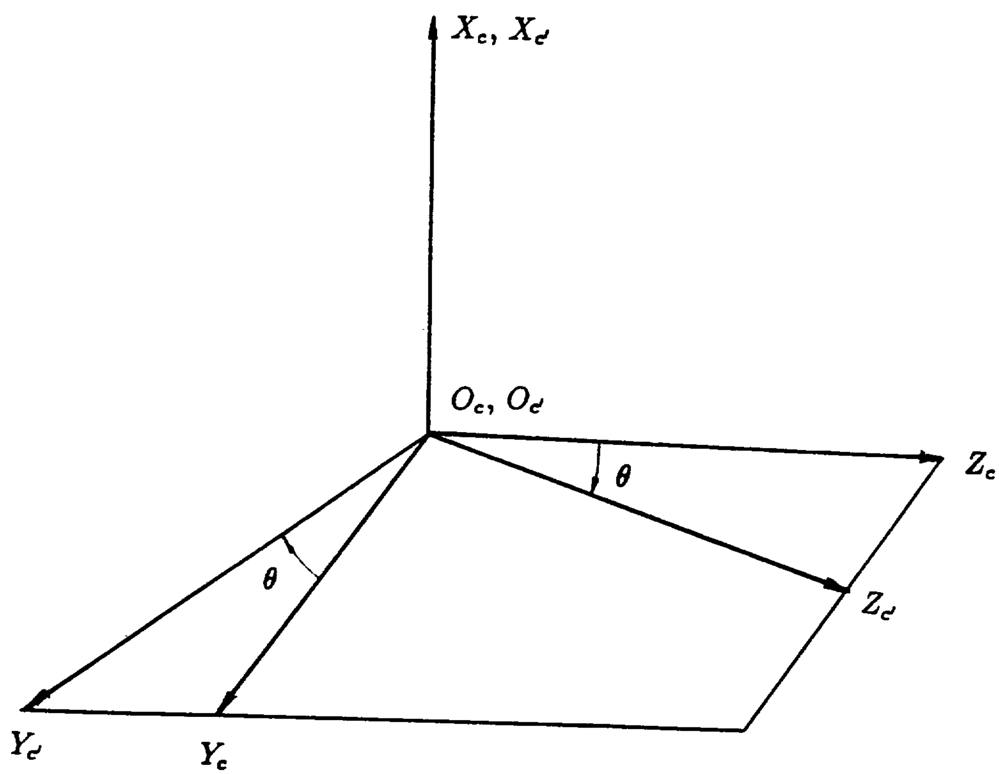


Figure 11: Coordinate transformations to generate spherical surfaces.

$$\vec{B}_c = \begin{bmatrix} B_{c_x} \\ B_{c_y} \\ B_{c_z} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{c_x} + R\cos\lambda \\ (R_{c_z} + R\sin\lambda)\sin\theta \\ (R_{c_z} + R\sin\lambda)\cos\theta \\ 1 \end{bmatrix} \tag{2.15}$$

Using equations (A.5) and (2.15), the unit normal to this spherical surface may be represented by

$$\vec{n}_c = \begin{bmatrix} n_{c_x} \\ n_{c_y} \\ n_{c_z} \end{bmatrix} = \pm \begin{bmatrix} \cos\lambda \\ \sin\lambda\sin\theta \\ \sin\lambda\cos\theta \end{bmatrix} \tag{2.16}$$

According to *Rodrigues' formula*, the principal directions on the generating surface correspond to $d\lambda = 0$ and $d\theta = 0$, respectively. The unit vector of the principal direction corresponding to $d\lambda = 0$ is

$$\vec{e}_{I_c} = \pm \begin{bmatrix} 0 \\ \cos\theta \\ -\sin\theta \end{bmatrix} \tag{2.17}$$

The principal curvature is

$$\kappa_I = \mp \frac{\sin\lambda}{R_{c_z} + R\sin\lambda} \tag{2.18}$$

The unit vector of the principal direction corresponding to $d\theta = 0$ is

$$\vec{e}_{II_c} = \pm \begin{bmatrix} -\sin\lambda \\ \cos\lambda\sin\theta \\ \cos\lambda\cos\theta \end{bmatrix} \tag{2.19}$$

The principal curvature is

$$\kappa_{II} = \mp \frac{1}{R} \tag{2.20}$$

2.3 Coordinate Systems and Sign Conventions

Left-hand gear-members are usually cut by the counterclockwise motion of the cradle that carries the head-cutter. This motion is viewed from the front of the cradle and from the back of the work spindle. Cutting is performed from the toe to the heel. Figure 12 shows the top and front views of the machine when a left-hand gear-member is cut.

Right-hand gear-members are usually cut by motions that are opposite to the motions of the left-hand members being cut. Cutting is performed from the heel to the toe. Figure 13 shows the top and front views of the machine for this case.

We set up five coordinate systems in either case. Coordinate system S_c is rigidly connected to the head cutter, coordinate system S_w is rigidly connected to the work, and coordinate systems S_m, S_p and S_a are rigidly connected to the frame. Axes Z_m and Z_p coincide with the root line and pitch line, respectively. Axis X_m is perpendicular to the generatrix of the root cone of the work. Axis X_p is perpendicular to the generatrix of the pitch cone of the work. Axes Z_a and Z_w coincide. Origin O_m is located at the machine center, and origins O_a and O_p are located at the apex of the pitch cone of the work.

Three special machine-tool settings, which are the machining offset, machine center to back, and the sliding base, are used only for the generation of pinions. The machining offset, denoted by E_m, is the shortest distance between the cradle axis and pinion axis. In figures 12 and 13, L_m represents a vector sum of machine center to back, X_{MCB}, and the sliding base, X_{SB}. The change

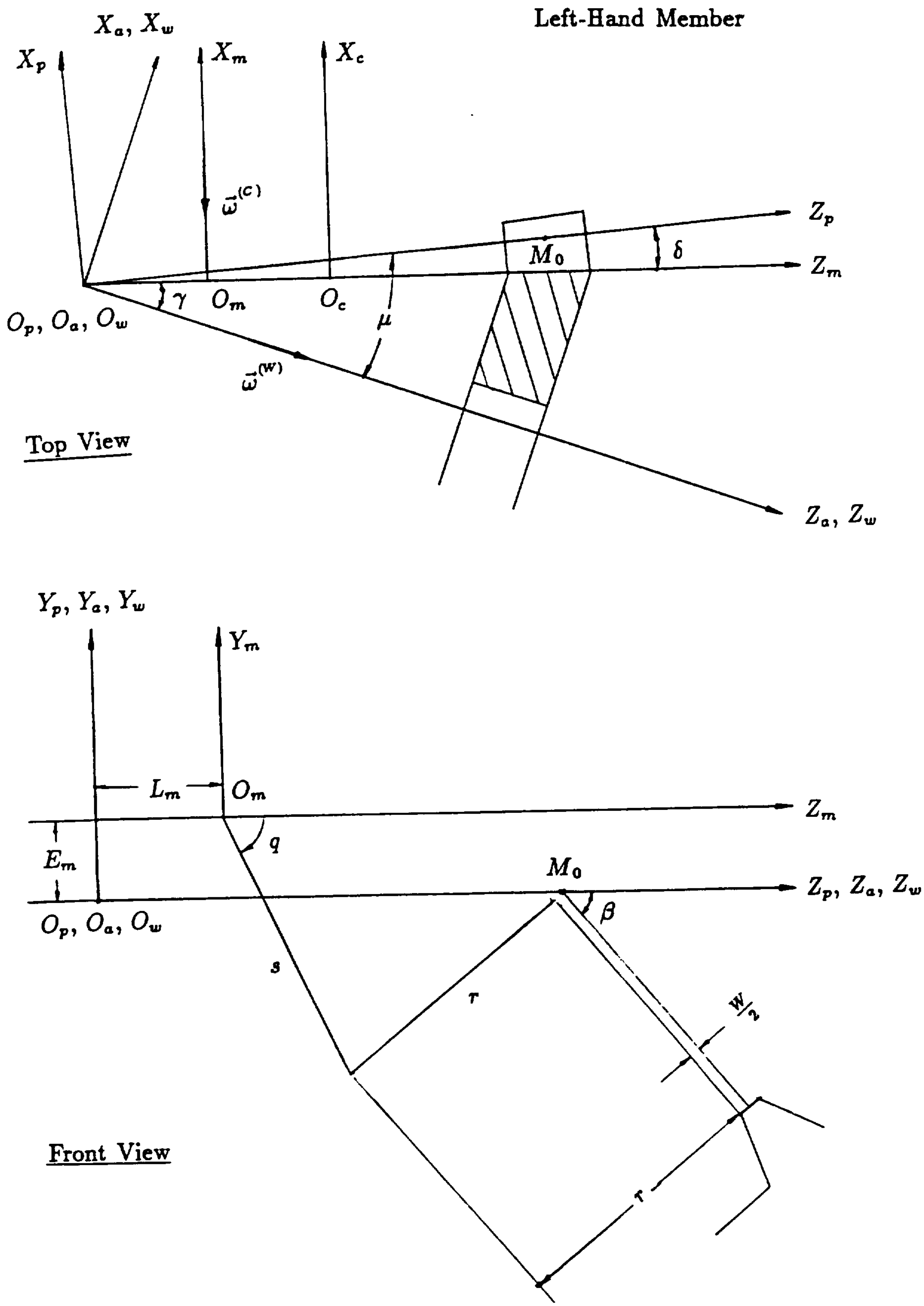
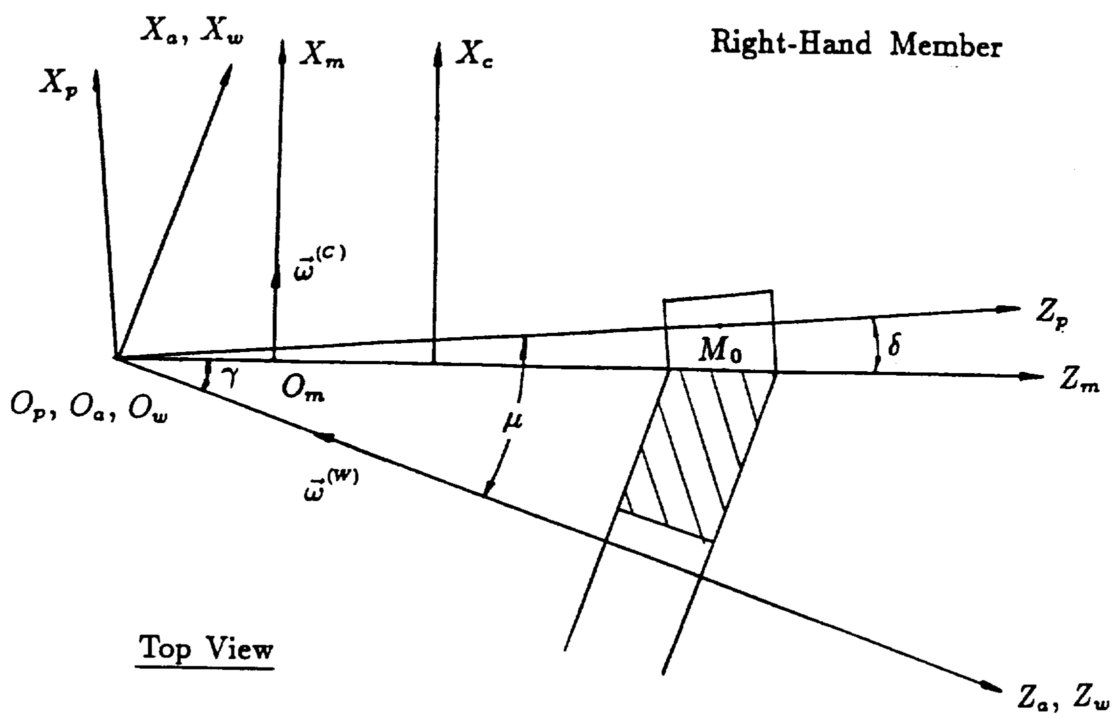


Figure 12: Top and front views of a left-hand gear generator.



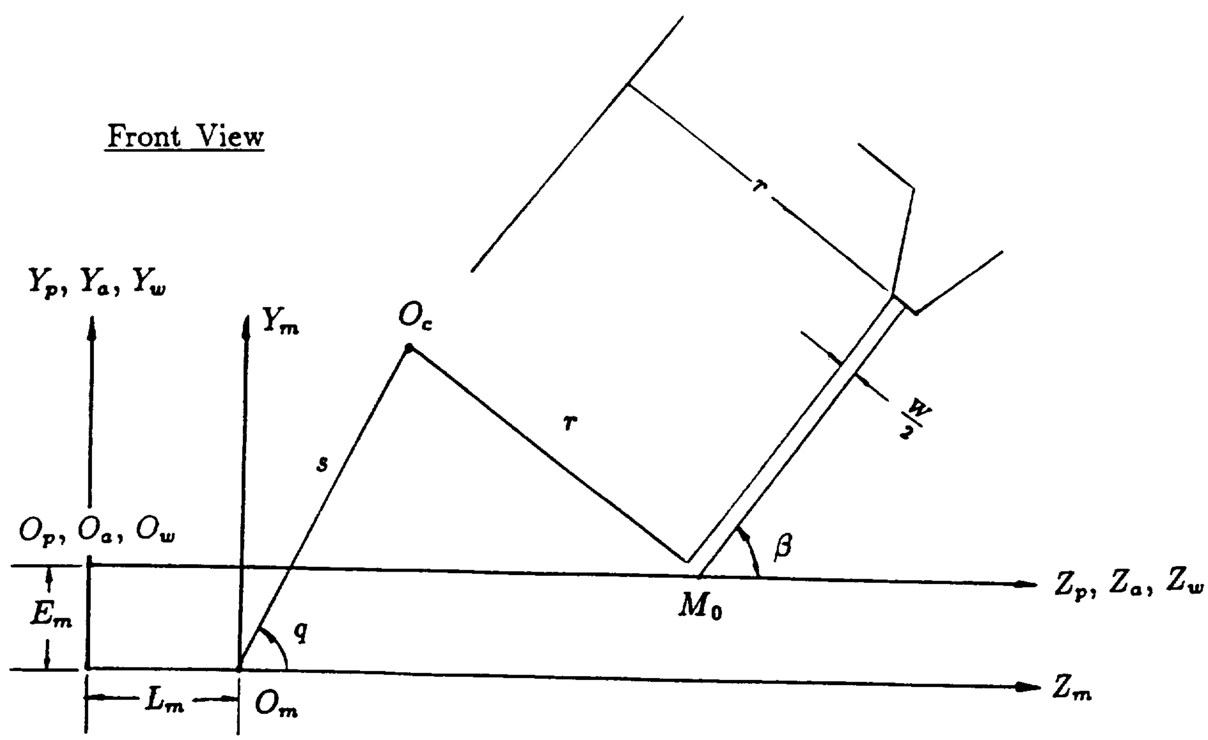


Figure 13: Top and front views of a right-hand gear generator.

TABLE 1: SIGN CONVENTIONS OF MACHINE-TOOL SETTINGS.

		Right-Hand Member	Left-Hand Member
Cradle Angle	+	counterclockwise (CCW)	clockwise (CW)
q	$-$	clockwise (CW)	counterclockwise (CCW)
Machining Offset	+	above machine center	below machine center
E_m	$-$	below machine center	above machine center
Machine Center to Back	+	work withdrawal	work withdrawal
X_{MCB}	$-$	work advance	work advance
Sliding Base	+	work withdrawal	work withdrawal
X_{SB}	$-$	work advance	work advance
L_m	+	X_{SB}: + and X_{MCB}: $-$	X_{SB}: + and X_{MCB}: $-$
	$-$	X_{SB}: $-$ and X_{MCB}: +	X_{SB}: $-$ and X_{MCB}: +

of machine center to back is directed parallel to the pinion axis and the direction of the sliding base is pointed parallel to the cradle axis.

The sign conventions for machine-tool settings are given in Table 1.

2.4 Generated Tooth Surfaces

The generated surface Σ_W is an envelope of the family of the tool surface Σ_C. Surfaces Σ_W and Σ_C contact each other at every instant along a line which is a spatial curve. Surface Σ_W is conjugate with Σ_C. In mathematical sense the determination of a conjugate surface is based on the theory of an envelope of a family of given surfaces. In differential geometry, to determine Σ_W we must find:

(a) the family of surfaces Σ_Φ generated by the given surface Σ_C in the S_w coordinate system and

(b) the envelope Σ_W of the family of surfaces Σ_Φ.

The matrix representation of the family of surfaces Σ_Φ may be represented by the matrix equation

$$[B_w] = [M_{wc}]\,[B_c] \tag{2.21}$$

where $[M_{wc}]$ is a matrix which describes the transformation of coordinates from the "old" coordinate system S_c to the "new" coordinate system S_w. From Figures 12 and 13, we obtain

$$[M_{wc}] = [M_{wa}]\,[M_{ap}]\,[M_{pm}]\,[M_{mc}] \tag{2.22}$$

We can obtain $[M_{wa}]$ from Figure 14 as

$$[M_{wa}] = \begin{bmatrix} \cos\phi_W & \pm\sin\phi_W & 0 & 0 \\ \mp\sin\phi_W & \cos\phi_W & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

where ϕ_W is the rotation angle of the work while it is being cut. Here the upper sign corresponds to the generation of a left-hand spiral bevel gear that is shown in Figure 12, and the lower sign corresponds to the generation of a right-hand spiral bevel gear shown in Figure 13. Henceforth we will obey this notation.

The transformation matrices $[M_{ap}]$ and $[M_{pm}]$ can be obtained from Figures 12 and 13 as

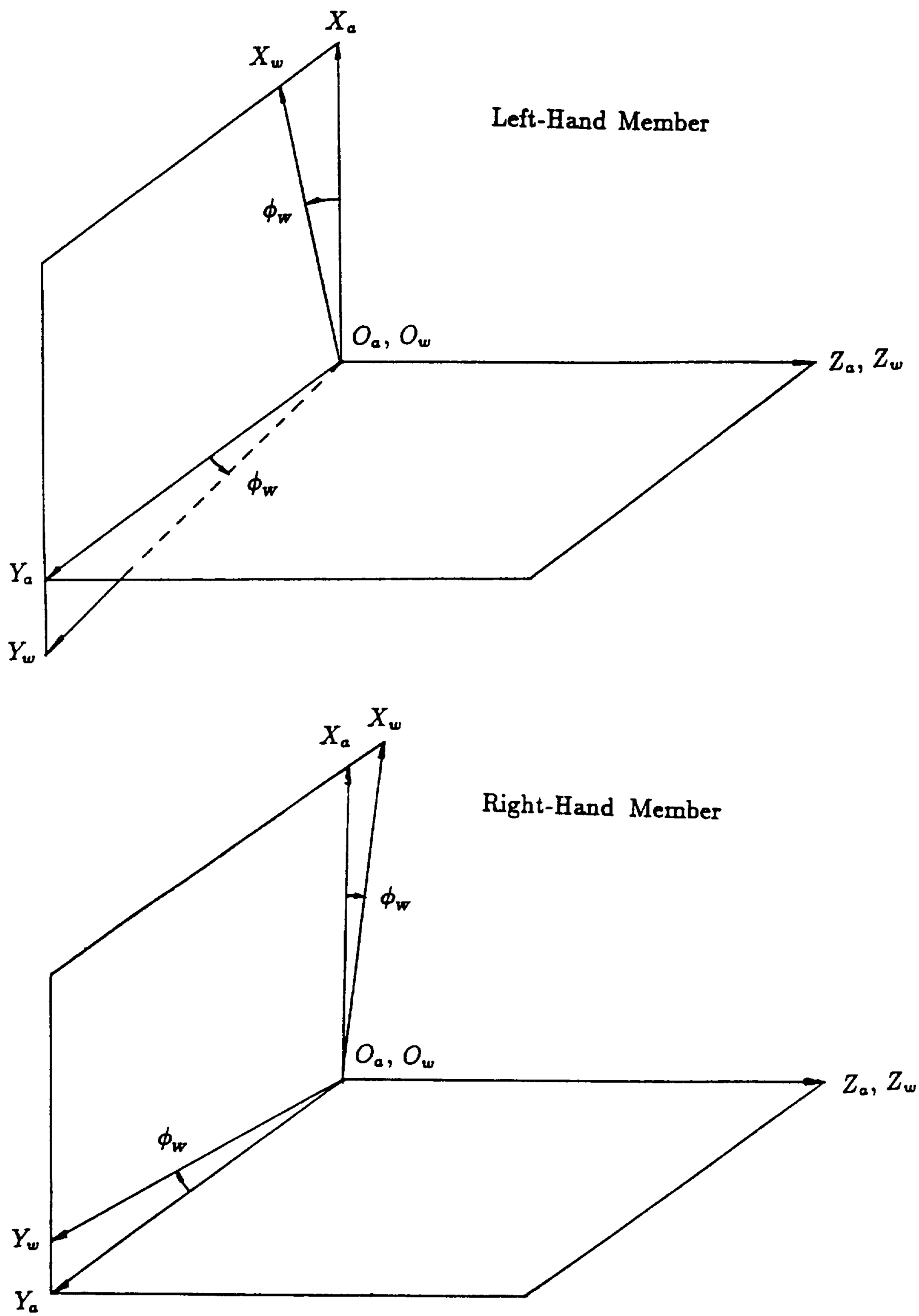


Figure 14: The rotation angle of the work while it is being cut.

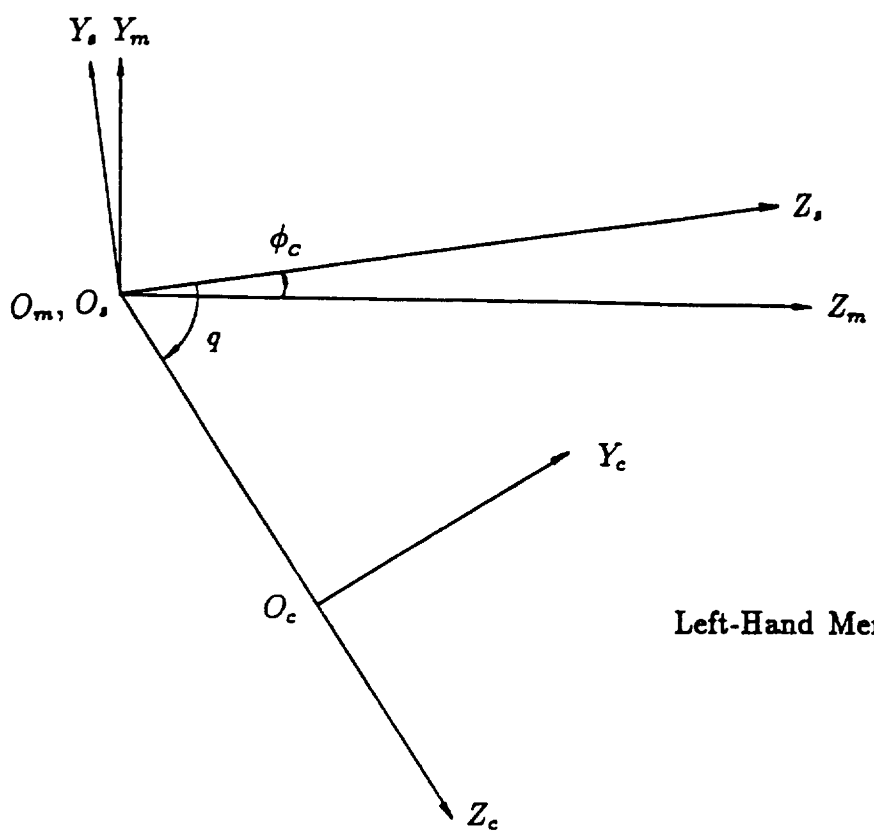
$$[M_{ap}] = \begin{bmatrix} \cos\mu & 0 & \sin\mu & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\mu & 0 & \cos\mu & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

$$[M_{pm}] = \begin{bmatrix} \cos\delta & 0 & -\sin\delta & -L_m \sin\delta \\ 0 & 1 & 0 & \pm E_m \\ \sin\delta & 0 & \cos\delta & L_m \cos\delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.25}$$

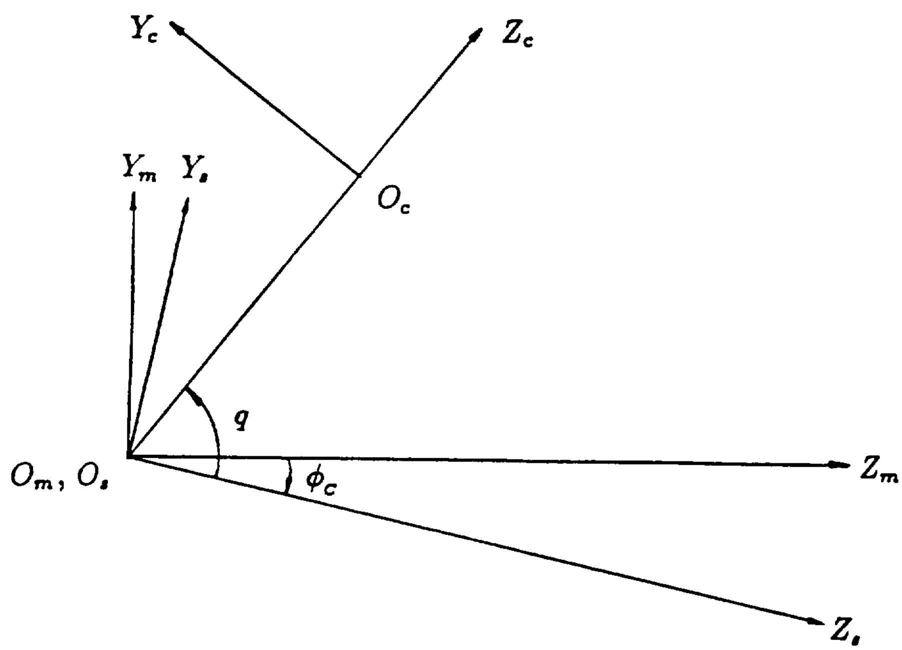
where μ and δ are the pitch angle and dedendum angle of the work, respectively. To derive the transformation matrix $[M_{mc}]$, let us apply an auxiliary coordinate system S_s rigidly connected to the tool (Figure 15). Thus

$$\begin{aligned} [M_{mc}] &= [M_{ms}][M_{sc}] \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi_C & \pm\sin\phi_C & 0 \\ 0 & \mp\sin\phi_C & \cos\phi_C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q & \mp\sin q & \mp s\sin q \\ 0 & \pm\sin q & \cos q & s\cos q \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.26}$$

where ϕ_C is the turn angle of the cradle while the work is cut, and s is the radial setting. The determination of the envelope Σ_W of the locus of surfaces Σ_Φ is based on necessary and sufficient conditions of envelope existence that have been developed in the classical Differential Geometry. A simpler method representation for determination of necessary condition of Σ_W existence is based



Left-Hand Member



Right-Hand Member

Figure 15: Auxiliary coordinate system S_s.

on the geometric property of conjugate surfaces: at points of tangency of the generating surface Σ_C and the generated surface Σ_W the common unit normal $\vec{n}$ to the surfaces is perpendicular to the slide velocity $\vec{V}^{(CW)}$ of these two surfaces [9, 10]. This is given by the scalar product

$$\vec{n} \cdot \vec{V}^{(CW)} = 0 \tag{2.27}$$

In the modern theory of gearing, equation (2.27) is called the equation of meshing. This equation is of fundamental importance in the kinematics of gearing. Since equation (2.27) is valid in any reference system, we will derive the equation of meshing in the S_m coordinate system. Let us designate ${}_{tr}\vec{V}_m^{(C)}$ and ${}_{tr}\vec{V}_m^{(W)}$ the transfer velocities of common contact points B_m on the cutter and the work, respectively. Thus

$$\vec{V}_m^{(CW)} = {}_{tr}\vec{V}_m^{(C)} - {}_{tr}\vec{V}_m^{(W)} \tag{2.28}$$

The cradle rotates about the Z_m axis with the angular velocity $\vec{\omega}_m^{(C)}$ (Figures 12 and 13); therefore, the transfer velocity ${}_{tr}\vec{V}_m^{(C)}$ is represented by the equation

$${}_{tr}\vec{V}_m^{(C)} = \vec{\omega}_m^{(C)} \times \vec{B}_m \tag{2.29}$$

The work rotates about the Z_a axis with the angular velocity $\vec{\omega}_m^{(W)}$ (Figures 12 and 13) which does not pass through the origin O_m of the S_m coordinate system. It is known from the theoretical mechanics that the angular velocity $\vec{\omega}_m^{(W)}$ may be substituted by an equal vector $\vec{\omega}_m^{(W)}$ which passes through O_m and a vector-moment

$$\overline{O_m O_a} \times \vec{\omega}_m^{(W)} \tag{2.30}$$

Note that the moment has the same unit and physical meaning as linear velocity. Thus

$$_{tr}\vec{V}_m^{(W)} = \vec{\omega}_m^{(W)} \times \vec{B}_m + \overline{O_m O_a} \times \vec{\omega}_m^{(W)} \tag{2.31}$$

It is evident from Figures 12 and 13 that

$$\left[\omega_m^{(C)}\right] = \mp\omega^{(C)} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.32}$$

$$\left[\omega_m^{(W)}\right] = \pm\omega^{(W)} \begin{bmatrix} -\sin\gamma \\ 0 \\ \cos\gamma \end{bmatrix} \tag{2.33}$$

$$\overline{O_m O_a} = \begin{bmatrix} 0 \\ \mp E_m \\ -L_m \end{bmatrix} \tag{2.34}$$

In equation (2.33) γ is the root angle of the work. Substituting equation (2.29)–(2.34) into equation (2.28), we obtain

$$\vec{V}_m^{(CW)} = \begin{bmatrix} \omega^{(W)}(E_m \pm B_{m_y})\cos\gamma \\ \pm\omega^{(C)} B_{m_z} + \omega^{(W)}[(B_{m_z} \mp L_m)\sin\gamma \mp B_{m_x}\cos\gamma] \\ \mp\omega^{(C)} B_{m_y} \pm \omega^{(W)}(B_{m_y} - E_m)\sin\gamma \end{bmatrix} \tag{2.35}$$

The coordinates of the common contact points B_m may be obtained from equations of the generating surface Σ_C. Then we get

$$[B_m] = [M_{mc}]\,[B_c] \tag{2.36}$$

The common unit normals $\vec{n}_m$ may be represented by the unit normals to Σ_C. Therefore

$$[n_m] = [L_{mc}]\,[n_c] \tag{2.37}$$

where $[L_{mc}]$ is the rotation matrix obtained by eliminating of the last row and last column of the corresponding matrix $[M_{mc}]$.

Hence, if Σ_C is a cone surface, substituting equations (2.1) and (2.26) into equation (2.36) we obtain

$$\begin{bmatrix} B_{m_x} \\ B_{m_y} \\ B_{m_z} \\ 1 \end{bmatrix} = \begin{bmatrix} r\cot\psi - u\cos\psi \\ u\sin\psi\sin\tau \mp s\sin(q - \phi_C) \\ u\sin\psi\cos\tau + s\cos(q - \phi_C) \\ 1 \end{bmatrix} \tag{2.38}$$

where $\tau = \theta \mp q \pm \phi_C$. Substituting equations (2.2) and (2.26) into equation (2.37) the unit normals may be represented as

$$\begin{bmatrix} n_{m_x} \\ n_{m_y} \\ n_{m_z} \end{bmatrix} = \pm \begin{bmatrix} \sin\psi \\ \cos\psi\sin\tau \\ \cos\psi\cos\tau \end{bmatrix} \tag{2.39}$$

Similarly if Σ_C is a spherical surface, substituting equations (2.15) and (2.26) into equation (2.36), we obtain

$$
\begin{bmatrix} B_{m_x} \\ B_{m_y} \\ B_{m_z} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{c_x} + R\cos\lambda \\ (R_{c_z} + R\sin\lambda)\sin\tau \mp s\sin(q - \phi_C) \\ (R_{c_z} + R\sin\lambda)\cos\tau + s\cos(q - \phi_C) \\ 1 \end{bmatrix} \tag{2.40}
$$

Substituting equations (2.16) and (2.26) into equation (2.37) the unit normals may be represented as

$$
\begin{bmatrix} n_{m_x} \\ n_{m_y} \\ n_{m_z} \end{bmatrix} = \pm \begin{bmatrix} \cos\lambda \\ \sin\lambda\sin\tau \\ \sin\lambda\cos\tau \end{bmatrix} \tag{2.41}
$$

Designate

$$
m_{CW} = \frac{\omega^{(C)}}{\omega^{(W)}} \tag{2.42}
$$

Using equations (2.27), (2.35), (2.38), and (2.39), we may obtain the equation of meshing for the case that Σ_C is a cone surface by

$$
\begin{gathered}
(u - r\cot\psi\cos\psi)\cos\gamma\sin\tau + s[(m_{CW} - \sin\gamma)\cos\psi\sin\theta \mp \cos\gamma\sin\psi\sin(q - \phi_C)] \\
\pm E_m(\cos\gamma\sin\psi + \sin\gamma\cos\psi\cos\tau) - L_m\sin\gamma\cos\psi\sin\tau = 0
\end{gathered} \tag{2.43}
$$

For Σ_C being a spherical surface, using equations (2.27), (2.35), (2.40), and (2.41), the equation of meshing is represented by

$$
\begin{gathered}
(R_{c_z}\cos\lambda - R_{c_x}\sin\lambda)\cos\gamma\sin\tau + s[(m_{CW} - \sin\gamma)\sin\lambda\sin\theta \mp \cos\gamma\cos\lambda\sin(q - \phi_C)] \\
\pm E_m(\cos\gamma\cos\lambda + \sin\gamma\sin\lambda\cos\tau) - L_m\sin\gamma\sin\lambda\sin\tau = 0
\end{gathered} \tag{2.44}
$$

Equations (2.43) and (2.44) relate the generating surface coordinates (u and θ for a cone surface or λ and θ for a surface of revolution) with the turn angle ϕ_C.

CHAPTER 3

SYNTHESIS OF SPIRAL BEVEL GEARS

3.1 Gear Machine-Tool Settings

We designate for the following discussions the gear-generating tool surface by Σ_G, the generated gear surface by Σ_2, the pinion-generating tool surface by Σ_P, and the generated pinion surface Σ_1. A parameter with the subscript i indicates that it is related to surface Σ_i. To set up the gear machine-tool settings, the following data are considered as given:

Γ:	shaft angle
N_2:	gear tooth number
N_1:	pinion tooth number
γ_2:	gear root angle
A:	mean pitch cone distance
β:	mean spiral angle
ψ_G:	blade angle for gear cutter
d_G:	average diameter of gear cutter
w_G:	point width

3.1.1 Preliminary Considerations

We prefer to calculate the values of pitch angles and dedendum angles rather than obtain them from the blank design summary because the data in the summary are not accurate enough for computer calculations.

The gear pitch angle is represented by

$$\mu_2 = \arctan \frac{\sin\Gamma}{\frac{N_1}{N_2} + \cos\Gamma} \tag{3.1}$$

The pinion pitch angle is

$$\mu_1 = \Gamma - \mu_2 \tag{3.2}$$

The dedendum angles are

$$\delta_1 = \mu_1 - \gamma_1, \qquad \delta_2 = \mu_2 - \gamma_2 \tag{3.3}$$

3.1.2 Gear Cutting Ratio

The process of gear generation is based on the imaginery meshing of a crown gear with the member-gear. The instantaneous axis of rotation by such meshing coincides with the pitch line, axis Z_p, that is shown in Figures 12 and 13. The generating surface Σ_G, which may be imagined as the surface of the crown gear, and the to be generated gear surface Σ_2 contact each other at a line at every instant. The ratio of angular velocities of the crown gear and the being generated gear (the cutting ratio) remains constant while the spatial line of contact moves over surfaces Σ_G and Σ_2. The determination of cutting ratio is based on following consideration. The angular velocity in relative motion is

$$\vec{\omega}^{(G2)} = \vec{\omega}^{(G)} - \vec{\omega}^{(2)} = a\vec{k}_p \tag{3.4}$$

This means that vectors $\vec{\omega}^{(G2)}$ and $\vec{k}_p$ are collinear. Since equation (3.4) is valid in any reference frame, let us derive it in the S_m coordinate system. From Figures 12 and 13 we have

$$\vec{k}_p = \begin{bmatrix} \sin\delta_2 \\ 0 \\ \cos\delta_2 \end{bmatrix} \tag{3.5}$$

By replacing the superscript 'c' by 'G' in equation (2.32) and 'w' by '2' in equation (2.33), we may represent in matrix from angular velocities $\vec{\omega}^{(G)}$ and $\vec{\omega}^{(2)}$. Consequently, we obtain the following equation

$$\frac{\mp\omega^{(G)} \pm \omega^{(2)}\sin\gamma_2}{\sin\delta_2} = \frac{\mp\omega^{(2)}\cos\gamma_2}{\cos\delta_2} \tag{3.6}$$

Equation (3.6) results in that

$$m_{G2} = \frac{\omega^{(G)}}{\omega^{(2)}} = \frac{\sin\mu_2}{\cos\delta_2} \tag{3.7}$$

3.1.3 Cutter Tip Radius, Radial Setting, and Cradle Angle

Figure 16 shows that the inside and outside tip radii of the head-cutter are represented by

$$r_G = \frac{1}{2}(d_G \mp w_G) \tag{3.8}$$

Figure 16 shows the front view of the installation of the head cutter. From the relations between the lengths and angles of the triangle $O_m O_c M_o$, we may express the radial setting s_G and cradle angle q_G as follows:

$$s_G = \sqrt{\frac{d_G^2}{4} + A^2 \cos^2 \delta_2 - d_G A \cos \delta_2 \sin \beta} \tag{3.9}$$

and

$$q_G = \arccos \frac{A^2 \cos^2 \delta_2 + s_G^2 - \frac{d_G^2}{4}}{2 A s_G \cos \delta_2} \tag{3.10}$$

3.2 Determination of the Mean Contact Point on the Gear Tooth Surfaces

The gear and pinion surfaces of spiral bevel gears are in point contact at every instant. The mean contact point is the center of the bearing contact and its location is selected generally at the middle of the working depth on the gear tooth. Figure 17 shows a gear tooth surface. Section $\overline{AD}$ is the gear tip and it is parallel to the generatrix of the root cone of the pinion. Section $\overline{BC}$ is the pinion tip and it is parallel to the root line of the gear. The working area is within $\square ABCD$. In the S_p coordinate system, line $\overline{AD}$ may be represented by

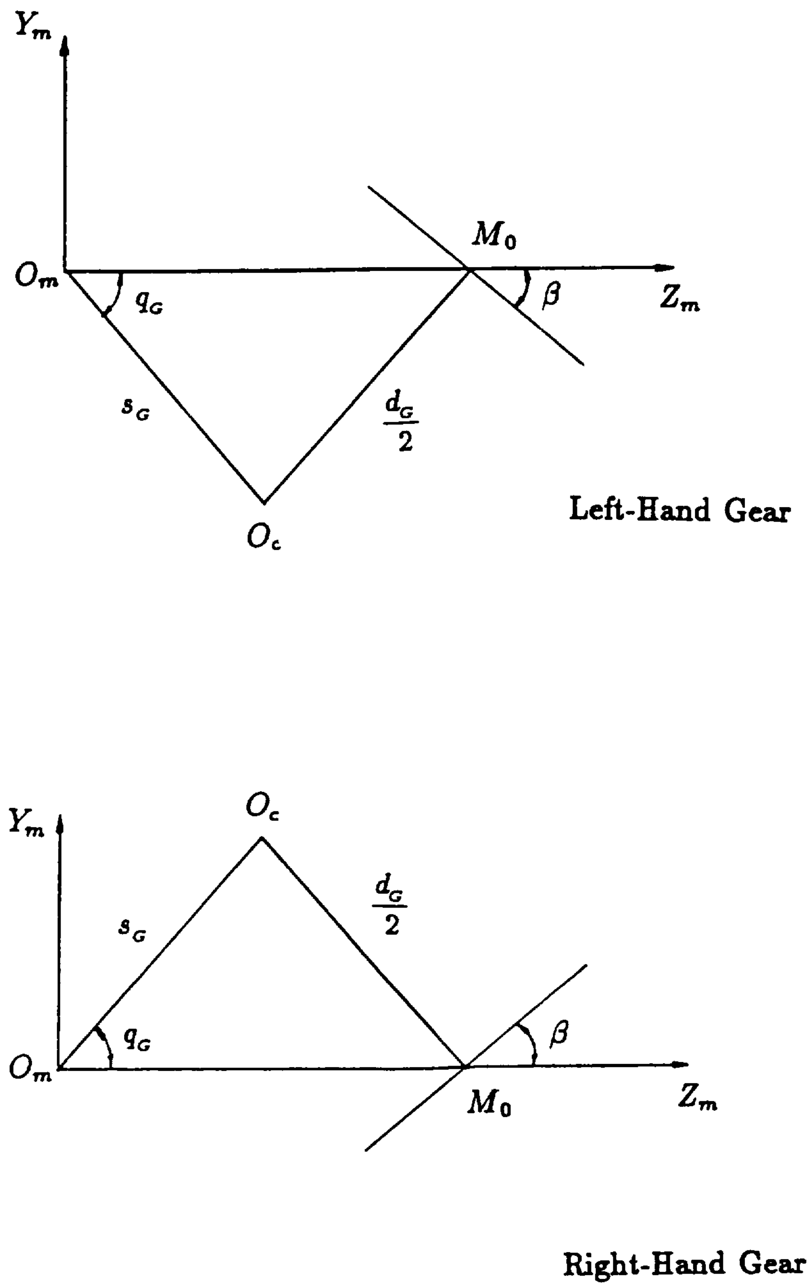


Figure 16: The front view of the installation of the head cutter

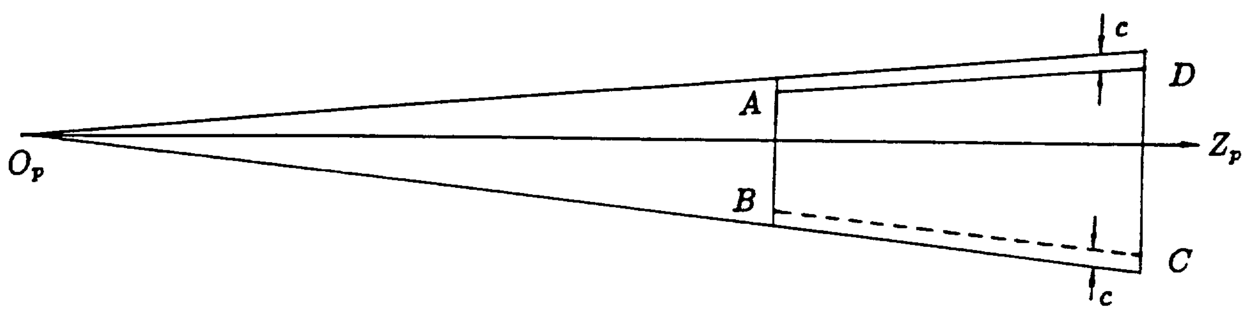


Figure 17: Gear tooth surface.

$$B_{p_x} = B_{p_z} \tan \delta_1 - c \tag{3.11}$$

where c is the clearance and δ_1 is the pinion dedendum angle. Line $\overline{BC}$ is represented by

$$B_{p_x} = -B_{p_z} \tan \delta_2 + c \tag{3.12}$$

The mean contact point is located on a line which passes through the middle point of the two points at which the normal section of the gear surface intersects line $\overline{AD}$ and line $\overline{BC}$, respectively. In addition, the mean contact point must be on the gear surface. This means that it must satisfy the equation of meshing for the gear being generated by the tool. We will use these two requirements to determine the location of the mean contact point and represent the procedure of derivations as follows

STEP 1: The initial guess for θ_G is

$$\theta_G = \pm(q_G - \beta + \pi/2)$$

STEP 2: Determination of u_G based on the given θ_G

Equation (2.43) determines parameter u_G. The turn angle ϕ_G is set to zero when equation (2.43) is applied.

STEP 3: Representation of gear tooth surface in coordinate systems S_c and S_p

Equation (2.1) determines the gear tooth surface in coordinate system S_c. The gear tooth surface may be represented in the S_p coordinate system as follows:

$$[B_p] = [M_{pm}][M_{mc}][B_c] \tag{3.13}$$

Transformation matrix $[M_{pm}]$ may be obtained from equation (2.25) by setting E_m and L_m to zero. Equation (2.26) determines matrix $[M_{mc}]$.

STEP 4: Determination of middle point

The x coordinate of the middle point M of the two points, which are the intersections of the normal section of the gear tooth surface and the gear tooth tips, may be obtained by

$$M_{p_x} = \frac{B_{p_z}(\tan\delta_1 - \tan\delta_2)}{2} \tag{3.14}$$

The above equation is derived by dividing the sum of equations (3.11) and (3.12) by 2.

STEP 5: Judgement of u_G

The acceptable value of u_G is determined by the following criterion:

$$|B_{p_x} - M_{p_x}| < \epsilon$$

where ϵ is a specified tolerance value. If the above criterion is satisfied, parameters u_G and θ_G of the mean contact point are determined. Otherwise, repeat STEP 2 to STEP 5 by a new value of θ_G until the criterion is satisfied.

As a matter of fact, the determination of the location of the mean contact point is the same as that of a root of equation

$$B_{p_x} - M_{p_x} = 0 \tag{3.15}$$

The new value of the θ_G in STEP 5 depends on which method is used to solve this equation. In this study *Newton's method* was used.

So far we have already determined parameters u_G and θ_G of the mean contact point. Repeating the task done in STEP 3, we have the coordinates of the mean contact point B. The common unit normal $\vec{n}$ to surfaces Σ_G and Σ_2 at the mean contact point B is

$$[n_p] = [L_{pm}]\,[L_{mc}]\,[n_c] \tag{3.16}$$

where matrix $[L_{pm}]$ is obtained by deleting the fourth row and column from matrix $[M_{pm}]$ given by equation (2.25). Similarly, we may obtain rotation matrix $[L_{mc}]$ from matrix $[M_{mc}]$ by equation (2.26). Although the unit normal has two directions, we choose the direction corresponding to the positive sign in equation (2.2) regardless of the hand of the gear. The principal directions at the mean contact point B on the gear tool surface Σ_G in the S_p coordinate system may be obtained by the following coordinate transformation:

$$\left[e_{G_{I,II_p}}\right] = [L_{pm}]\,[L_{mc}]\left[e_{G_{I,II_c}}\right] \tag{3.17}$$

Here we choose positive sign in equation (2.9) as the direction of the first principal direction. The second principal direction is determined by rotating of the first principal direction about unit normal by 90°.

The principal curvatures and directions at the mean contact point B on the gear surface Σ_2 may be derived according to the formula expressed in Section A.2. Note that surfaces Σ_G and Σ_2 are in line contact. To apply these formula, we may consider that surfaces Σ_2 and Σ_G are equivalent to surfaces $\Sigma_{\mathcal{F}}$ and Σ_Q, respectively, in Section A.2.

The derivation of the principal curvatures and directions at the mean contact point B is performed as follows:

STEP 1: We represent the angular velocity $\vec{\omega}^{(2)}$ in the S_p coordinate system as follows:

$$\left[\omega_p^{(2)}\right] = \pm\omega^{(2)} \begin{bmatrix} -\sin\mu_2 \\ 0 \\ \cos\mu_2 \end{bmatrix} \tag{3.18}$$

This is a direct result from drawings of Figures 12 and 13.

STEP 2: We represent the angular velocity $\vec{\omega}^{(G)}$ in the S_p coordinate system as follows:

$$\left[\omega_p^{(G)}\right] = \mp\omega^{(G)} \begin{bmatrix} \cos\delta_2 \\ 0 \\ \sin\delta_2 \end{bmatrix} \tag{3.19}$$

This is also a direct result from Figures 12 and 13.

STEP 3: The relative angular velocity $\vec{\omega}_p^{(2G)}$ is represented by

$$\vec{\omega}_p^{(2G)} = \vec{\omega}_p^{(2)} - \vec{\omega}_p^{(G)} = \pm\omega^{(2)} \begin{bmatrix} -\sin\mu_2 + m_{G2}\cos\delta_2 \\ 0 \\ \cos\mu_2 + m_{G2}\sin\delta_2 \end{bmatrix} \tag{3.20}$$

STEP 4: The transfer velocity of the mean point B on surface Σ_G is

$$_{tr}\vec{V}_p^{(G)} = \vec{\omega}_p^{(G)} \times \vec{B}_p \tag{3.21}$$

STEP 5: The transfer velocity of the mean point B on surface Σ_2 is

$$_{tr}\vec{V}_p^{(2)} = \vec{\omega}_p^{(2)} \times \vec{B}_p \tag{3.22}$$

STEP 6: The relative velocity of the mean point B is

$$\vec{V}_p^{(2G)} = {_{tr}\vec{V}_p^{(2)}} - {_{tr}\vec{V}_p^{(G)}} \tag{3.23}$$

STEP 7: the projection of $\vec{V}_p^{(2G)}$ on the $\vec{e}_{G_{I_p}}$ is

$$V_{G_I}^{(2G)} = \vec{V}_p^{(2G)} \cdot \vec{e}_{G_{I_p}} \tag{3.24}$$

STEP 8: The projection of $\vec{V}_p^{(2G)}$ on the $\vec{e}_{G_{II_p}}$ is

$$V_{G_{II}}^{(2G)} = \vec{V}_p^{(2G)} \cdot \vec{e}_{G_{II_p}} \tag{3.25}$$

STEP 9: Using equation (A.33), we obtain

$$a_{13} = -\kappa_{G_I} V_{G_I}^{(2G)} - [\,\vec{\omega}_p^{(2G)} \vec{n}_p \vec{e}_{G_{I_p}}\,] \tag{3.26}$$

STEP 10: Using equation (A.35), we have

$$a_{23} = -\kappa_{G_{II}} V_{G_{II}}^{(2G)} - [\vec{\omega}_p^{(2G)} \vec{n}_p \vec{e}_{G_{II_p}}] \tag{3.27}$$

STEP 11: Using equation (A.36), we obtain

$$a_{33} = \kappa_{G_I}\left(V_{G_I}^{(2G)}\right)^2 + \kappa_{G_{II}}\left(V_{G_{II}}^{(2G)}\right)^2 - \left[\vec{n}_p \vec{\omega}_p^{(2G)} - \vec{V}_p^{(2G)}\right] - \vec{n}_p \cdot \left(\vec{\omega}_p^{(2)} \times {}_{tr}\vec{V}_p^{(G)} - \vec{\omega}_p^{(G)} \times {}_{tr}\vec{V}_p^{(2)}\right) \tag{3.28}$$

Note that $m'_{G2} = 0$

STEP 12: To determine the principal directions at point B on gear surface, we first use equation (A.40). Thus

$$\tan 2\sigma_{2G} = \frac{2a_{13}a_{23}}{a_{23}^2 - a_{13}^2 + (\kappa_{G_I} - \kappa_{G_{II}})a_{33}} \tag{3.29}$$

Rotating unit vector $\vec{e}_{G_I}$ about the unit normal vector $\vec{n}$ by $-\sigma_{2G}$, we may obtain unit vector $\vec{e}_{2_I}$. Rotating unit vector $\vec{e}_{2_I}$ about the unit normal vector $\vec{n}$ by $\pi/2$, we may have unit vector $\vec{e}_{2_{II}}$.

STEP 13: Using equations (A.41) and (A.42), we may determine the principal curvatures on the gear surfaces as follows:

$$\kappa_{2_I} - \kappa_{2_{II}} = \frac{a_{23}^2 - a_{13}^2 + (\kappa_{G_I} - \kappa_{G_{II}})a_{33}}{a_{33}\cos 2\sigma_{2G}} \tag{3.30}$$

$$\kappa_{2_I} + \kappa_{2_{II}} = (\kappa_{G_I} + \kappa_{G_{II}}) - \frac{a_{13}^2 + a_{23}^2}{a_{33}} \tag{3.31}$$

STEP 14: Eliminating $\kappa_{2_{II}}$ by considering the sum of equations (3.30) and (3.31) together and then dividing the sum by 2, we can determine κ_{2_I}. Eliminating κ_{2_I} by dividing the difference of equations (3.31) and (3.30) by 2, we can determine $\kappa_{2_{II}}$.

3.3 Local Synthesis

The determination of pinion machine-tool settings is based on the idea of local synthesis of gear tooth surfaces proposed by Litvin [2, 3, 4]. The goal of local synthesis for meshing of spiral bevel gears is to satisfy the following requirements:

1. The gear tooth surfaces must contact each other at the prescribed mean contact point B.

2. The contact ellipse for the gear tooth surface must have the desired dimensions at point B.

3. The tangent to the contact path must have the prescribed direction at point B.

4. The instant gear ratio $m_{21}(\phi_1)$ and its derivative $m'_{21}(\phi_1)$ must have the prescribed values at point B.

The local synthesis for the gear tooth surfaces connects the concept of meshing and the concept of bearing contact. It provides the optimal conditions of meshing for the gear tooth surfaces being in mesh at, and within the neighborhood of, the mean contact point B. The local synthesis needs the information on the characteristics of the tooth surfaces of the zero, first, and second orders.

Starting the local synthesis we already know the location of the mean contact point B on the gear surface, the unit normal to the gear tooth surface at point B, the principal curvatures and directions at point B on the gear tooth surface.

We will consider the local synthesis in a fixed coordinate system S_f. Figure 18 shows the relations among S_f, fixed coordinate systems which are attached to the frame of the gear generator, and fixed coordinate systems connected to the frame of the pinion generator. From Figure 18 we know that S_f and $S_{p(G)}$ coincide with each other. Therefore, the coordinates of the mean contact point B, the orientation of the surface unit normal, and the principal directions at the point B on the gear surface are known since they have been determined in the $S_{p(G)}$ system.

3.3.1 Preliminary Considerations

Spiral bevel gears transform rotation motion between intersecting shafts with an instantaneous point contact of surfaces. It corresponds to the second case discussed in Section A.2. Some elements of matrix $[A]$ shown in equation (A.30) are not related with the principal curvatures and directions of the pinion surface; therefore, they may be derived at the stage where the principal curvatures and directions are not known yet. We will consider that all derivations are performed in S_f coordinate system. Throughout the rest of the report, we will drop the subscript if it is considered in the S_f coordinate system.

The following representation is the result of direct observation of drawings of Figure 18.

$$\left[\omega^{(1)}\right] = \pm\omega^{(1)} \begin{bmatrix} \sin\mu_1 \\ 0 \\ \cos\mu_1 \end{bmatrix} \tag{3.32}$$

$$\left[\omega^{(2)}\right] = \pm m_{21}\omega^{(1)} \begin{bmatrix} -\sin\mu_2 \\ 0 \\ \cos\mu_2 \end{bmatrix} \tag{3.33}$$

Recall that the upper sign in the equations corresponds to a left-hand member. As far as a pair of spiral bevel gears is concerned, the hands of the spiral must be opposite; a left-hand gear (pinion) and a right-hand pinion (gear) constitute a pair. Therefore, if we take the upper sign in

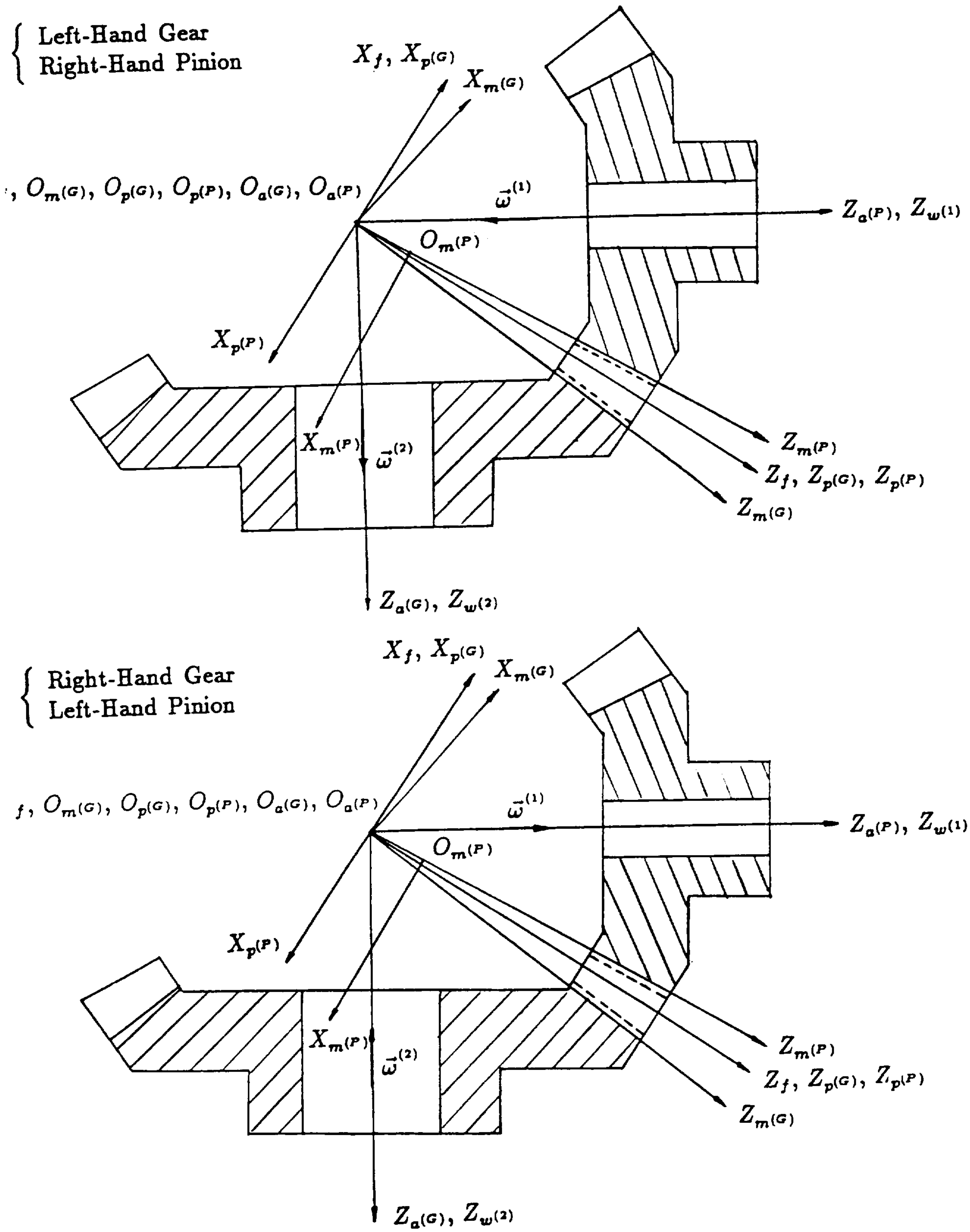


Figure 18: Coordinate systems for local synthesis.

equation (3.32), we must pick up the lower sign in equation (3.33). The relative angular velocity is

$$\vec{\omega}^{(12)} = \vec{\omega}^{(1)} - \vec{\omega}^{(2)} \tag{3.34}$$

The transfer velocity of the mean contact point B on surface Σ_1 is

$${}_{tr}\vec{V}^{(1)} = \vec{\omega}^{(1)} \times \vec{B} \tag{3.35}$$

The transfer velocity of the mean contact point B on surface Σ_2 is

$${}_{tr}\vec{V}^{(2)} = \vec{\omega}^{(2)} \times \vec{B} \tag{3.36}$$

The relative velocity of the mean point B is

$$\vec{V}^{(12)} = {}_{tr}\vec{V}^{(1)} - {}_{tr}\vec{V}^{(2)} \tag{3.37}$$

The projection of $\vec{V}^{(12)}$ on the vector $\vec{e}_{2_I}$ is

$$V_{2_I}^{(12)} = \vec{V}^{(12)} \cdot \vec{e}_{2_I} \tag{3.38}$$

The projection of $\vec{V}^{(12)}$ on the vector $\vec{e}_{2_{II}}$ is

$$V_{2_{II}}^{(12)} = \vec{V}^{(12)} \cdot \vec{e}_{2_{II}} \tag{3.39}$$

Let surfaces Σ_1 and $\Sigma_{\mathcal{F}}$, Σ_2 and $\Sigma_{\mathcal{Q}}$, be equivalent, respectively. Equation (A.33) yields

$$a_{31} = -\kappa_{2_I} V_{2_I}^{(12)} - [\vec{\omega}^{(12)} \vec{n} \vec{e}_{2_I}] \tag{3.40}$$

Using equation (A.35), we obtain

$$a_{32} = -\kappa_{2_{II}} V_{2_{II}}^{(12)} - [\vec{\omega}^{(12)} \vec{n} \vec{e}_{2_{II}}] \tag{3.41}$$

Equation (A.36) yields

$$\begin{aligned} a_{33} &= \kappa_{2_I} \left(V_{2_I}^{(12)}\right)^2 + \kappa_{2_{II}} \left(V_{2_{II}}^{(12)}\right)^2 - [\vec{n}\vec{\omega}^{(12)}\vec{V}^{(12)}] \\ &\quad - \vec{n} \cdot \left(\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(2)} - \vec{\omega}^{(2)} \times {}_{tr}\vec{V}^{(1)}\right) + \left(\omega^{(1)}\right)^2 m'_{21} \left(\vec{n} \times \vec{k}_2\right) \cdot \vec{B} \end{aligned} \tag{3.42}$$

where $\vec{k}_2$ is the unit vector along the axis of rotation of the gear. It is represented by (Figure 18)

$$[k_2] = \begin{bmatrix} -\sin\mu_2 \\ 0 \\ \cos\mu_2 \end{bmatrix} \tag{3.43}$$

In general, spiral bevel gears are designed and manufactured with non-conjugate tooth surfaces. Varying the machine-tool settings it is possible to obtain a lead function of transmission errors, a parabolic function with pinion lagging, or a parabolic function with gear lagging. Only a parabolic function with gear lagging is good for applications. Therefore, for the convex side of gear tooth m'_{21} we must provide a negative value, and for the concave side of gear tooth m'_{21} must be positive. The absolute value of m'_{21} controls the level of the transmission errors. We consider m'_{21} as an input.

On the gear surface a path of contact that appears almost straight and substantially vertical to the root may fully satisfy the operating requirements in many cases; however, it should not be assumed that this is true for all cases. Sometimes a different direction or shape may be preferable [11]. The tendency of the direction of the contact path may be determined by the relative velocity ${}_r\vec{V}^{(2)}$ at the mean contact point on the gear surface. Let ν_2 denote the angle between the unit vector $\vec{e}_{2_I}$ at the mean contact point on the gear surface and the direction of tangent at the same point to the path of contact. The relation between the principal directions and the direction of the contact path may be represented as follows (Figure 19):

$$\nu_2 = \Upsilon + \sigma_{2G} \tag{3.44}$$

The angle Υ is measured counterclockwise from the root to the tangent of the path. This angle is considered as an input.

3.3.2 Relations Between Directions of the Paths of the Mean Contact Point in its Motion over the Gear and Pinion Tooth Surfaces

Figure 20 shows the common tangent plane to the gear and pinion surfaces at the mean contact point B. The notations in Figure 20 are as follows:

$\vec{e}_{2_I}$ and $\vec{e}_{2_{II}}$:	unit vectors of the principal directions on the gear surface
$\vec{V}^{(12)}$:	sliding velocity at point B
${}_r\vec{V}^{(1)}$:	velocity vector of contact point B in its motion over the pinion surface
${}_r\vec{V}^{(2)}$:	velocity vector of contact point B in its motion over the gear surface

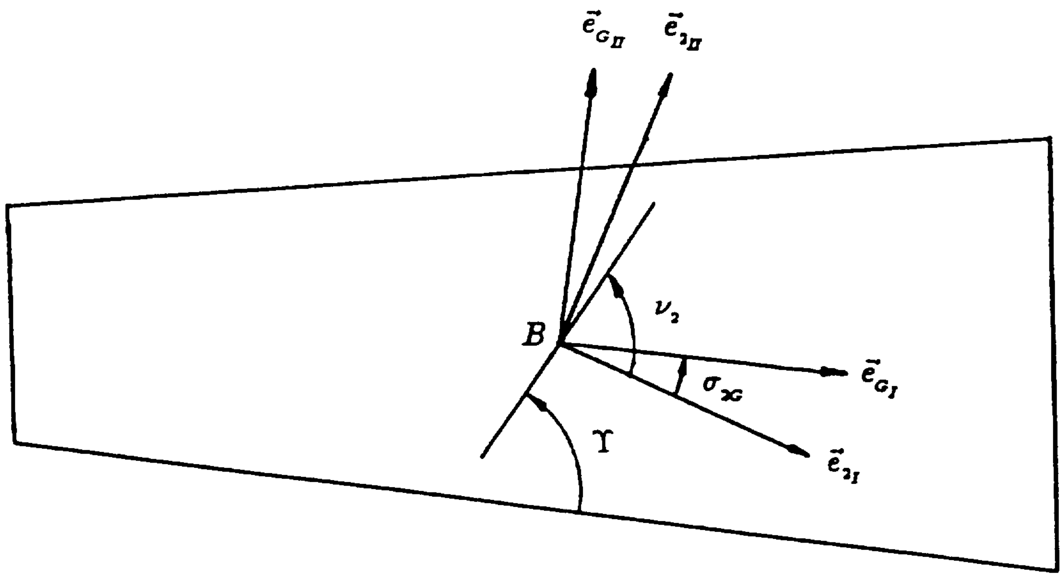


Figure 19: The direction of the contact path.

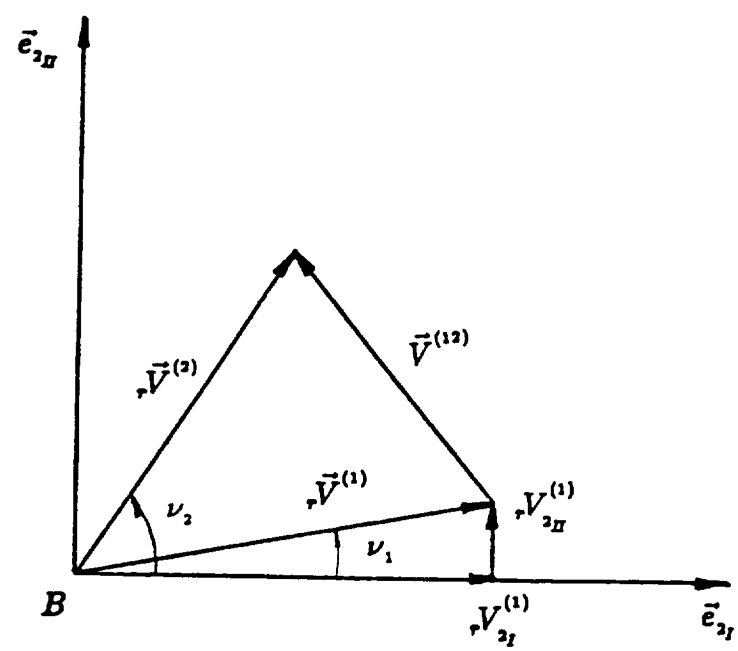


Figure 20: Common plane at the mean contact point.

${}_rV_{2_I}^{(1)}$ and ${}_rV_{2_{II}}^{(1)}$: the projections of vector ${}_r\vec{V}^{(1)}$ on vectors $\vec{e}_{2_I}$ and $\vec{e}_{2_{II}}$

ν_1 and ν_2: angles formed between vectors ${}_r\vec{V}^{(1)}$ and $\vec{e}_{2_I}$, ${}_r\vec{V}^{(2)}$ and $\vec{e}_{2_I}$, respectively

The relation between angles ν_1 and ν_2 depends on parameters in motion and the principal curvatures of the gear tooth surface. For derivations we will use the following equations:

$${}_r\vec{V}^{(2)} = {}_r\vec{V}^{(1)} + \vec{V}^{(12)} \tag{3.45}$$

that yields

$${}_rV_{2_I}^{(2)} = {}_rV_{2_I}^{(1)} + V_{2_I}^{(12)} \tag{3.46}$$

$${}_rV_{2_{II}}^{(2)} = {}_rV_{2_{II}}^{(1)} + V_{2_{II}}^{(12)} \tag{3.47}$$

From the geometric relations shown in Figure 20 we have

$${}_rV_{2_{II}}^{(2)} = {}_rV_{2_I}^{(2)} \tan\nu_2 \tag{3.48}$$

$${}_rV_{2_{II}}^{(1)} = {}_rV_{2_I}^{(1)} \tan\nu_1 \tag{3.49}$$

Substituting equation (3.48) and (3.49) into equation (3.47), and then substituting equation (3.46) into (3.47), we obtain an expression for ${}_rV_{2_I}^{(1)}$ in terms of $V_{2_I}^{(12)}$, $V_{2_{II}}^{(12)}$, ν_1, and ν_2 as follows

$${}_rV_{2_I}^{(1)} = \frac{V_{2_{II}}^{(12)} - V_{2_I}^{(12)}\tan\nu_2}{\tan\nu_2 - \tan\nu_1} \tag{3.50}$$

According to equation (A.29), ${}_{r}V_{2_I}^{(1)}$ and ${}_{r}V_{2_{II}}^{(1)}$ are related as follows:

$$a_{31}\,{}_{r}V_{2_I}^{(1)} + a_{32}\,{}_{r}V_{2_{II}}^{(1)} = a_{33} \tag{3.51}$$

Here surface Σ_2 is equivalent to surface $\Sigma_{\mathcal{Q}}$; surface Σ_1 to surface $\Sigma_{\mathcal{F}}$. Substituting equation (3.49) into equation (3.51), we have

$$(a_{31} + a_{32}\tan\nu_1)\,{}_{r}V_{2_I}^{(1)} = a_{33} \tag{3.52}$$

Finally, combining equations (3.50) and (3.52), we have the relation between angles ν_1 and ν_2

$$\tan\nu_1 = \frac{(a_{33} + a_{31}V_{2_I}^{(12)})\tan\nu_2 - a_{31}V_{2_{II}}^{(12)}}{a_{32}(V_{2_{II}}^{(12)} - V_{2_I}^{(12)}\tan\nu_2) + a_{33}} \tag{3.53}$$

3.3.3 Principal Curvatures and Directions of the Pinion Tooth Surface at the Mean Contact Point

The derivation of principal curvatures and directions of the pinion tooth surface at the mean contact point is based on the following procedure.

STEP 1: Representation of $\mathcal{A}$ and $\mathcal{B}$ in terms of coefficients a_{11}, a_{12}, and a_{22}

We recall that the lengths of semiaxes of the contact ellipse, a and b, are determined by parameters $\mathcal{A}$, $\mathcal{B}$, and ε(see Section A.4).

The sum of equations (A.31) and (A.34) yields

$$a_{11} + a_{22} = \kappa_{2\Sigma} - \kappa_{1\Sigma} \tag{3.54}$$

Substituting equation (A.31) by equation (A.34) we obtain

$$a_{11} - a_{22} = \kappa_{2\Delta} - \kappa_{1\Delta} \cos 2\sigma_{12} \tag{3.55}$$

We may represent parameter $\mathcal{A}$ in equation (A.54) in terms of a_{11}, a_{12}, and a_{22} as follows:

$$\mathcal{A} = -\frac{1}{4}\left[(a_{11} + a_{22}) + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}\right] \tag{3.56}$$

Also, the representation of parameter $\mathcal{B}$ in equation (A.55) in terms of a_{11}, a_{12}, and a_{22} gives

$$\mathcal{B} = -\frac{1}{4}\left[(a_{11} + a_{22}) - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}\right] \tag{3.57}$$

Furthermore, equations (3.56) and (3.57) yield

$$[(a_{11} + a_{22}) + 4\mathcal{A}]^2 = (a_{11} - a_{22})^2 + 4a_{12}^2 = [(a_{11} + a_{22}) + 4\mathcal{B}]^2 \tag{3.58}$$

Let $\mathcal{T}$ denote the smaller absolute value of $\mathcal{A}$ and $\mathcal{B}$. Therefore, equation (3.57) can be written as

$$[(a_{11} + a_{22}) + 4\mathcal{T}]^2 = (a_{11} - a_{22})^2 + 4a_{12}^2 \tag{3.59}$$

STEP 2: Representation of coefficients a_{11}, a_{12}, and a_{22} in terms of ${}_rV_{2_I}^{(1)}$ and ${}_rV_{2_{II}}^{(1)}$

Using the first two equations in (A.29) and equation (3.54), we may derive a system of three linear equations in unknowns a_{11}, a_{12}, and a_{22}

$$\left.\begin{array}{lllllll} {}_rV_{2_I}^{(1)} a_{11} & + & {}_rV_{2_{II}}^{(1)} a_{12} & & & = a_{13} \\ & & {}_rV_{2_I}^{(1)} a_{12} & + & {}_rV_{2_{II}}^{(1)} a_{22} & = a_{23} \\ a_{11} & + & & + & a_{22} & = \kappa_\Lambda \end{array}\right\} \tag{3.60}$$

where $\kappa_\Lambda = \kappa_{2\Sigma} - \kappa_{1\Sigma}$. Using *Cramer's rule* we may solve equations (3.60) as follows:

$$a_{11} = \frac{a_{13}\ {}_rV_{2_I}^{(1)} - a_{23}\ {}_rV_{2_{II}}^{(1)} + \kappa_\Lambda \left({}_rV_{2_{II}}^{(1)}\right)^2}{\left({}_rV_{2_I}^{(1)}\right)^2 + \left({}_rV_{2_{II}}^{(1)}\right)^2} \tag{3.61}$$

$$a_{12} = \frac{a_{13}\ {}_rV_{2_{II}}^{(1)} + a_{23}\ {}_rV_{2_I}^{(1)} - \kappa_\Lambda\ {}_rV_{2_I}^{(1)}\ {}_rV_{2_{II}}^{(1)}}{\left({}_rV_{2_I}^{(1)}\right)^2 + \left({}_rV_{2_{II}}^{(1)}\right)^2} \tag{3.62}$$

$$a_{22} = \frac{-a_{13}\ {}_rV_{2_I}^{(1)} + a_{23}\ {}_rV_{2_{II}}^{(1)} + \kappa_\Lambda \left({}_rV_{2_I}^{(1)}\right)^2}{\left({}_rV_{2_I}^{(1)}\right)^2 + \left({}_rV_{2_{II}}^{(1)}\right)^2} \tag{3.63}$$

The third equation in (A.29) is

$$a_{31}\ {}_rV_{2_I}^{(1)} + a_{32}\ {}_rV_{2_{II}}^{(1)} = a_{33} \tag{3.64}$$

Substituting equation (3.49) into equation (3.64), we have

$${}_rV_{2_I}^{(1)} = \frac{a_{33}}{a_{13} + a_{23} \tan\nu_1} \tag{3.65}$$

Plugging equations (3.49) and (3.65) into equations (3.61)–(3.63), we obtain the following results

$$a_{11} = d_1 \kappa_\Lambda + b_1 \tag{3.66}$$

$$a_{12} = d_2 \kappa_\Lambda + b_2 \tag{3.67}$$

$$a_{13} = d_3 \kappa_\Lambda + b_1 \tag{3.68}$$

where

$$d_1 = \frac{\tan^2 \nu_1}{1 + \tan^2 \nu_1} \tag{3.69}$$

$$d_2 = \frac{-\tan \nu_1}{1 + \tan^2 \nu_1} \tag{3.70}$$

$$d_3 = \frac{1}{1 + \tan^2 \nu_1} \tag{3.71}$$

$$b_1 = \frac{a_{13}^2 - a_{23}^2 \tan^2 \nu_1}{a_{33}(1 + \tan^2 \nu_1)} \tag{3.72}$$

$$b_2 = \frac{(a_{23} + a_{13} \tan \nu_1)(a_{13} + a_{23} \tan \nu_1)}{a_{33}(1 + \tan^2 \nu_1)} \tag{3.73}$$

STEP 3: Determination of κ_Λ

Equations (3.59) and (3.66)–(3.71) lead to

$$\kappa_\Lambda = -\frac{\left[4\mathcal{T}^2 - (b_1^2 + b_2^2)\right](1 + \tan^2\nu_1)}{2\mathcal{T}(1 + \tan^2\nu_1) + b_1(1 - \tan^2\nu_1) + 2b_2\tan\nu_1} \tag{3.74}$$

Since $\kappa_\Lambda = \kappa_{2\Sigma} - \kappa_{1\Sigma}$, equation (3.74) becomes

$$\kappa_{1\Sigma} = \kappa_{2\Sigma} + \frac{\left[4\mathcal{T}^2 - (b_1^2 + b_2^2)\right](1 + \tan^2\nu_1)}{2\mathcal{T}(1 + \tan^2\nu_1) + b_1(1 - \tan^2\nu_1) + 2b_2\tan\nu_1} \tag{3.75}$$

Note that

$$\mathcal{T} = \frac{\varepsilon}{t^2} \tag{3.76}$$

where t is the semimajor axis of the contact ellipse. This is an input datum. In general, it is about one sixth of the width of the gear tooth. Gleason Works suggests that the elastic approach ε is 0.00025 inches [11].

STEP 4: Determination of a_{11}, a_{12}, and a_{22}

Substituting equation (3.74) into equations (3.66)–(3.68), we obtain a_{11}, a_{12}, and a_{22}.

STEP 5: Determination of σ_{12}

Using equations (A.32) and (3.55), we obtain

$$\tan 2\sigma_{12} = \frac{2a_{12}}{\kappa_{2\Delta} - a_{11} + a_{22}} \tag{3.77}$$

It provides two solutions for σ_{12}, and we will choose the smaller value. Rotating unit vector $\vec{e}_{2_I}$ about the unit normal vector $\vec{n}$ by $-\sigma_{12}$, we may obtain unit vector $\vec{e}_{1_I}$. Rotating unit vector $\vec{e}_{1_I}$ about the unit normal vector $\vec{n}$ by $\pi/2$, we may obtain unit vector $\vec{e}_{2_{II}}$.

STEP 6: Determination of $\kappa_{1\Delta}$

Using equation (A.32), we obtain

$$\kappa_{1\Delta} = \frac{2a_{12}}{\sin 2\sigma_{12}} \tag{3.78}$$

STEP 7: Determination of κ_{1_I} and $\kappa_{1_{II}}$

The principal curvatures of the pinion surface at the mean contact point B are determined by

$$\kappa_{1_I} = \frac{\kappa_{1\Sigma} + \kappa_{1\Delta}}{2}, \qquad \kappa_{1_{II}} = \frac{\kappa_{1\Sigma} + \kappa_{1\Delta}}{2} \tag{3.79}$$

3.3.4 First Order Characteristics

Four surfaces, the gear head-cutter surface Σ_G, the gear surface Σ_2, the pinion head-cutter surface Σ_P, and the pinion surface Σ_1, are in tangency simultaneously at the mean contact point B. It implies that these four surfaces have a common normal at the mean contact point. We can use this information to determine pinion blade angle ψ_P and parameter τ_P.

The representation of the unit normal to the pinion head-cutter surface in the S_f coordinate system is

$$
\begin{aligned}
[n_f] &= [L_{fp^{(P)}}][L_{p^{(P)}m^{(P)}}][L_{m^{(P)}c^{(P)}}][n_{c^{(P)}}] \\
&= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\delta_1 & 0 & -\sin\delta_1 \\ 0 & 1 & 0 \\ \sin\delta_1 & 0 & \cos\delta_1 \end{bmatrix} \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_P & \pm\sin\phi_P \\ 0 & \mp\sin\phi_P & \cos\phi_P \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos q_P & \mp\sin q_P \\ 0 & \pm\sin q_P & \cos q_P \end{bmatrix} \begin{bmatrix} n_{c_x^{(P)}} \\ n_{c_y^{(P)}} \\ n_{c_z^{(P)}} \end{bmatrix}
\end{aligned}
\tag{3.80}
$$

Let us consider the straight-edged blade first. Equation (2.2) describes the unit normal in the S_c coordinate system. Before plugging equation (2.2) into equation (3.80), we must investigate the sense of equation (2.2). From Figure 18 we know we must choose the minus sign for the unit normal. Therefore, equations (2.2) and (3.80) yield (subscript 'f' is dropped)

$$
\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \cos\delta_1 \sin\psi_P - \sin\delta_1 \cos\psi_P \cos\tau_P \\ \cos\psi_P \sin\tau_P \\ -\sin\delta_1 \sin\psi_P - \cos\delta_1 \cos\psi_P \cos\tau_P \end{bmatrix} \tag{3.81}
$$

Multiplying n_x by $\cos\delta_1$, n_z by $-\sin\delta_1$, and then considering their sum, we obtain

$$
n_x \cos\delta_1 - n_z \sin\delta_1 = \sin\psi_P \tag{3.82}
$$

Obviously, the pinion blade angle is

$$
\psi_P = \begin{cases} \arcsin(n_x \cos\delta_1 - n_z \sin\delta_1) & \text{Gear Concave Side} \\ (\pi - \psi_P) & \text{Gear Convex Side} \end{cases} \tag{3.83}
$$

The x component in equation (3.81) may be rewritten as

$$\cos \tau_p = \frac{n_x - \cos \delta_1 \sin \psi_p}{- \sin \delta_1 \cos \psi_p} \tag{3.84}$$

The y component in equation (3.81) may be rewritten as

$$\sin \tau_p = \frac{n_y}{\cos \psi_p} \tag{3.85}$$

The parameter τ_p may be obtained by

$$\tau_p = 2 \arctan \frac{\sin \tau_p}{1 + \cos \tau_p} \tag{3.86}$$

Let us now consider the curve-edged blade. Equations (2.16) and (3.80) yield

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \cos \delta_1 \cos \lambda_p - \sin \delta_1 \sin \lambda_p \cos \tau_p \\ \sin \lambda_p \sin \tau_p \\ - \sin \delta_1 \cos \lambda_p - \cos \delta_1 \sin \lambda_p \cos \tau_p \end{bmatrix} \tag{3.87}$$

Multiplying n_x by $\cos \delta_1$, n_z by $- \sin \delta_1$, and then considering their sum, we obtain

$$\cos \lambda_p = n_x \cos \delta_1 - n_z \sin \delta_1 \tag{3.88}$$

The quadrant in which the parameter λ_p locates may be determined by the discussion stated in Section 2.2.

The blade angle is the angle formed by a line tangent to the blade surface at the mean contact point and a line perpendicular to the cutter head face. Thus we have

$$\psi_p = \begin{cases} 5/2\pi - \lambda_p & \text{pinion concave side, blade concave down;} \\ 3/2\pi - \lambda_p & \text{pinion concave side, blade concave up;} \\ 1/2\pi - \lambda_p & \text{pinion convex side, blade concave down;} \\ 3/2\pi - \lambda_p & \text{pinion convex side, blade concave up.} \end{cases}$$

Rewriting the x component in equation (3.87), we have

$$\cos \tau_p = \frac{n_x - \cos \delta_1 \cos \lambda_p}{- \sin \delta_1 \sin \lambda_p} \tag{3.89}$$

The y component in equation (3.87) may be rewritten as

$$\sin \tau_p = \frac{n_y}{\sin \lambda_p} \tag{3.90}$$

Substituting equations (3.89) and (3.90) into equations (3.86), we may obtain τ_p.

3.3.5 Principal Curvatures and Directions of the Pinion Cutter Surface at the Mean Contact Point

The first principal direction of the pinion cutter surface at the mean contact point may be represented in the S_p coordinate system as follows:

$$[e_{P_{I_f}}] = [L_{fp^{(P)}}][L_{p^{(P)}m^{(P)}}][L_{m^{(P)}c^{(P)}}][e_{P_{I_c}}] \tag{3.91}$$

Using equation (2.9) and (3.91), we may obtain the first principal direction for the straight-edged cutter. It is

$$\left[e_{P_{I_f}}\right] = \pm \begin{bmatrix} \sin\delta_1 \sin\tau_P \\ \cos\tau_P \\ \cos\delta_1 \sin\tau_P \end{bmatrix} \tag{3.92}$$

Using equations (2.17) and (3.91), we may obtain the first principal direction for the curve-edged cutter. The result is the same as for the straight-edged cutter, that is described in equation (3.92). In above equation, there are two senses. Only the direction which forms the smaller angle with the gear cutter first principal direction can be chosen. From the first order information we have already determined the parameter τ_P; therefore, the first principal direction of the pinion cutter is also determined. The unit vector of the second principal direction of the pinion cutter surface may be obtained by rotating the unit vector of the first principal direction of the pinion cutter surface, $\vec{e}_{P_I}$, about the common normal, $\vec{n}$, by an angle $\pi/2$.

We use the concept discussed in Section A.2 to derive the principal curvatures of the pinion cutter surface at the mean contact point. We recall that surfaces Σ_P and Σ_1 are in line contact in the process of generation. Hence, using equation (A.37), we obtain

$$a_{11}a_{22} - a_{12}^2 = 0 \tag{3.93}$$

Substituting equations(A.31), (A.32), and (A.34) into (3.93), we obtain the first principal curvature of the pinion cutter

$$\kappa_{P_I} = \frac{\kappa_{P_{II}}(\kappa_{1_I}\cos^2\sigma_{P1} + \kappa_{1_{II}}\sin^2\sigma_{P1}) - \kappa_{1_I}\kappa_{1_{II}}}{\kappa_{P_{II}} - \kappa_{1_I}\sin^2\sigma_{P1} - \kappa_{1_{II}}\cos^2\sigma_{P1}} \tag{3.94}$$

The second curvature of the pinion cutter is zero for a straight-edged cutter (see equation (2.12)) and $\mp 1/R$ for a curve-edged cutter (see equation (2.20)). Since the principal curvatures and directions of the pinion cutter surface at the mean contact point have been determined, some data relating to pinion machine-tool settings may be obtained without any difficulty.

Let us consider a straight-edged cutter first. Rewriting equation (2.10), we may obtain

$$u_P = \frac{1}{\kappa_{P_I}\tan\psi} \tag{3.95}$$

We choose only the positive sign in equation (2.10) since we have specified the direction of the unit normal $\vec{n}$. We may represent the mean contact point B in the $S_{m(P)}$ coordinate system as follows:

$$[B_{m(P)}] = \left[M_{m(P)p(P)}\right]\left[M_{p(P)f}\right][B_f] \tag{3.96}$$

where

$$\left[M_{m(P)p(P)}\right] = \begin{bmatrix} \cos\delta_1 & 0 & \sin\delta_1 & 0 \\ 0 & 1 & 0 & \mp E_m \\ -\sin\delta_1 & 0 & \cos\delta_1 & -L_m \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.97}$$

and

$$\left[M_{p(P)f}\right] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.98}$$

Considering only the x component of the above equation, we obtain

$$B_{m_x} = -B_{f_x} \cos\delta_1 + B_{f_z} \sin\delta_1 \tag{3.99}$$

Using equation (2.38), we obtain r_P as follows:

$$r_P = (B_{m_x} + u_P \cos\psi_P) \tan\psi_P \tag{3.100}$$

Let us now consider the curve-edged cutter. Using equation (2.18), we obtain

$$R_{c_z} = \pm \frac{\sin\lambda_P}{\kappa_{P_I}} - R\sin\lambda_P \tag{3.101}$$

The parameter R_{c_x} may be obtained by equations (2.40) and (3.99). That is

$$R_{c_x} = B_{m_x} - R\cos\lambda_P \tag{3.102}$$

The cutter tip radius may be represented by

$$r_P = R_{c_z} \pm \sqrt{|R^2 - R_{c_x}^2|} \tag{3.103}$$

3.4 Pinion Machine-Tool Settings

There are five machine-tool settings m_{P1}, E_m, L_m, s_P, and q_P to be determined. The key to the solution of this problem is the determination of the cutting ratio m_{P1}. Let us consider this problem first.

3.4.1 Determination of Pinion Cutting Ratio

We will use the relations between principal curvatures and directions for the pinion cutter surface and the pinion surface to derive the pinion cutting ratio m_{P1}. To apply the equations described in Section A.2, we consider that surfaces Σ_1 and $\Sigma_{\mathcal{F}}$ are equivalent, and that surfaces Σ_P and Σ_Q are equivalent. Also, the following data are considered as given: (1) the principal curvatures of the pinion surface at the mean contact point, κ_{1_I} and $\kappa_{1_{II}}$; (2) the principal directions of the pinion surface at the mean contact point, $\vec{e}_{1_I}$ and $\vec{e}_{1_{II}}$; (3) the coordinates of the mean contact point; (4) the unit normal at the mean contact point; (5) the coefficients a_{11}, a_{12}, and a_{22}.

The procedure to determine m_{P1} is as follows:

STEP 1: Representation of $\vec{\omega}^{(1P)}$

The angular velocity of the pinion is represented by

$$\vec{\omega}^{(1)} = \pm\omega^{(1)} \begin{bmatrix} \sin\mu_1 \\ 0 \\ \cos\mu_1 \end{bmatrix} \tag{3.104}$$

The angular velocity of the pinion cutter is represented by

$$\vec{\omega}^{(P)} = \pm m_{P1}\omega^{(1)} \begin{bmatrix} \cos\delta_1 \\ 0 \\ -\sin\delta_1 \end{bmatrix} \tag{3.105}$$

Therefore, we may obtain the relative angular velocity $\vec{\omega}^{(1P)}$ as follows:

$$\vec{\omega}^{(1P)} = \pm\omega^{(1)} \begin{bmatrix} \sin\mu_1 - m_{P1}\cos\delta_1 \\ 0 \\ \cos\mu_1 + m_{P1}\sin\delta_1 \end{bmatrix} \tag{3.106}$$

STEP 2: Representation of $[\,\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_I}]$

The scalar $[\,\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_I}]$ is represented by

$$\begin{aligned}
\left[\,\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_I}\right] &= \omega^{(1)} \begin{vmatrix} \pm(\sin\mu_1 - m_{P1}\cos\delta_1) & 0 & \pm(\cos\mu_1 + m_{P1}\sin\delta_1) \\ n_x & n_y & n_z \\ e_{P_{I_x}} & e_{P_{I_y}} & e_{P_{I_z}} \end{vmatrix} \\
&= \pm\Big\{\Big[(n_z e_{P_{I_y}} - n_y e_{P_{I_z}})\cos\delta_1 + (n_x e_{P_{I_y}} - n_y e_{P_{I_x}})\sin\delta_1\Big]\, m_{P1} \\
&\quad + \Big[(n_y e_{P_{I_z}} - n_z e_{P_{I_y}})\sin\mu_1 + (n_x e_{P_{I_y}} - n_y e_{P_{I_x}})\cos\mu_1\Big]\Big\}\,\omega^{(1)} \\
&= (c_{11} m_{P1} + c_{12})\,\omega^{(1)}
\end{aligned} \tag{3.107}$$

STEP 3: Representation of $[\,\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_{II}}]$

The scalar $[\,\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_{II}}]$ is represented by

$$\left[\vec{\omega}^{(1P)}\vec{n}\vec{e}_{P_{II}}\right] = \omega^{(1)}\begin{vmatrix} \pm(\sin\mu_1 - m_{P1}\cos\delta_1) & 0 & \pm(\cos\mu_1 + m_{P1}\sin\delta_1) \\ n_x & n_y & n_z \\ e_{P_{II_x}} & e_{P_{II_y}} & e_{P_{II_z}} \end{vmatrix}$$

$$= \pm\left[(n_y e_{P_{II_z}} - n_z e_{P_{II_y}})\sin\mu_1 + (n_x e_{P_{II_y}} - n_y e_{P_{II_x}})\cos\mu_1\right]\omega^{(1)}$$

$$= c_{22}\,\omega^{(1)} \tag{3.108}$$

STEP 4: Representation of $\vec{V}^{(1P)}$

The velocity ${}_{tr}\vec{V}^{(1)}$ may be obtained by

$$ {}_{tr}\vec{V}^{(1)} = \vec{\omega}^{(1)} \times \vec{B} = \pm\omega^{(1)}\begin{bmatrix} -B_y\cos\mu_1 \\ B_x\cos\mu_1 - B_z\sin\mu_1 \\ B_y\sin\mu_1 \end{bmatrix} \tag{3.109}$$

The velocity ${}_{tr}\vec{V}^{(P)}$ may be obtained by

$$ {}_{tr}\vec{V}^{(P)} = \vec{\omega}^{(P)} \times \vec{B} + \overline{O_f O_m} \times \vec{\omega}^{(P)} = \omega^{(1)} m_{P1}\begin{bmatrix} (E_m \pm B_y)\sin\delta_1 \\ \pm(L_m - B_x\sin\delta_1 - B_y\cos\delta_1) \\ (E_m \pm B_y)\cos\delta_1 \end{bmatrix} \tag{3.110}$$

The sliding velocity $\vec{V}^{(1P)}$ is described by

$$
\begin{aligned}
\vec{V}^{(1P)} &= {}_{tr}\vec{V}^{(1)} - {}_{tr}\vec{V}^{(P)} \\
&= \omega^{(1)} \begin{bmatrix} \mp B_y \cos\mu_1 - m_{P1}(E_m \pm B_y)\sin\delta_1 \\ \pm(B_x \cos\mu_1 - B_z \sin\mu_1) \mp m_{P1}[L_m - (B_x \sin\delta_1 + B_z \cos\delta_1)] \\ \pm B_y \sin\mu_1 - m_{P1}(E_m \pm B_y)\cos\delta_1 \end{bmatrix}
\end{aligned}
\tag{3.111}
$$

STEP 5: Representation of $V_{P_I}^{(1P)}$ and $V_{P_{II}}^{(1P)}$

Using equations (A.33) and (3.107), we have

$$
a_{13} = -\kappa_{P_I} V_{P_I}^{(1P)} - (c_{11} m_{P1} + c_{12})\,\omega^{(1)} \tag{3.112}
$$

Equations (A.35) and (3.108) yield

$$
a_{23} = -\kappa_{P_{II}} V_{P_{II}}^{(1P)} - c_{22}\,\omega^{(1)} \tag{3.113}
$$

From equation (A.37) we have

$$
a_{11} a_{23} - a_{12} a_{13} = 0 \tag{3.114}
$$

Using equations (3.112) - (3.114), we obtain

$$
a_{12} \kappa_{P_I} V_{P_I}^{(1P)} - a_{11} \kappa_{P_{II}} V_{P_{II}}^{(1P)} = [-a_{12} c_{11} m_{P1} + (a_{11} c_{22} - a_{12} c_{12})]\,\omega^{(1)} \tag{3.115}
$$

Moreover, we know that

$$\vec{V}^{(1P)} = V_{P_I}^{(1P)} \vec{e}_{P_I} + V_{P_{II}}^{(1P)} \vec{e}_{P_{II}} \tag{3.116}$$

Considering only the x component in equations (3.111) and (3.116), we obtain

$$V_{P_I}^{(1P)} e_{P_{I_x}} + V_{P_{II}}^{(1P)} e_{P_{II_x}} = [\mp B_y \cos\mu_1 - m_{P1}(E_m \pm B_y)\sin\delta_1]\,\omega^{(1)} \tag{3.117}$$

Considering only the z component in equations (3.111) and (3.116), we receive

$$V_{P_I}^{(1P)} e_{P_{I_z}} + V_{P_{II}}^{(1P)} e_{P_{II_z}} = [\pm B_y \sin\mu_1 - m_{P1}(E_m \pm B_y)\cos\delta_1]\,\omega^{(1)} \tag{3.118}$$

Multiplying equation (3.117) by $\cos\delta_1$ and equation (3.118) by $\sin\delta_1$, and adding the resulting equations, we obtain

$$V_{P_{II}}^{(1P)} = \mp \frac{B_y \cos\gamma_1}{e_{P_{II_x}} \cos\delta_1 - e_{P_{II_z}} \sin\delta_1}\,\omega^{(1)} = t_4 \omega^{(1)} \tag{3.119}$$

Substituting equation (3.119) into equation (3.117), we obtain

$$V_{P_I}^{(1P)} = \left(-\frac{c_{11}}{\kappa_{P_I}} m_{P1} + \frac{a_{11}\kappa_{P_{II}} t_4 + a_{11}c_{22} - a_{12}c_{12}}{a_{12}\kappa_{P_I}} \right) \omega^{(1)} = (t_1 m_{P1} + t_2)\,\omega^{(1)} \tag{3.120}$$

Step 6: Representation of $\vec{V}^{(1P)}$

The matrix form of equation (3.116) may be represented by

$$\vec{V}^{(1P)} = \begin{bmatrix} V_{P_I}^{(1P)} e_{P_{I_x}} + V_{P_{II}}^{(1P)} e_{P_{II_x}} \\ V_{P_I}^{(1P)} e_{P_{I_y}} + V_{P_{II}}^{(1P)} e_{P_{II_y}} \\ V_{P_I}^{(1P)} e_{P_{I_z}} + V_{P_{II}}^{(1P)} e_{P_{II_z}} \end{bmatrix} \tag{3.121}$$

Substituting equations (3.119) and (3.120) into equation (3.121), we have

$$\begin{aligned} \vec{V}^{(1P)} &= \omega^{(1)} \begin{bmatrix} (t_1 m_{P1} + t_2) e_{P_{I_x}} + t_4 e_{P_{II_x}} \\ (t_1 m_{P1} + t_2) e_{P_{I_y}} + t_4 e_{P_{II_y}} \\ (t_1 m_{P1} + t_2) e_{P_{I_z}} + t_4 e_{P_{II_z}} \end{bmatrix} \\ &= \omega^{(1)} \begin{bmatrix} u_{11} m_{P1} + u_{12} \\ u_{21} m_{P1} + u_{22} \\ u_{31} m_{P1} + u_{32} \end{bmatrix} \end{aligned} \tag{3.122}$$

STEP 7: Representation of $[\, \vec{n}\vec{\omega}^{(1P)} \vec{V}^{(1P)}]$

The scalar $[\, \vec{n}\vec{\omega}^{(1P)} \vec{V}^{(1P)}]$ may be represented by

$$\begin{aligned} \left[\vec{n}\vec{\omega}^{(1P)} \vec{V}^{(1P)}\right] &= \left[\omega^{(1)}\right]^2 \begin{vmatrix} n_x & n_y & n_z \\ \pm(\sin\mu_1 + m_{P1}\cos\delta_1) & 0 & \pm(\cos\mu_1 + m_{P1}\sin\delta_1) \\ u_{11} m_{P1} + u_{12} & u_{21} m_{P1} + u_{22} & u_{31} m_{P1} + u_{32} \end{vmatrix} \\ &= \left(v_1 m_{P1}^2 + v_2 m_{P1} + v_3\right) \left[\omega^{(1)}\right]^2 \end{aligned} \tag{3.123}$$

where

$$v_1 = \pm \left[(u_{11}\sin\delta_1 - u_{31}\cos\delta_1) n_y - (n_z\cos\delta_1 + n_x\sin\delta_1) u_{21}\right] \tag{3.124}$$

$$v_2 = \mp [(u_{21} \cos\mu_1 + u_{22} \sin\delta_1)n_x - (u_{21} \sin\mu_1 - u_{22} \cos\delta_1)n_z - (u_{11} \cos\mu_1 + u_{12} \sin\delta_1 + u_{32} \cos\delta_1 - u_{31} \sin\mu_1)n_y] \tag{3.125}$$

$$v_3 = \mp [(u_{22} n_x \cos\mu_1 - (u_{12} \cos\mu_1 - u_{32} \sin\mu_1)n_y - u_{22} n_z \sin\mu_1] \tag{3.126}$$

STEP 8: Representation of $\vec{n} \cdot (\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)} - \vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)})$

The velocity ${}_{tr}\vec{V}^{(P)}$ may be described by

$${}_{tr}\vec{V}^{(P)} = {}_{tr}\vec{V}^{(1)} - \vec{V}^{(1P)} \tag{3.127}$$

Substituting equations (3.109) and (3.122) into equation (3.127), we have

$${}_{tr}\vec{V}^{(P)} = \omega^{(1)} \begin{bmatrix} -u_{11} m_{P1} \mp B_y \cos\mu_1 - u_{12} \\ -u_{21} m_{P1} \mp B_z \sin\mu_1 \pm B_x \cos\mu_1 - u_{22} \\ -u_{31} m_{P1} \pm B_y \sin\mu_1 - u_{32} \end{bmatrix} \tag{3.128}$$

Vector $(\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)})$ is represented by

$$\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)} = \left[\omega^{(1)}\right]^2 \begin{bmatrix} \pm[u_{21} m_{P1} - (B_z \sin\mu_1 - B_x \cos\mu_1 \pm u_{22})] \cos\mu_1 \\ (\mp u_{11} \cos\mu_1 \pm u_{31} \sin\mu_1) m_{P1} - B_y \mp u_{12} \cos\mu_1 \pm u_{32} \sin\mu_1 \\ \mp[u_{21} m_{P1} - (B_z \sin\mu_1 - B_x \cos\mu_1 \pm u_{22})] \sin\mu_1 \end{bmatrix} \tag{3.129}$$

Vector $(\vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)})$ is represented by

$$\vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)} = \left[\omega^{(1)}\right]^2 \begin{bmatrix} -(B_z \sin\mu_1 - B_x \cos\mu_1) m_{P1} \sin\delta_1 \\ -B_y m_{P1} \sin\gamma_1 \\ -(B_z \sin\mu_1 - B_x \cos\mu_1) m_{P1} \cos\delta_1 \end{bmatrix} \tag{3.130}$$

Subtracting equation (3.130) from equation (3.129), we obtain

$$\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)} - \vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)} = \left[\omega^{(1)}\right]^2 \begin{bmatrix} h_{11} m_{P1} + h_{12} \\ h_{21} m_{P1} + h_{22} \\ h_{31} m_{P1} + h_{32} \end{bmatrix} \tag{3.131}$$

where

$$h_{11} = \pm u_{21} \cos\mu_1 - (B_x \cos\mu_1 - B_z \sin\mu_1) \sin\delta_1 \tag{3.132}$$

$$h_{12} = (B_z \sin\mu_1 - B_x \cos\mu_1 \pm u_{22}) \cos\mu_1 \tag{3.133}$$

$$h_{21} = B_y \sin\gamma_1 \mp u_{11} \cos\mu_1 \pm u_{31} \sin\mu_1 \tag{3.134}$$

$$h_{22} = -(B_y \pm u_{12} \cos\mu_1 \mp u_{32} \sin\mu_1) \tag{3.135}$$

$$h_{31} = \mp u_{21} \sin\mu_1 - (B_x \cos\mu_1 - B_z \sin\mu_1) \cos\delta_1 \tag{3.136}$$

$$h_{32} = -(B_z \sin\mu_1 - B_x \cos\mu_1 \pm u_{22}) \sin\mu_1 \tag{3.137}$$

Therefore, we may obtain $\vec{n} \cdot (\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)} - \vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)})$ as follows:

$$\vec{n} \cdot \left(\vec{\omega}^{(1)} \times {}_{tr}\vec{V}^{(P)} - \vec{\omega}^{(P)} \times {}_{tr}\vec{V}^{(1)}\right) = (f_1 m_{P1} + f_2) \left[\omega^{(1)}\right]^2 \tag{3.138}$$

where

$$f_1 = n_x h_{11} + n_y h_{21} + n_z h_{31} \tag{3.139}$$

$$f_2 = n_x h_{12} + n_y h_{22} + n_z h_{32} \tag{3.140}$$

STEP 9: Representation of m_{P1}

Using equations (A.33), (3.107), and (3.120), the equation for a_{13} may be represented by

$$a_{13} = -\left(\kappa_{P_I} t_2 + c_{12}\right) \omega^{(1)} \tag{3.141}$$

Using equations (A.35), (3.108), and (3.119), a_{23} may be described by

$$a_{23} = -\left(\kappa_{P_{II}} t_4 + c_{22}\right) \omega^{(1)} \tag{3.142}$$

Using equations (A.36), (3.119), (3.120), (3.123), and (3.138), the expression for a_{33} may be represented by

$$a_{33} = \left[(2\kappa_{P_I} t_1 t_2 - v_2 - f_1) m_{P1} + (\kappa_{P_I} t_2^2 + \kappa_{P_{II}} t_4^2 - v_3 - f_2)\right] \left[\omega^{(1)}\right]^2 \tag{3.143}$$

From equation (A.39) we know that

$$a_{12} a_{33} - a_{13} a_{23} = 0 \tag{3.144}$$

Equations (3.141)-(3.144) yield

$$m_{P1} = -\frac{a_{12}(\kappa_{P_I} t_2^2 + \kappa_{P_{II}} t_4^2 - v_3 - f_2) - (\kappa_{P_I} t_2 + c_{12})(\kappa_{P_{II}} t_4 + c_{22})}{a_{12}(2\kappa_{P_I} t_1 t_2 - v_2 - f_1)} \tag{3.145}$$

3.4.2 Determination of parameters E_m and L_m

Parameters E_m and L_m of the pinion machine-tool settings have been shown in Figures 12 and 13. Since the pinion cutting ratio m_{P1} has been determined, it is very easy to find these two parameters. We may determine vector $\vec{V}^{(1P)}$ from equation (3.122). Applying equation (3.111), then, we obtain

$$E_m = \frac{\mp B_y \cos \mu_1 - V_x^{(1P)}}{m_{P1} \sin \delta_1} \mp B_y \tag{3.146}$$

$$L_m = \frac{B_y \cos \mu_1 - B_z \sin \mu_1 \mp V_y^{(1P)}}{m_{P1}} + B_x \sin \delta_1 + B_z \cos \delta_1 \tag{3.147}$$

3.4.3 Determination of Pinion Radial Setting and Cradle Angle

The determination of the pinion radial setting and the cradle angle is based on the consideration that the position vectors of the pinion tooth surface and head-cutter surface must coincide at the mean contact point. Equation (3.96) describes the mean contact point B in the $S_{m^{(P)}}$ coordinate system. Considering the y and z components in equation (3.96), we obtain

$$B_{m_y^{(P)}} = -B_{f_y} \mp E_m \tag{3.148}$$

$$B_{m_z^{(P)}} = B_{f_x} \sin\delta_1 + B_{f_z} \cos\delta_1 - L_m \tag{3.149}$$

For a straight-edged cutter, by using equations (2.38), (3.148), and (3.149), we have

$$s_P \sin q_P = \pm B_{f_y} + E_m \pm u_P \sin\psi_P \sin\tau_P \tag{3.150}$$

$$s_P \cos q_P = B_{f_x} \sin\delta_1 + B_{f_z} \cos\delta_1 - L_m - u_P \sin\psi_P \cos\tau_P \tag{3.151}$$

For a curved-edged cutter, by using equations (2.40), (3.148), and (3.149), we have

$$s_P \sin q_P = \pm B_{f_y} + E_m \pm \frac{\cos\lambda_P \sin\tau_P}{\kappa_{P_I}} \tag{3.152}$$

$$s_P \cos q_P = B_{f_x} \sin\delta_1 + B_{f_z} \cos\delta_1 - L_m \pm \frac{\cos\lambda_P \cos\tau_P}{\kappa_{P_I}} \tag{3.153}$$

Using $\sin^2 q_P + \cos^2 q_P = 1$, we eliminate q_P and solve for pinion radius s_P. Eliminating s_P, we may determine the pinion cradle angle q_P.

CHAPTER 4

CONCLUSION

As it was mentioned in Chapter 1, the reduction of transmission errors of spiral bevel gears is a difficult problem. Although it is possible to generate conjugate spiral bevel gears, with zero transmission errors, we have to take into account that the gear are very sensitive to misalignment. Using the TCA programs we have found that even a small misalignment of gears results in discontinuity of functions of transmission errors that is accompanied with the jump of the function at the transfer points. Thus the idea of gears with non-zero transmission has to be complemented with the modification of the process for their generation that allows to reduce the sensitivity of gears to their misalignment.

From the result of computation by TCA programs we know that gear misalignment causes a linear or almost linear function of transmission errors. Litvin has discovered that a sum of a parabolic function and a linear function represents again a parabolic function that is just translated with respect to the initial parabolic function. Then, if a parabolic function is predesigned, it becomes possible to keep the same level of transmission errors for aligned as well as misaligned gears.

Gear misalignment is also accompanied with the shift of the bearing contact to the edge of gear tooth surface. To keep the shift of bearing contact in reasonable limits, it is necessary to limit the tolerances for gear misalignment and the respective value of predesigned parabolic function.

In Chapter 2 the basic concept and methods of Gleason systems have been presented. Equations that describe the surface of the head cutter, which is either a cone surface or a surface of revolution, have been derived. These equations covers the determination of position vectors, surface unit normal vectors, principal curvatures, and principal directions.

Mathematical models for geometry of spiral bevel gears have been also proposed in Chapter 2. The gear surface is represented as an envelope of the family of the tool surfaces. The tool surface and being generated gear surface are considered conjugate ones. Based on the geometric properties of conjugate surfaces, the equation of meshing has been derived.

The determination of pinion machine-tool settings is based on the method of local synthesis. The first derivative of gear ratio, the tangent to the contact path, and the dimensions of the contact ellipse of the gear surface at the mean contact point are considered as input to local synthesis. Thus the level of transmission errors and the bearing contact are under control. It provides the optimal conditions of meshing for the gear surfaces being in meshing at, and within the neighborhood of, the mean contact point.

Equations that determine the principal curvatures and directions at the mean contact point on the pinion surface have been derived. They are functions of the principal curvatures and directions at the mean contact point on the gear surface and the input of local synthesis. Based on the information on the characteristics of the pinion surface of the zero (position), first (normal), and second (principal curvatures and directions) orders, equations that determine the pinion basic machine-tool settings have been derived.

In Appendix A the basic concept and methods of theory of gearing that have been used in this work have been presented. Numerical examples are given in Appendix B. These examples include determination of machine-tool settings and results of computation by TCA programs. Computer programs have been developed, that include machine-tool settings and TCA. They are listed in Appendix C. The computer programs cover determination of machine-tool settings for straight-lined as well as curved blades. The developed TCA programs allow to simulate the meshing of aligned and misaligned gears.

APPENDIX A

GEOMETRY AND KINEMATICS OF GEARS IN THREE DIMENSIONS

A.1 Concept of Surfaces[1]

Most of the ideas underlying gear theory are based on strict definitions proposed in the field of differential geometry. In what follows we introduce the concept that is applied in this report.

All in all we require that our functions can be differentiated at least once and usually more times. Accordingly we say a function F belongs to class C^n on an interval $\mathcal{I}$ if the nth order derivative of F exists and is continuous on $\mathcal{I}$. In addition, we denote the class of continuous functions by C^0.

A parametric representation of a surface Σ is a continuous mapping of an open rectangle $\Re$, given in the plane P of the parameters (u, v), onto a three-dimensional space E^3 such that

$$\vec{B}(u,v) \in C^0, \qquad (u,v) \in \Re \tag{A.1}$$

where $\vec{B}$ is the position vector which determines the point surface (Figure 21). The vector function $\vec{B}(u,v)$ may be represented by

$$\vec{B}(u,v) = B_x(u,v)\,\vec{\imath} + B_y(u,v)\,\vec{\jmath} + B_z(u,v)\,\vec{k} \tag{A.2}$$

[1] Adopted from the manuscript of the book "Theory of Gearing" by Litvin, in press by NASA.

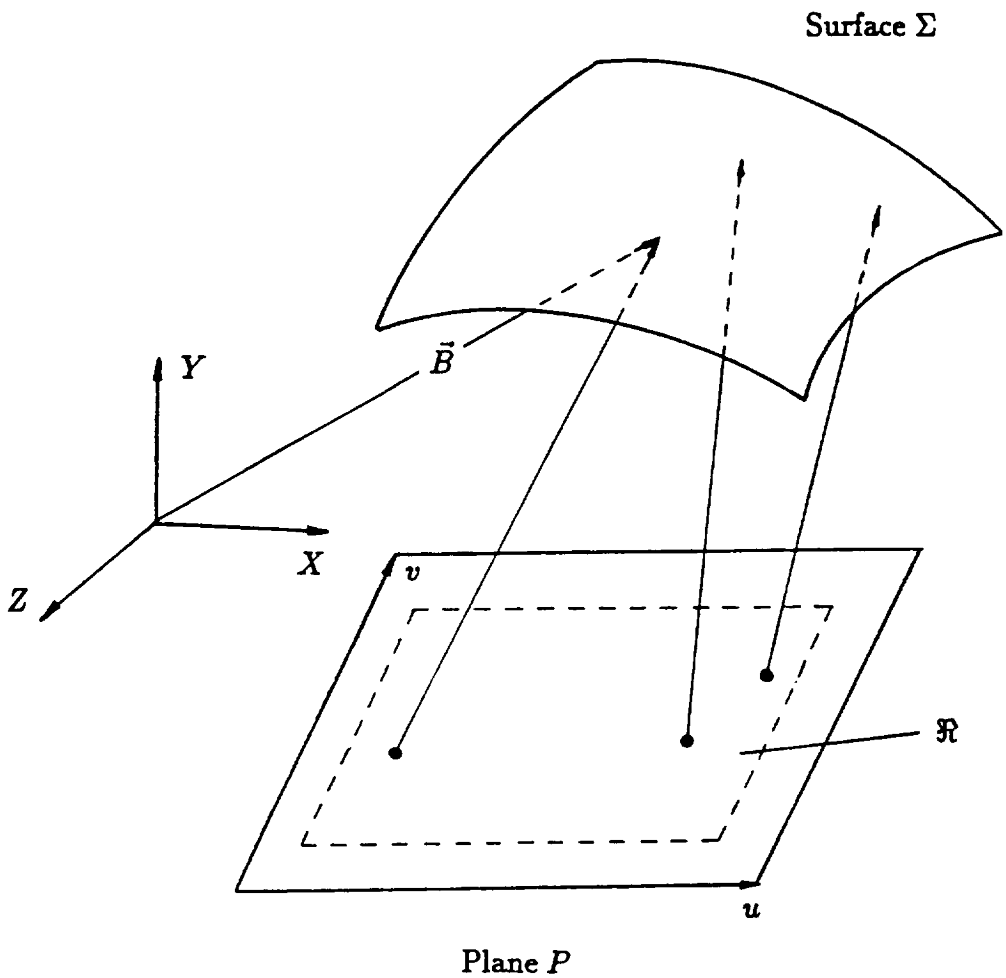


Figure 21: A parametric representation of a surface.

where $\vec{\imath}$, $\vec{\jmath}$, and $\vec{k}$ are unit vectors of the coordinate axes.

We call a surface point $\vec{B}(u,v)$ a regular point if at this point

$$\vec{B}_u \times \vec{B}_v \neq 0 \tag{A.3}$$

where

$$\vec{B}_u = \frac{\partial \vec{B}}{\partial u}, \qquad \vec{B}_v = \frac{\partial \vec{B}}{\partial v}$$

A surface is called a regular one if each point on it is a regular point.

A regular surface has the following properties:

- It is at least class of C^1.
- There is a one-to-one correspondence between the points of plane P (of the parameters (u,v)) and the three-dimensional space E^3.
- A regular surface has a tangent plane at all its points.

The normal vector $\vec{N}$ to the surface at a point B is

$$\vec{N} = \vec{B}_u \times \vec{B}_v \tag{A.4}$$

and its unit normal is represented by

$$\vec{n} = \frac{\vec{N}}{|\vec{N}|} = \frac{\vec{B}_u \times \vec{B}_v}{\left|\vec{B}_u \times \vec{B}_v\right|} \tag{A.5}$$

The direction of the surface normal $\vec{N}$ and unit normal $\vec{n}$, with respect to the surface, depends on the order of the factors of the cross product (equation (A.4)). By changing the order of the factors, we may change the direction of the normal to the opposite direction.

A surface is uniquely determined by certain local invariant quantities called the first and second fundamental forms. The first fundamental form of a surface is defined by

$$\begin{aligned} I \;=\; d\vec{B}\cdot d\vec{B} \;&=\; (\vec{B}_u\,du + \vec{B}_v\,dv)\cdot(\vec{B}_u\,du + \vec{B}_v\,dv) \\ &=\; (\vec{B}_u\cdot\vec{B}_u)du^2 + 2(\vec{B}_u\cdot\vec{B}_v)du^2\,dv^2 + (\vec{B}_v\cdot\vec{B}_v)dv^2 \qquad \text{(A.6)} \\ &=\; E\,du^2 + 2F\,du\,dv + G\,dv^2 \end{aligned}$$

where we set

$$E = \vec{B}_u\cdot\vec{B}_u, \qquad F = \vec{B}_u\cdot\vec{B}_v, \qquad G = \vec{B}_v\cdot\vec{B}_v$$

The second fundamental form is

$$\begin{aligned} II \;=\; -\,d\vec{B}\cdot d\vec{n} \;&=\; -(\vec{B}_u\,du + \vec{B}_v\,dv)\cdot(\vec{n}_u\,du + \vec{n}_v\,dv) \\ &=\; -(\vec{B}_u\cdot\vec{n}_u)du^2 - (\vec{B}_u\cdot\vec{n}_v + \vec{B}_v\cdot\vec{n}_u)du\,dv - (\vec{B}_v\cdot\vec{n}_v)dv^2 \qquad \text{(A.7)} \\ &=\; L\,du^2 + 2M\,du\,dv + N\,dv^2 \end{aligned}$$

where we have

$$L = -\vec{B}_u\cdot\vec{n}_u, \qquad M = -\frac{1}{2}(\vec{B}_u\cdot\vec{n}_v + \vec{B}_v\cdot\vec{n}_u), \qquad N = -\vec{B}_v\cdot\vec{n}_v$$

The second fundamental form exists only if the surface is at least class C^2. In this report we will consider all the gear tooth surfaces as regular surfaces with class at least C^2.

c - 2

On a given surface various curves pass through a common point B and have the same unit tangent vector $\vec{\tau}$ at B (Figure 22). One of these curves (designated by L_0) is located on the plane P, which is drawn through the unit tangent vector $\vec{\tau}$ and the surface unit normal $\vec{n}$. The curvature of curve L_0 is called normal curvature. Since the unit tangent vector $\vec{\tau}$ of the surface may have different directions on the surface, for each direction there is a normal curvature. The normal curvature is a function of the first and second fundamental forms:

$$\kappa_n = \frac{II}{I} = \frac{L\,du^2 + 2M\,du\,dv + N\,dv^2}{E\,du^2 + 2F\,du\,dv + G\,dv^2} \tag{A.8}$$

The extreme value of the normal curvature taken at a certain point of the surface are called the principal curvatures. The directions of the normal sections of the surface with the extreme normal curvatures are called the principal directions. Equation (A.8) yields

$$\mathcal{F} = \kappa_n(E\,du^2 + 2F\,du\,dv + G\,dv^2) - (L\,du^2 + 2M\,du\,dv + N\,dv^2) = 0 \tag{A.9}$$

For a given point on the surface, E, F, G, L, M, and N are constant. The normal curvature κ_n is a function of the ratio du and dv. Therefore, equation (A.9) is an identity of du and dv. From calculus, the partial derivative

$$\frac{\partial \mathcal{F}}{\partial\, du} = 0 \tag{A.10}$$

Substituting equation (A.9) into equation (A.10), it yields

$$\kappa_n(E\,du + F\,dv) - (L\,du + M\,dv) + \frac{\partial \kappa_n}{\partial\, du}(E\,du^2 + 2F\,du\,dv + G\,dv^2) = 0 \tag{A.11}$$

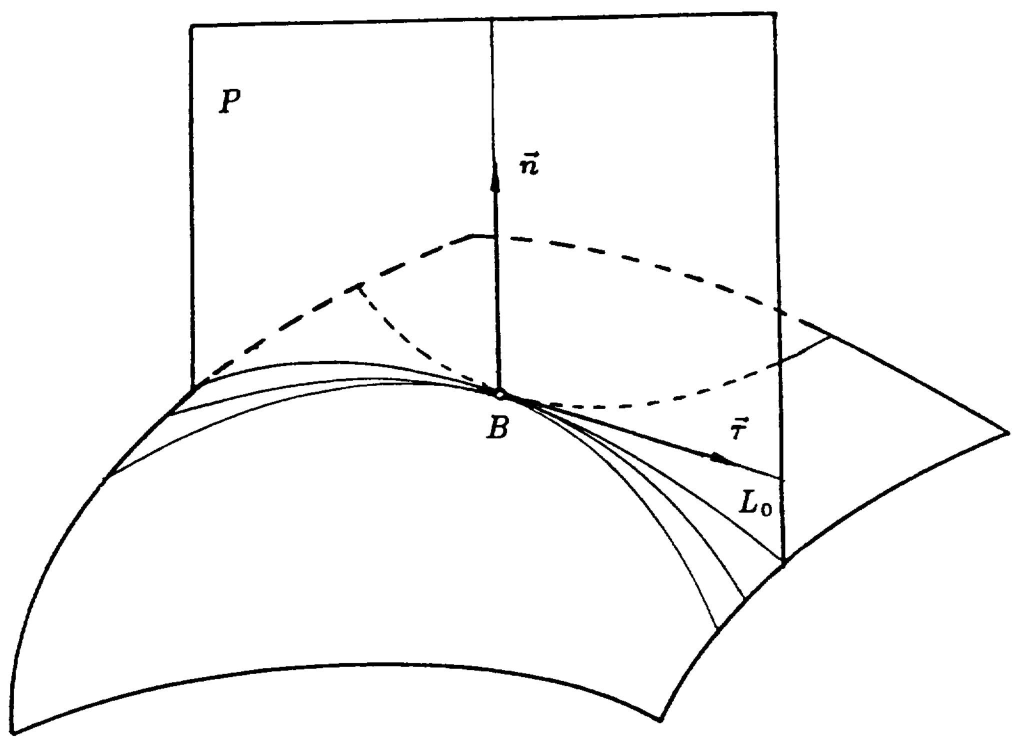


Figure 22: The normal curvature.

Also, the partial derivative

$$\frac{\partial \mathcal{F}}{\partial \, dv} = 0 \tag{A.12}$$

Substituting equation (A.9) into equation (A.12), it yields

$$\kappa_n(F\,du + G\,dv) - (M\,du + N\,dv) + \frac{\partial \kappa_n}{\partial \, dv} E\,du^2 + 2F\,du\,dv + G\,dv^2 = 0 \tag{A.13}$$

Recall that the principal curvatures are the extreme values of the normal curvature κ_n. Thus $\partial\kappa_n/\partial\,du = 0$ and $\partial\kappa_n/\partial\,dv = 0$ if κ_n is the principal curvature. Equations (A.11) and (A.13) yield

$$(\kappa_n E - L)\,du + (\kappa_n F - M)\,dv = 0 \tag{A.14}$$

and

$$(\kappa_n F - M)\,du + (\kappa_n G - N)\,dv = 0, \tag{A.15}$$

respectively. Solving the homogeneous system of equation (A.14) and (A.15) by eliminating du and dv, we obtain

$$(EG - F^2)\kappa_n^2 - (EN - 2FM + GL)\kappa_n + (LN - M^2) = 0 \tag{A.16}$$

The discriminant of equation (A.16) is

$$\begin{aligned}\triangle &= (EN - 2FM + GL)^2 - 4(EG - F^2)(LN - M^2) \\ &= \left[(EN - GL) - \tfrac{2F}{E}(EM - FL)\right]^2 + \frac{4(EG - F^2)}{E^2}(EM - FL)^2\end{aligned} \tag{A.17}$$

Equation (A.17) shows that the discriminant is greater than or equal to zero. Thus the equation has either two distinct real roots—the principal curvatures at a nonumbilical point, or a single real root with multiplicity two—the curvature at an umbilical point. The discriminant is equal to zero if and only if

$$EN - GL = EM - FL = 0 \tag{A.18}$$

Since $E \neq 0$ and $G \neq 0$, equation (A.18) can be shown to be identically equal to

$$\frac{L}{E} = \frac{M}{F} = \frac{N}{G} = \aleph \tag{A.19}$$

Considering equations (A.8) and (A.19), we obtain

$$\kappa_n = \aleph \tag{A.20}$$

This means that the principal curvature is the same as the normal curvature at any direction. Thus each direction may be considered as a principal direction. Any point which is on a plane or at which a surface turns into a plane[2] is an umbilical point. Any point on a spherical surface[3] is also an umbilical point.

Two distinct principal curvatures can always be obtained at a nonumbilical point. These two curvatures correspond to two distinct principal directions. By canceling κ_n from equations (A.14)

[2]The normal curvature on each direction is zero.
[3]The normal curvature on each direction is the inverse of the radius.

and (A.15), we have the following equation for principal directions

$$(EM - FL)\,du^2 - (GL - EN)\,du\,dv + (FN - GM)\,dv^2 = 0 \tag{A.21}$$

The discriminant of the above equation is identical to equation (A.17). At a nonumbilical point equation (A.21) can be represented as a product of two co-factors $(A_i\,du + B_i\,dv)(i = 1, 2)$ since the discriminant is larger than zero. This means that it represents two perpendicular directions. Thus we may conclude that at a nonumbilical point there exist two distinct principal curvatures in two perpendicular directions.

After representing of E, F, G, L, M, and N in the form of $\vec{B}_u$, $\vec{B}_v$, $\vec{n}_u$, and $\vec{n}_v$ by equations (A.6) and (A.7), equations (A.14) and (A.15) will yield

$$\vec{B}_u \cdot (\kappa_n\,d\vec{B} + d\vec{n}) = 0 \tag{A.22}$$

$$\vec{B}_v \cdot (\kappa_n\,d\vec{B} + d\vec{n}) = 0 \tag{A.23}$$

Obviously,

$$\vec{n} \cdot (\kappa_n\,d\vec{B} + d\vec{n}) = 0 \tag{A.24}$$

Therefore, $\kappa_n\,d\vec{B} + d\vec{n}$ is a zero vector since it is orthogonal to $\vec{B}_u$, $\vec{B}_v$, and $\vec{n}$. In short, we have

$$d\vec{n} = -\kappa_n\,d\vec{B} \tag{A.25}$$

The above equation, which completely characterizes the principal curvatures and directions, is called *Rodrigues' formula*. This formula simplifies the calculations to obtain principal curvatures and principal directions. The matrix form of *Rodrigues' formula* is

$$\begin{bmatrix} \dfrac{\partial n_x}{\partial u}\, du + \dfrac{\partial n_x}{\partial v}\, dv \\[2ex] \dfrac{\partial n_y}{\partial u}\, du + \dfrac{\partial n_y}{\partial v}\, dv \\[2ex] \dfrac{\partial n_z}{\partial u}\, du + \dfrac{\partial n_z}{\partial v}\, dv \end{bmatrix} = -\kappa_{I,II} \begin{bmatrix} \dfrac{\partial B_x}{\partial u}\, du + \dfrac{\partial B_x}{\partial v}\, dv \\[2ex] \dfrac{\partial B_y}{\partial u}\, du + \dfrac{\partial B_y}{\partial v}\, dv \\[2ex] \dfrac{\partial B_z}{\partial u}\, du + \dfrac{\partial B_z}{\partial v}\, dv \end{bmatrix} \tag{A.26}$$

Matrix equation (A.26) yields three scalar equations in three unknowns, the ratio du/dv, and the principal curvatures κ_I and κ_{II}. Using any two of the scalar equations we may develop a quadratic equation (provided $dv \neq 0$)

$$A_2 \left(\frac{du}{dv}\right)^2 + A_1 \frac{du}{dv} + A_0 = 0 \tag{A.27}$$

The two roots of this equation correspond to two principal directions on the surface. By putting both roots into the third scalar equation, we may determine the principal curvatures κ_I and κ_{II}.

It is possible to have either positive or negative principal curvatures. The sense of the principal curvature depends on the location of the center of curvature on the normal. The principal curvature is positive if the center of curvature is located on the positive normal.

The normal curvature on each direction may be expressed in terms of principal curvatures. This is so called *Euler's Theorem*. That states

$$\kappa_n = \kappa_I \cos^2 \varpi + \kappa_{II} \sin^2 \varpi \tag{A.28}$$

where ϖ is the angle formed by the tangent to the normal curvature and principal direction with curvature κ_I.

A.2 Relations Between Principal Curvatures and Directions for Mating Surfaces

Consider two gear surfaces $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{Q}}$ which are in meshing. Moreover, we have the following assumptions:

1. The rotation angles, $\phi_{\mathcal{F}}$ and $\phi_{\mathcal{Q}}$, of both gears are given;

2. The function $\phi_{\mathcal{Q}}(\phi_{\mathcal{F}})$ has continuous derivatives of second order;

3. The angular velocity $\omega^{(\mathcal{F})}$ of gear $\mathcal{F}$ is constant.

Then relations between principal curvatures and directions of these mating surfaces may be determined. Such relations were first proposed by Litvin [12] and then extended for the case $m'_{\mathcal{F}\mathcal{Q}} \neq 0$ by Litvin and Gutman [3], where $m_{\mathcal{F}\mathcal{Q}} = \omega^{(\mathcal{F})}/\omega^{(\mathcal{Q})}$ is the gear ratio.

The relations may be expressed by a system of three linear equations in two unknowns ${}_rV_{\mathcal{Q}_I}^{(\mathcal{F})}$ and ${}_rV_{\mathcal{Q}_{II}}^{(\mathcal{F})}$:

$$a_{j1}\ {}_rV_{\mathcal{Q}_I}^{(\mathcal{F})} + a_{j2}\ {}_rV_{\mathcal{Q}_{II}}^{(\mathcal{F})} = a_{j3} \qquad (j = 1, 2, 3) \tag{A.29}$$

where ${}_rV_{\mathcal{Q}_I}^{(\mathcal{F})}$ and ${}_rV_{\mathcal{Q}_{II}}^{(\mathcal{F})}$ are the projections of the relative velocity ${}_r\vec{V}^{(\mathcal{F})}$ at the contact point B on the principal directions on surface $\Sigma_{\mathcal{Q}}$. The equation may be represented by a symmetric augmented matrix $[A]$. That is

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{A.30}$$

Here

$$a_{11} = \kappa_{\mathcal{Q}_I} - \kappa_{\mathcal{F}_I}\cos^2\sigma - \kappa_{\mathcal{F}_{II}}\sin^2\sigma = \kappa_{\mathcal{Q}_I} - \frac{\kappa_{\mathcal{F}_I} + \kappa_{\mathcal{F}_{II}}}{2} - \frac{\kappa_{\mathcal{F}_I} - \kappa_{\mathcal{F}_{II}}}{2}\cos 2\sigma \quad \text{(A.31)}$$

$$a_{12} = a_{21} = \frac{\kappa_{\mathcal{F}_I} - \kappa_{\mathcal{F}_{II}}}{2}\sin 2\sigma \quad \text{(A.32)}$$

$$a_{13} = a_{31} = -\kappa_{\mathcal{Q}_I} V_{\mathcal{Q}_I}^{(\mathcal{F}\mathcal{Q})} - [\vec{\omega}^{(\mathcal{F}\mathcal{Q})}\vec{n}\vec{e}_{\mathcal{Q}_I}] \quad \text{(A.33)}$$

$$a_{22} = \kappa_{\mathcal{Q}_{II}} - \kappa_{\mathcal{F}_I}\sin^2\sigma - \kappa_{\mathcal{F}_{II}}\cos^2\sigma = \kappa_{\mathcal{Q}_{II}} - \frac{\kappa_{\mathcal{F}_I} + \kappa_{\mathcal{F}_{II}}}{2} - \frac{\kappa_{\mathcal{F}_I} - \kappa_{\mathcal{F}_{II}}}{2}\cos 2\sigma \quad \text{(A.34)}$$

$$a_{23} = a_{32} = -\kappa_{\mathcal{Q}_{II}} V_{\mathcal{Q}_{II}}^{(\mathcal{F}\mathcal{Q})} - [\vec{\omega}^{(\mathcal{F}\mathcal{Q})}\vec{n}\vec{e}_{\mathcal{Q}_{II}}] \quad \text{(A.35)}$$

$$\begin{aligned} a_{33} = {} & \kappa_{\mathcal{Q}_I}\left(V_{\mathcal{Q}_I}^{(\mathcal{F}\mathcal{Q})}\right)^2 + \kappa_{\mathcal{Q}_{II}}\left(V_{\mathcal{Q}_{II}}^{(\mathcal{F}\mathcal{Q})}\right)^2 - [\vec{n}\vec{\omega}^{(\mathcal{F}\mathcal{Q})}\vec{V}^{(\mathcal{F}\mathcal{Q})}] \\ & - \vec{n}\cdot\left(\vec{\omega}^{(\mathcal{F})} \times {}_{tr}\vec{V}^{(\mathcal{Q})} - \vec{\omega}^{(\mathcal{Q})} \times {}_{tr}\vec{V}^{(\mathcal{F})}\right) + \left(\omega^{(\mathcal{F})}\right)^2 m'_{\mathcal{Q}\mathcal{F}}(\vec{n} \times \vec{k}_{\mathcal{Q}}) \cdot (\vec{B} - \overline{O_{\mathcal{Q}}O_{\mathcal{F}}}) \end{aligned} \quad \text{(A.36)}$$

$\kappa_{\mathcal{F}_I}$ and $\kappa_{\mathcal{F}_{II}}$ are the principal curvatures at the contact point B of gear $\mathcal{F}$,

$\kappa_{\mathcal{Q}_I}$ and $\kappa_{\mathcal{Q}_{II}}$ are the principal curvatures at the contact point B of gear $\mathcal{Q}$,

$\vec{e}_{\mathcal{Q}_I}$ and $\vec{e}_{\mathcal{Q}_{II}}$ are the unit vectors of the principal directions at the contact point B of gear $\mathcal{Q}$,

σ is the angle measured counterclockwise from $\vec{e}_{\mathcal{F}_I}$, the unit vector of the principal direction at the contact point B of gear $\mathcal{F}$, to $\vec{e}_{\mathcal{Q}_I}$,

$\vec{\omega}^{(\mathcal{F})}$ and $\vec{\omega}^{(\mathcal{Q})}$ are the angular velocities of gears $\mathcal{F}$ and $\mathcal{Q}$, respectively,

$\vec{\omega}^{(\mathcal{F}\mathcal{Q})}$ is the relative angular velocity of gear $\mathcal{F}$ with respective to gear $\mathcal{Q}$,

$\vec{n}$ is the common unit normal vector,

$\vec{V}^{(\mathcal{F}\mathcal{Q})}$ is the relative velocity of the contact point on gear $\mathcal{F}$ with respect to the same contact point on gear $\mathcal{Q}$,

$V_{\mathcal{Q}_I}^{(\mathcal{F}\mathcal{Q})}$ and $V_{\mathcal{Q}_{II}}^{(\mathcal{F}\mathcal{Q})}$ are the projections of $\vec{V}^{(\mathcal{F}\mathcal{Q})}$ on the $\vec{e}_{\mathcal{Q}_I}$ and $\vec{e}_{\mathcal{Q}_{II}}$, respectively,

${}_{tr}\vec{V}^{(\mathcal{F})}$ and ${}_{tr}\vec{V}^{(\mathcal{Q})}$ are the transfer velocities of contact point B on gear $\mathcal{F}$ and gear $\mathcal{Q}$, respectively,

$\vec{B}$ is the position vector of the common contact point B,

$\overline{O_{\mathcal{Q}}O_{\mathcal{F}}}$ is the position vector from $O_{\mathcal{Q}}$ to $O_{\mathcal{F}}$,

$\vec{k}_{\mathcal{Q}}$ is the unit vector of the axis of rotation of gear $\mathcal{Q}$, and

$m'_{\mathcal{Q}\mathcal{F}}$ is the derivative of the rotation ratio of gear $\mathcal{Q}$ to gear $\mathcal{F}$. It is represented as

$$m'_{\mathcal{Q}\mathcal{F}} = \frac{d}{d\phi_{\mathcal{F}}} m_{\mathcal{Q}\mathcal{F}}(\phi_{\mathcal{F}})$$

where $\phi_{\mathcal{F}}$ is the rotation angle of gear $\mathcal{F}$, and

$$m_{\mathcal{Q}\mathcal{F}}(\phi_{\mathcal{F}}) = \frac{\omega^{(\mathcal{Q})}}{\omega^{(\mathcal{F})}}$$

Totally, there are two cases of tangency of gear tooth surfaces:

1. The surfaces $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{Q}}$ are in line contact and B is just a point of the instantaneous line of contact.

2. The surfaces $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{Q}}$ are in point contact and B is the single point of tangency at the considered instant.

In the case of line contact of mating surfaces, the rank of matrix $[A]$ is equal to one. Thus all determinants of the second order formed from the elements of $[A]$ are zero. This yields

$$\frac{a_{11}}{a_{12}} = \frac{a_{12}}{a_{22}} = \frac{a_{13}}{a_{23}} \tag{A.37}$$

$$\frac{a_{11}}{a_{13}} = \frac{a_{12}}{a_{23}} = \frac{a_{13}}{a_{33}} \tag{A.38}$$

$$\frac{a_{12}}{a_{13}} = \frac{a_{22}}{a_{23}} = \frac{a_{23}}{a_{33}} \tag{A.39}$$

Using equations (A.31)–(A.39) we obtain

$$\tan 2\sigma = \frac{2a_{13}a_{23}}{a_{23}^2 - a_{13}^2 + (\kappa_{\mathcal{Q}_I} - \kappa_{\mathcal{Q}_{II}})a_{33}} \tag{A.40}$$

$$\kappa_{\mathcal{F}_I} - \kappa_{\mathcal{F}_{II}} = \frac{a_{23}^2 - a_{13}^2 + (\kappa_{\mathcal{Q}_I} - \kappa_{\mathcal{Q}_{II}})a_{33}}{a_{33}\cos 2\sigma} \tag{A.41}$$

$$\kappa_{\mathcal{F}_I} + \kappa_{\mathcal{F}_{II}} = (\kappa_{\mathcal{Q}_I} + \kappa_{\mathcal{Q}_{II}}) - \frac{a_{13}^2 + a_{23}^2}{a_{33}} \tag{A.42}$$

For the case when surfaces $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{Q}}$ are in point contact, the rank of matrix [A] described by equation (A.30) is two. Consequently,

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{13} \end{vmatrix} = 0 \tag{A.43}$$

Equation (A.43) yields the relation

$$f(\kappa_{\mathcal{F}_I}, \kappa_{\mathcal{F}_{II}}, \kappa_{\mathcal{Q}_I}, \kappa_{\mathcal{Q}_{II}}, \sigma) = 0 \tag{A.44}$$

In general, equations of the generated surface are evidently much more complicated than those of the generating one. Therefore, a direct way to obtain the principal curvatures and directions of the generated surface is a very difficult task. This work can be significantly simplified if we apply the relations, described in this section, between principal curvatures and directions of meshing surfaces.

A.3 Relative Normal Curvature

The relative normal curvature, κ_r, of two mating surface, $\Sigma_{\mathcal{F}}$ and $\Sigma_{\mathcal{Q}}$, at the contact point B is defined as the difference of the normal curvatures of both surfaces taken in a common normal section of surfaces and represented as

$$\kappa_r = \kappa_n^{(\mathcal{Q})} - \kappa_n^{(\mathcal{F})} \tag{A.45}$$

Suppose the common normal section form an angle ϖ with the unit vector $\vec{e}_{\mathcal{Q}_I}$ and an angle $(\varpi + \sigma)$ with the unit vector $\vec{e}_{\mathcal{F}_I}$ (Figure 23). According to *Euler's Theorem* (equation (A.28)), we obtain

$$\kappa_n^{(\mathcal{Q})} = \kappa_{\mathcal{Q}_I} \cos^2 \varpi + \kappa_{\mathcal{Q}_{II}} \sin^2 \varpi \qquad \kappa_n^{(\mathcal{F})} = \kappa_{\mathcal{F}_I} \cos^2(\varpi + \sigma) + \kappa_{\mathcal{F}_{II}} \sin^2(\varpi + \sigma) \tag{A.46}$$

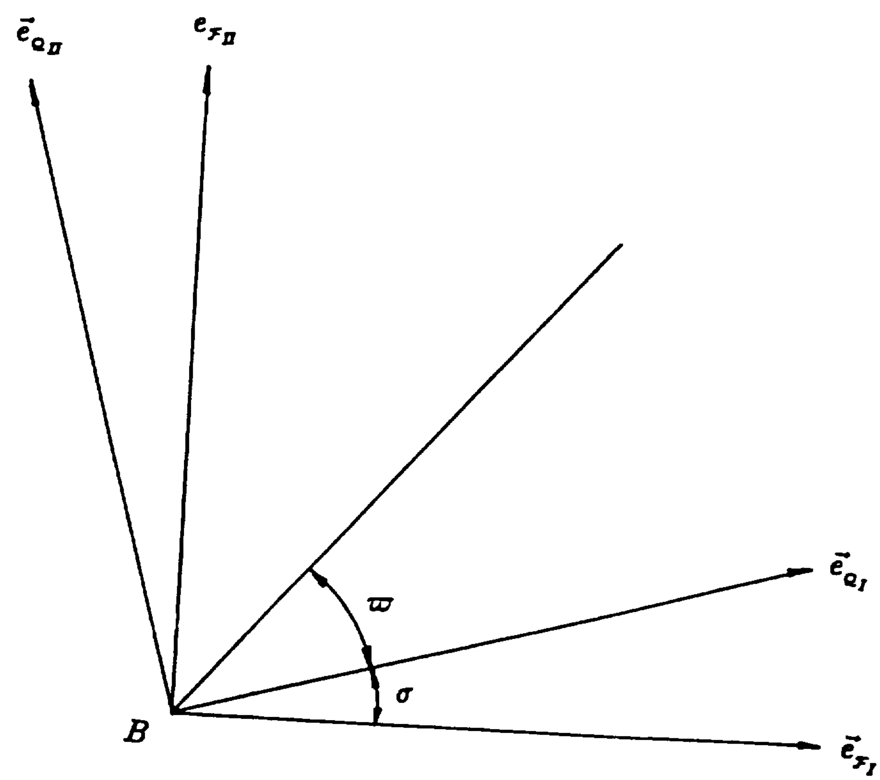


Figure 23: A tangent plane to a surface

Substituting equation (A.46) into (A.45), after simple transformations we get

$$\begin{aligned}\kappa_r &= (\kappa_{\mathcal{Q}_I} - \kappa_{\mathcal{F}_I}\cos^2\sigma - \kappa_{\mathcal{F}_{II}}\sin^2\sigma)\cos^2\varpi + (\kappa_{\mathcal{Q}_{II}} - \kappa_{\mathcal{F}_I}\sin^2\sigma - \kappa_{\mathcal{F}_{II}}\cos^2\sigma)\sin^2\varpi \\ &\quad + \frac{1}{2}(\kappa_{\mathcal{F}_I} - \kappa_{\mathcal{F}_{II}})\sin 2\sigma \sin 2\varpi\end{aligned} \tag{A.47}$$

Equation (A.47) and expressions for a_{11}, a_{12}, and a_{22} in equations (A.31), (A.32), and (A.34) yield

$$\kappa_r = \frac{1}{2}\left[a_{11} + a_{22} + (a_{11} - a_{22})\cos 2\varpi\right] + a_{12}\sin 2\varpi \tag{A.48}$$

The extreme values of function $\kappa_r(\varpi)$ may be determined by

$$\frac{d}{d\varpi}(\kappa_r) = 0 \tag{A.49}$$

Thus we obtain

$$\tan 2\varpi = \frac{2a_{12}}{a_{11} - a_{22}} \tag{A.50}$$

This equation has two solutions ϖ_1 and ϖ_2. Moreover, $|\varpi_1 - \varpi_2| = \pi/2$. This means that there are two perpendicular directions for the extreme relative normal curvatures. Using equations (A.48) and (A.50) the extreme values of the relative normal curvatures are represented by

$$\kappa_r = \frac{1}{2}\left[(a_{11} + a_{22}) \pm \sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}\right] \tag{A.51}$$

We may determine whether or not two surfaces interfere each other by the concept of relative normal curvatures. If two surfaces contact at a point with any interference, the sign of the relative normal curvature in each direction must remain the same. In other words, the product of two extreme values of the relative normal curvatures is positive. Equations discussed in this section were first proposed by Litvin [9].

A.4 Contact Ellipse

In theory the tooth surfaces of a pair of spiral bevel gears are in contact at a single point at every instant. In practice the surface of the solids is deformed elastically over a region surrounding the initial point of contact, thereby bring the two bodies into contact over a small area in the neighborhood of the initial contact point [13, 14]. Such an area is an ellipse whose center of symmetry is the theoretical point of contact and the dimensions depend on the elastic approach and principal curvatures and directions of the contacting surfaces. If the approach of surfaces under the action of load is given, the size and orientation of the contact ellipse can be defined as a result of a geometric solution. Litvin[9, 15] has investigated the mathematical modeling of the contact ellipse.

Let us now consider that two surfaces Σ_1 and Σ_2 are contact at a single point B. The principal curvatures, κ_{1_I} and $\kappa_{1_{II}}$ of Σ_1 and κ_{2_I} and $\kappa_{2_{II}}$ of Σ_2, at point B are known. Also known are unit vectors $\vec{e}_{1_I}$ and $\vec{e}_{1_{II}}$, which are directed along the principal directions of Σ_1 at point B, and $\vec{e}_{2_I}$ and $\vec{e}_{2_{II}}$, which are directed along the principal directions of Σ_2 at point B. Unit vectors $\vec{e}_{1_I}$ and $\vec{e}_{2_I}$ determine the tangent plane (Figure 24). Angle σ_{12}, which is measured counterclockwise from $\vec{e}_{1_I}$ to $\vec{e}_{2_I}$, is also determined since $\vec{e}_{1_I}$ and $\vec{e}_{2_I}$ have already been known. Then the contact ellipse may be described as

$$\frac{\zeta^2}{a^2} + \frac{\eta^2}{b^2} = 1 \tag{A.52}$$

in which ζ and η are coordinates with respect to the ζ and η axes with origin at the contact point B. The lengths of semiaxes a and b are

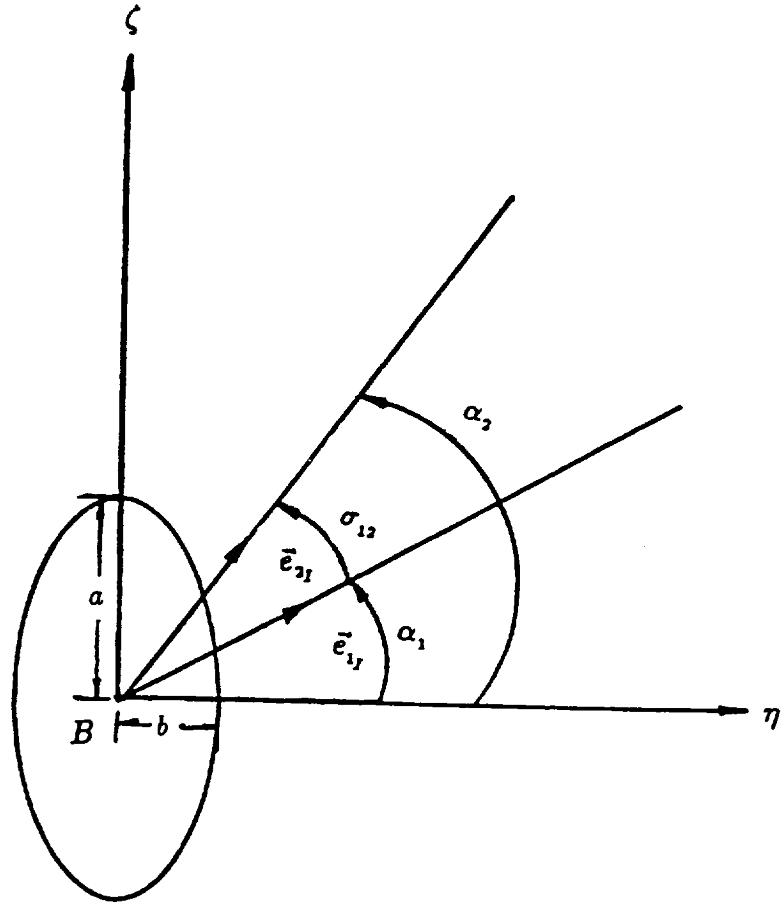


Figure 24: Contact ellipse on the tangent plane.

$$a = \sqrt{\left|\frac{\varepsilon}{\mathcal{A}}\right|}, \qquad b = \sqrt{\left|\frac{\varepsilon}{\mathcal{B}}\right|} \tag{A.53}$$

where ε is the approach, and

$$\mathcal{A} = \frac{1}{4}\left(\kappa_{1\Sigma} - \kappa_{2\Sigma} - \sqrt{\kappa_{1\Delta}^2 - 2\kappa_{1\Delta}\kappa_{2\Delta}\cos 2\sigma_{12} + \kappa_{2\Delta}^2}\right) \tag{A.54}$$

$$\mathcal{B} = \frac{1}{4}\left(\kappa_{1\Sigma} - \kappa_{2\Sigma} + \sqrt{\kappa_{1\Delta}^2 - 2\kappa_{1\Delta}\kappa_{2\Delta}\cos 2\sigma_{12} + \kappa_{2\Delta}^2}\right) \tag{A.55}$$

where

$$\kappa_{1\Sigma} = \kappa_{1_I} + \kappa_{1_{II}}, \qquad \kappa_{2\Sigma} = \kappa_{2_I} + \kappa_{2_{II}} \tag{A.56}$$

$$\kappa_{1\Delta} = \kappa_{1_I} - \kappa_{1_{II}}, \qquad \kappa_{2\Delta} = \kappa_{2_I} - \kappa_{2_{II}} \tag{A.57}$$

The angle α_1 which determines the orientation of the ellipse may be obtained by equations

$$\cos 2\alpha_1 = \frac{\kappa_{1\Delta} - \kappa_{2\Delta}\cos 2\sigma_{12}}{\sqrt{\kappa_{1\Delta}^2 - 2\kappa_{1\Delta}\kappa_{2\Delta}\cos 2\sigma_{12} + \kappa_{2\Delta}^2}} \tag{A.58}$$

and

$$\sin 2\alpha_1 = \frac{2\kappa_{2\Delta}\sin 2\sigma_{12}}{\sqrt{\kappa_{1\Delta}^2 - 2\kappa_{1\Delta}\kappa_{2\Delta}\cos 2\sigma_{12} + \kappa_{2\Delta}^2}} \tag{A.59}$$

Finally

$$\alpha_1 = \arctan \frac{\sin 2\alpha_1}{1 + \cos 2\alpha_1} \tag{A.60}$$

Note that the angle α_1 is measured counterclockwise from the η axis to the unit vector $\vec{e}_{1_I}$.

Since

$$\mathcal{A}^2 - \mathcal{B}^2 = \frac{1}{4}(\kappa_{2\Sigma} - \kappa_{1\Sigma})\sqrt{\kappa_{1\Delta}^2 - 2\kappa_{1\Delta}\kappa_{2\Delta}\cos 2\sigma_{12} + \kappa_{2\Delta}^2}$$

the semimajor axis of the contact ellipse may be determined by the following conditions:

- The length of the semimajor axis, which is along the η axis, is b

  if $\kappa_{2\Sigma} > \kappa_{1\Sigma}$ or $|\mathcal{A}| > |\mathcal{B}|$.

- The length of the semimajor axis, which is along the ζ axis, is a

  if $\kappa_{1\Sigma} > \kappa_{2\Sigma}$ or $|\mathcal{B}| > |\mathcal{A}|$.

APPENDIX B

NUMERICAL EXAMPLES

In this section, we will use the synthesis method discussed in Chapter 3 to determine the machine-tool settings for a pair of spiral bevel gear drive, and then we will use the TCA to simulate the meshing of this pair under alignment and misalignment conditions. Two cases are considered here. Both cases use straight blades to cut gears, but for the pinion, case 1 uses straight blades, and case 2 uses curved blades.

The major blank data is represented in Table 2. Table 3 shows the input for case 1, and Table 4 shows the input for case 2. The output for the gear machine-tool settings is shown in Table 5, which is the same for both cases. For the pinion machine-tool settings, case 1 is shown in Table 6, and case 2 is shown in Table 7.

Two conditions of misalignment are considered when the TCA is applied to simulate the meshing. They are the shift of pinion along its axis, which is denoted by $\triangle A$, and the error of pinion shaft offset, which denoted by $\triangle V$. We consider that $\triangle A$ is positive when the mounting distance of pinion is increased. The sense of $\triangle V$ is the same as y_f shown in Figure 18. The output of the TCA is shown from Figure 25 to Figure 34 for case 1, and from Figure 35 to Figure 44 for case 2, respectively.

Table 2: BLANK DATA.

	Pinion	Gear
Number of Teeth	10	41
Diametral Pitch	5.559	
Shaft Angle	90°	
Mean Cone Distance	3.226	
Outer Cone Distance	3.796	
Whole Depth	0.335	
Working Depth	0.302	
Clearance	0.033	
Face Width	1.139	
Root Cone Angle	12°1′	72°25′
Mean Spiral Angle	—	35°
Hand of Spiral	R.H.	L.H.

Table 3: INPUT DATA FOR CASE 1.

	Gear Convex Side	Gear Concave Side
Gear Blade Angle	20°	
Gear Cutter Average Diameter	6	
Gear Cutter Point Width	0.08	
First Derivative of Gear Ratio	-0.0035	0.0052
Semimajor Axis of Contact Ellipse	0.171	0.181
Contact Path Direction Angle	90°	75°

Table 4: INPUT DATA FOR CASE 2.

	Gear Convex Side	Gear Concave Side
Gear Blade Angle	20°	
Gear Cutter Average Diameter	6	
Gear Cutter Point Width	0.08	
First Derivative of Gear Ratio	-0.0037	0.0055
Semimajor Axis of Contact Ellipse	0.171	0.171
Contact Path Direction Angle	90°	75°
Radius of Blade	40.	50.

Table 5: GEAR MACHINE-TOOL SETTINGS.

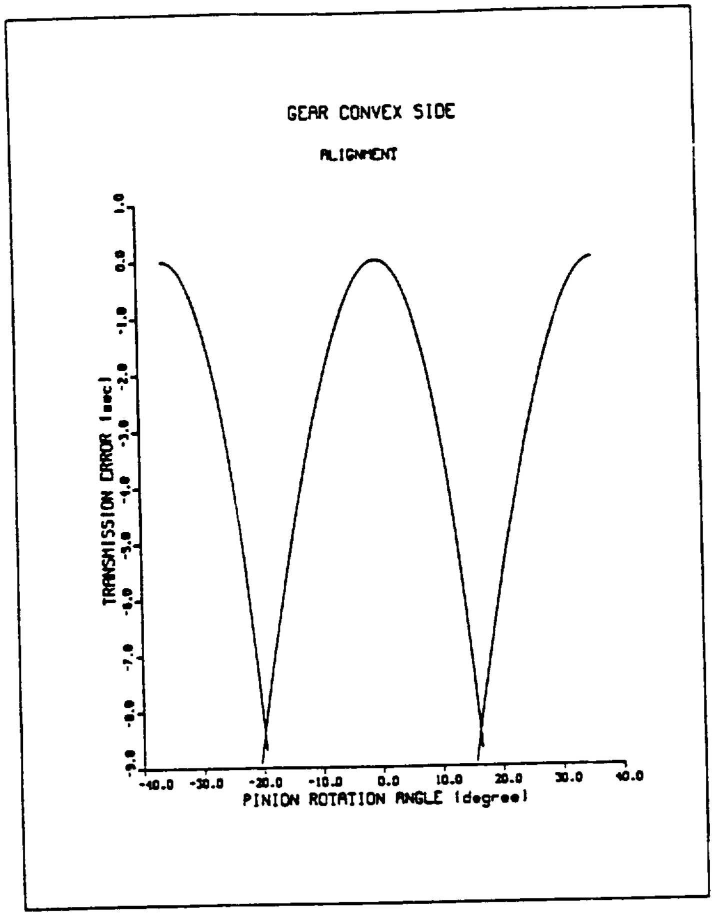
Radial	2.87798
Cradle Angle	58.6365
Ratio of Roll	0.973748

Table 6: PINION MACHINE-TOOL SETTINGS WITH STRAIGHT BLADE.

	Pinion Concave Side	Pinion Convex Side
Blade Angle	16.5561°	22.9907°
Tip Radius of Cutter	2.96469	3.07037
Radial	2.99331	2.69783
Cradle Angle	63.1869°	54.1910°
Ratio of Roll	0.22900	0.25348
Machining Offset	0.17404	-0.24459
Machine Center to Back + Sliding Base	0.021231	0.052118

Table 7: PINION MACHINE-TOOL SETTINGS WITH CURVED BLADE.

	Pinion Concave Side	Pinion Convex Side
Blade Angle	16.5561°	22.9907°
Blade Center	(11.557, 0., -35.309)	(19.685, 0., 49.006)
Tip Radius of Cutter	2.98467	3.04386
Radial	2.95578	2.74261
Cradle Angle	63.0025°	54.0900°
Ratio of Roll	0.23157	0.24915
Machining Offset	0.12042	-0.18825
Machine Center to Back + Sliding Base	0.01690	0.03605



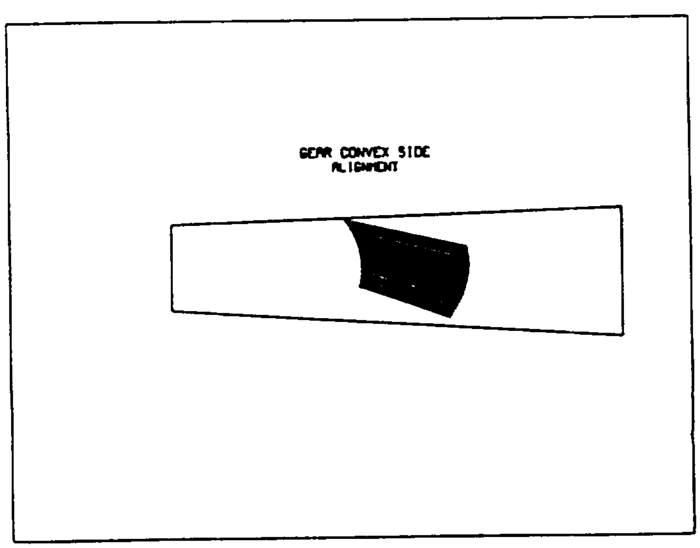
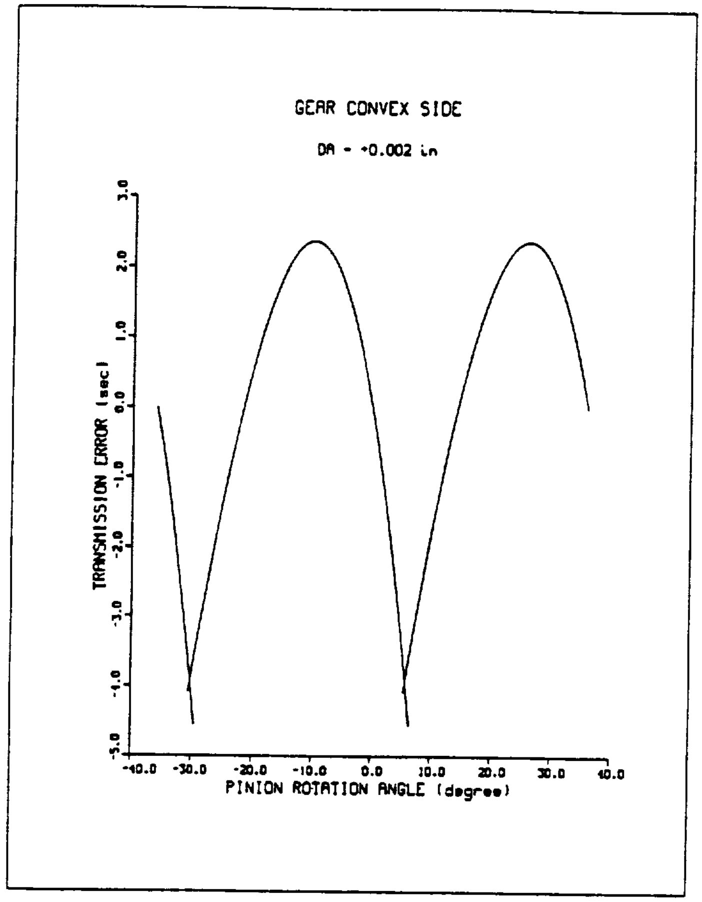


Figure 25: Straight-edged blade, gear convex side, alignment.



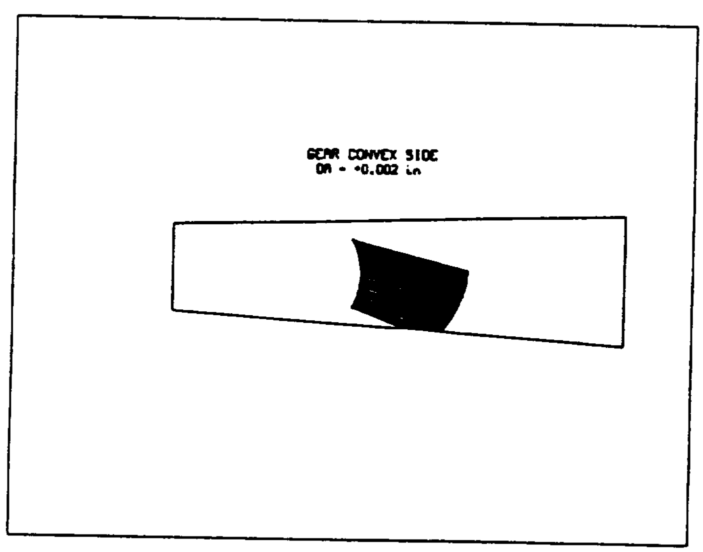
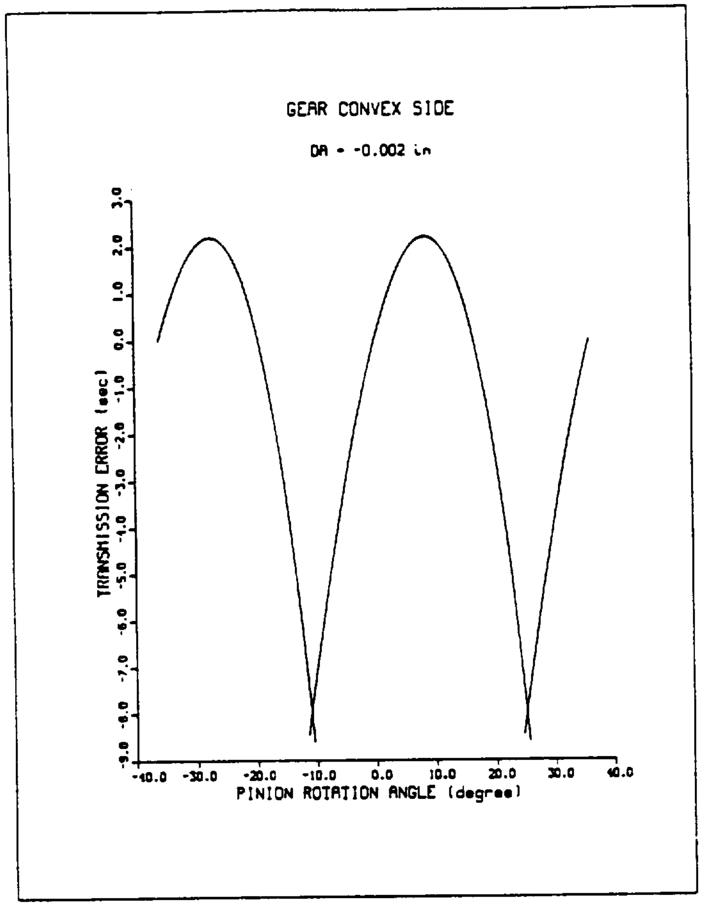


Figure 26: Straight-edged blade, gear convex side, $\triangle A = +0.002$ inches.



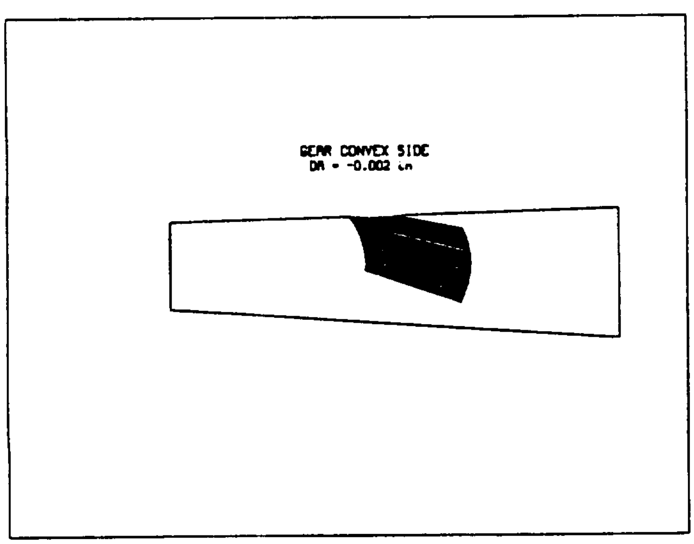
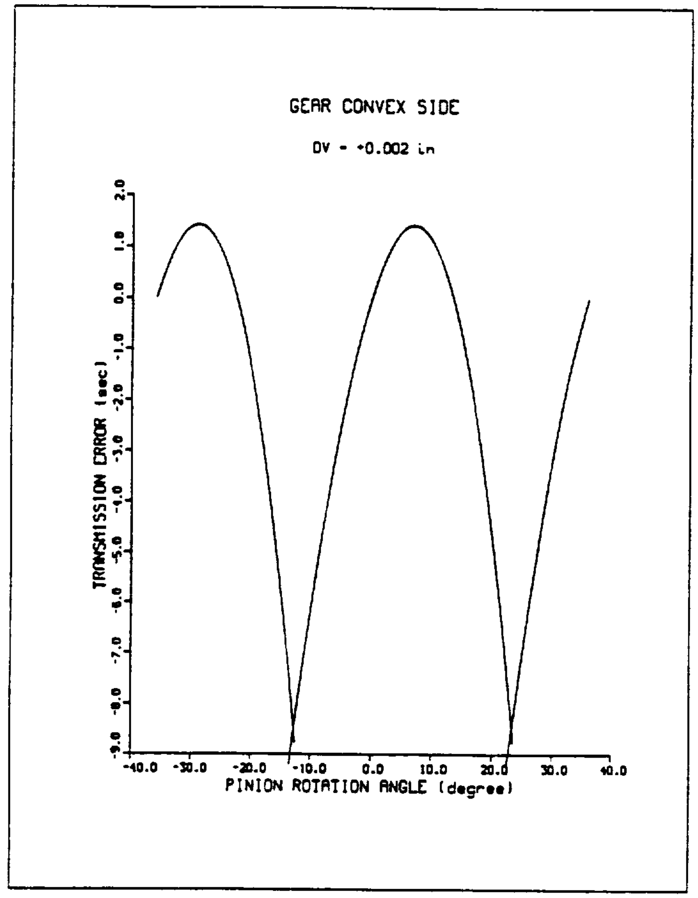


Figure 27: Straight-edged blade, gear convex side, $\triangle A = -0.002$ inches.



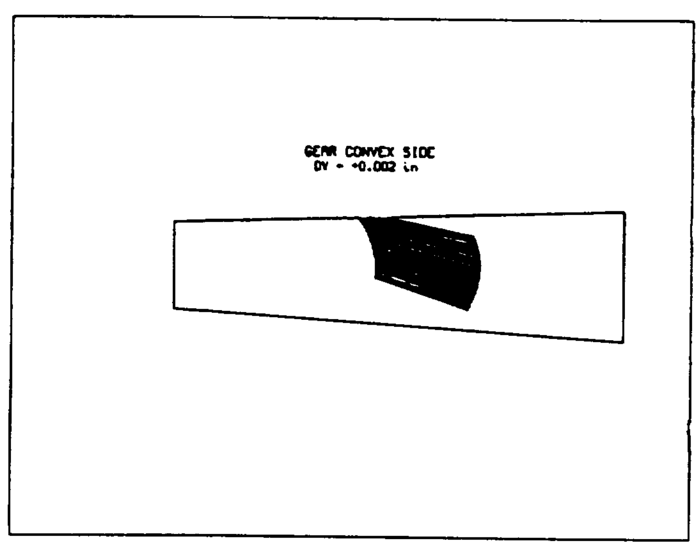
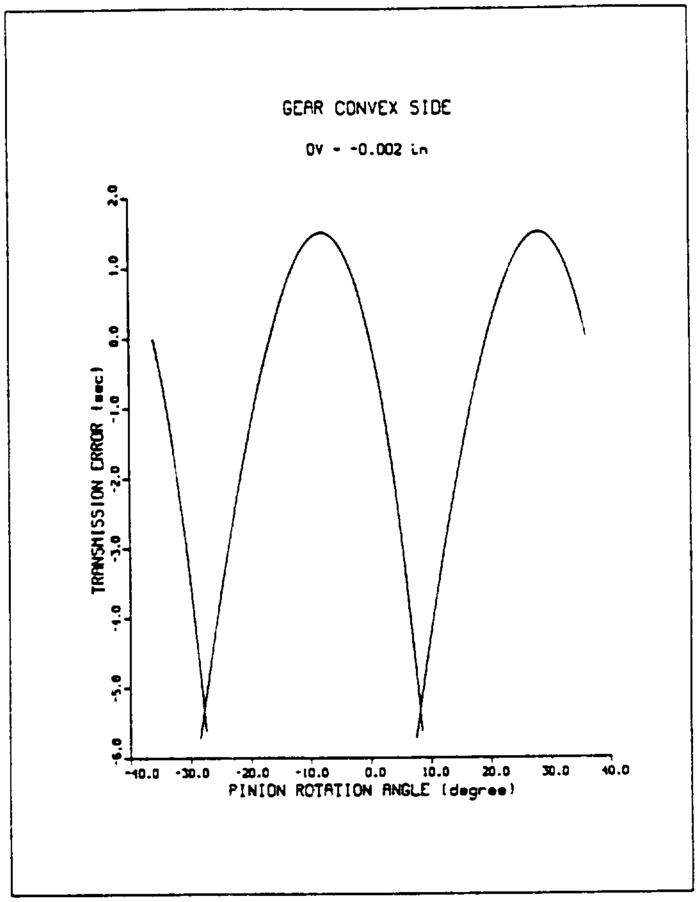


Figure 28: Straight-edged blade, gear convex side, $\triangle V = +0.002$ inches.



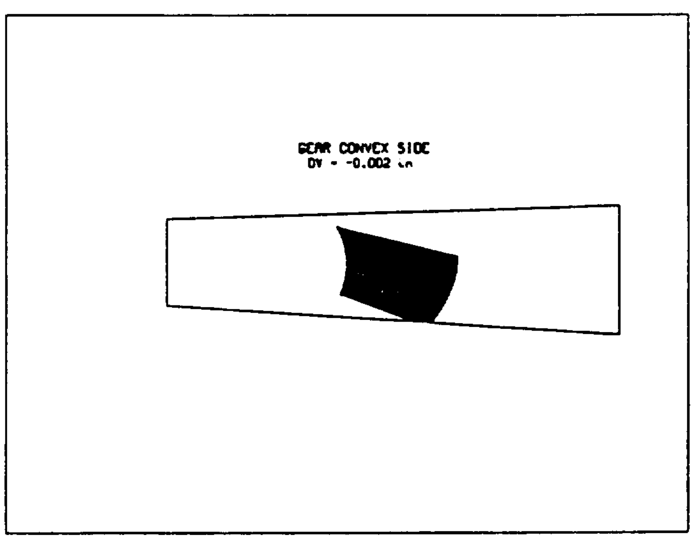
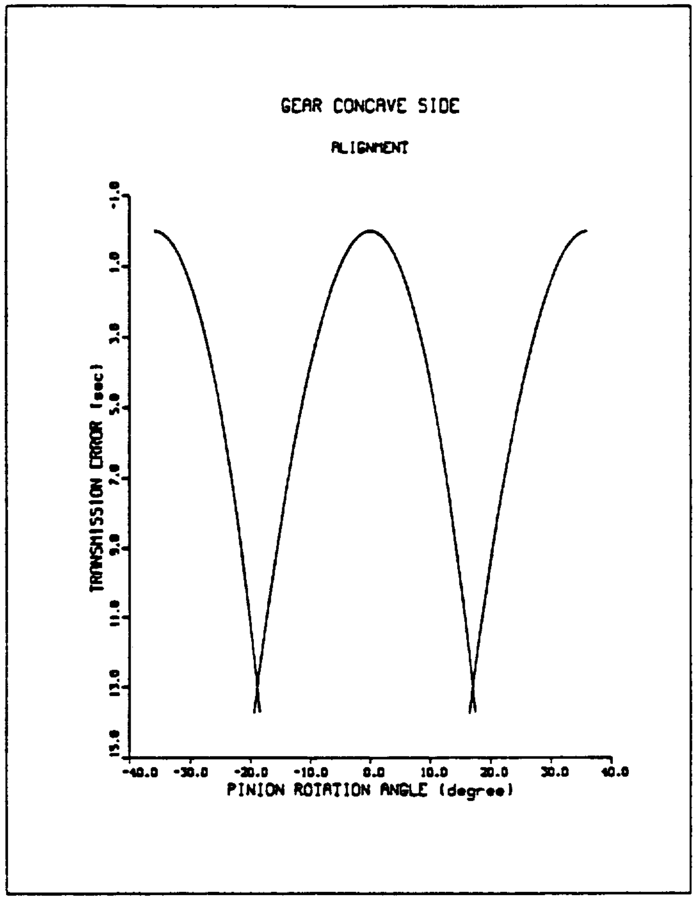


Figure 29: Straight-edged blade, gear convex side, $\triangle V = -0.002$ inches.



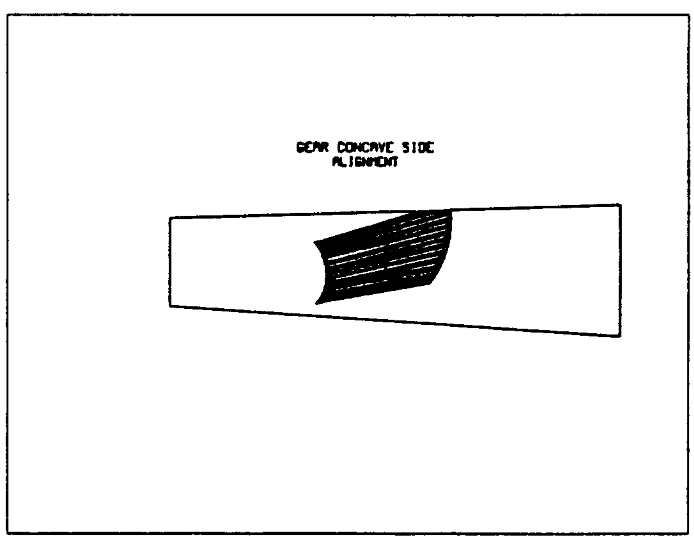
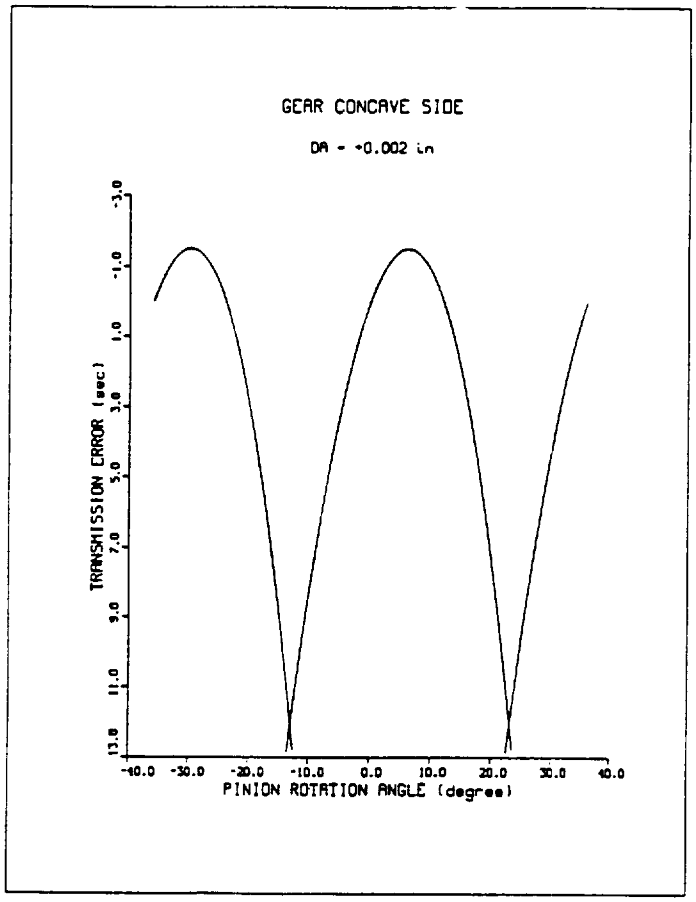


Figure 30: Straight-edged blade, gear concave side, alignment.



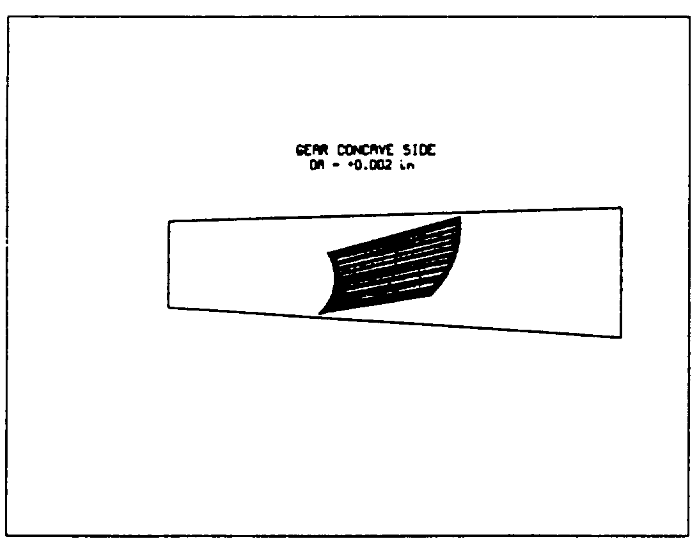
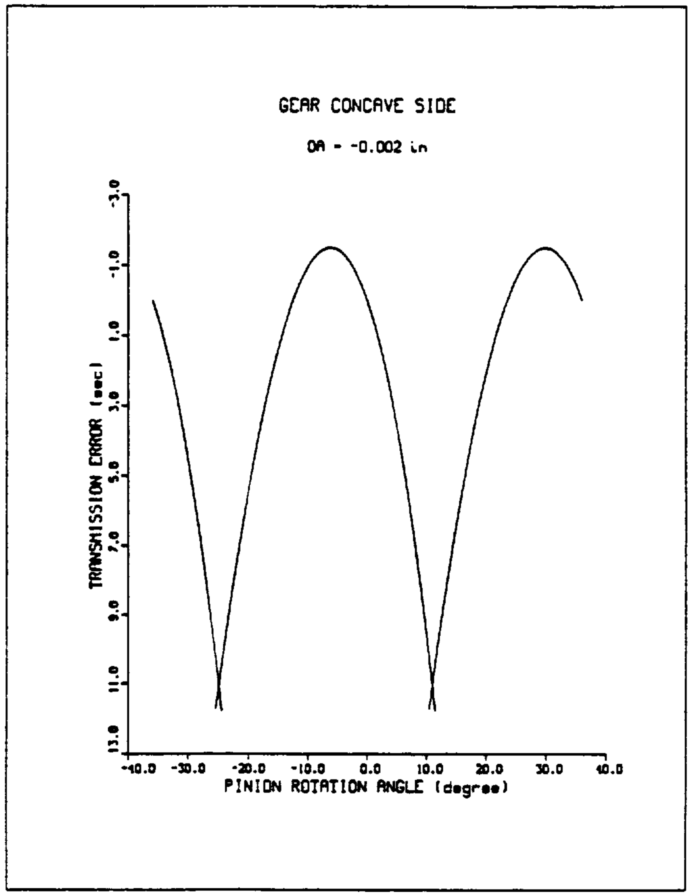


Figure 31: Straight-edged blade, gear concave side, $\triangle A = +0.002$ inches.



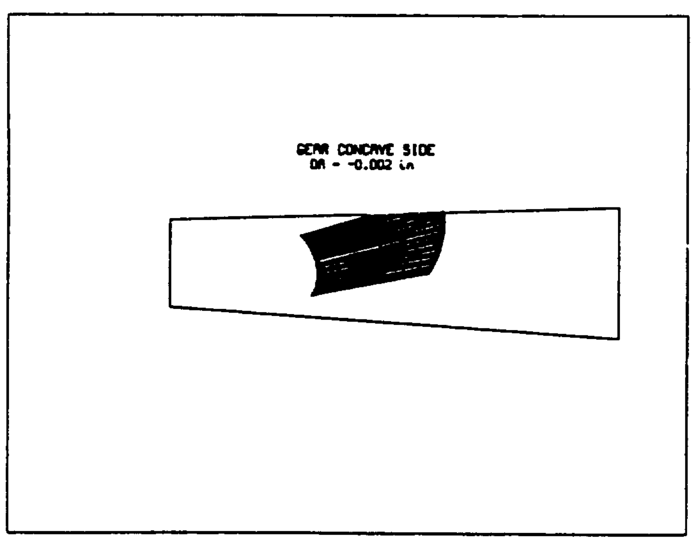
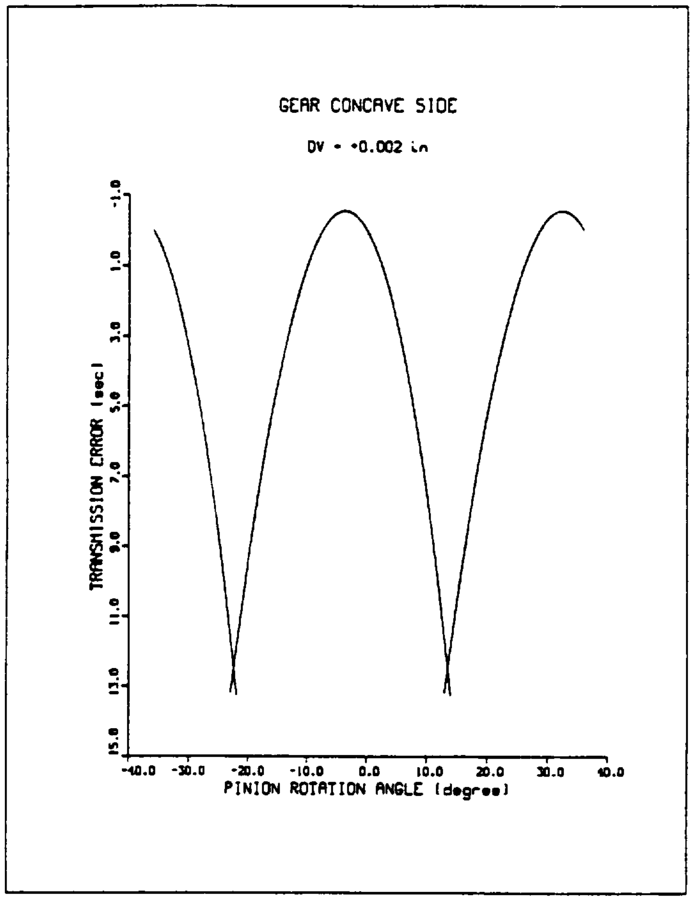


Figure 32: Straight-edged blade, gear concave side, $\triangle A = -0.002$ inches.



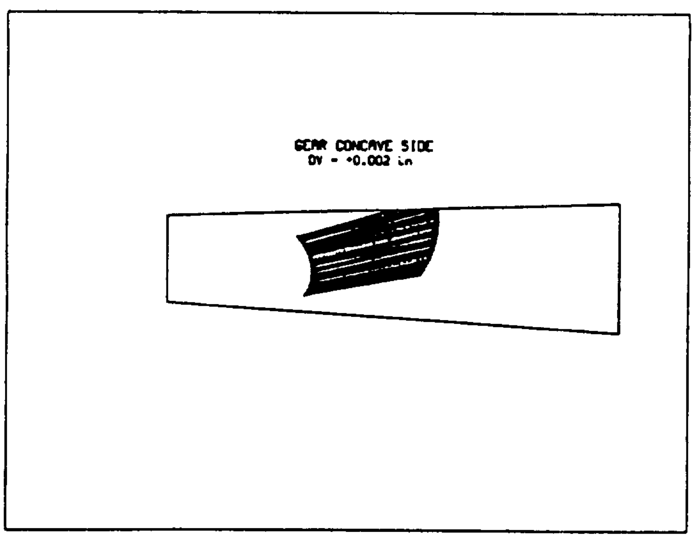
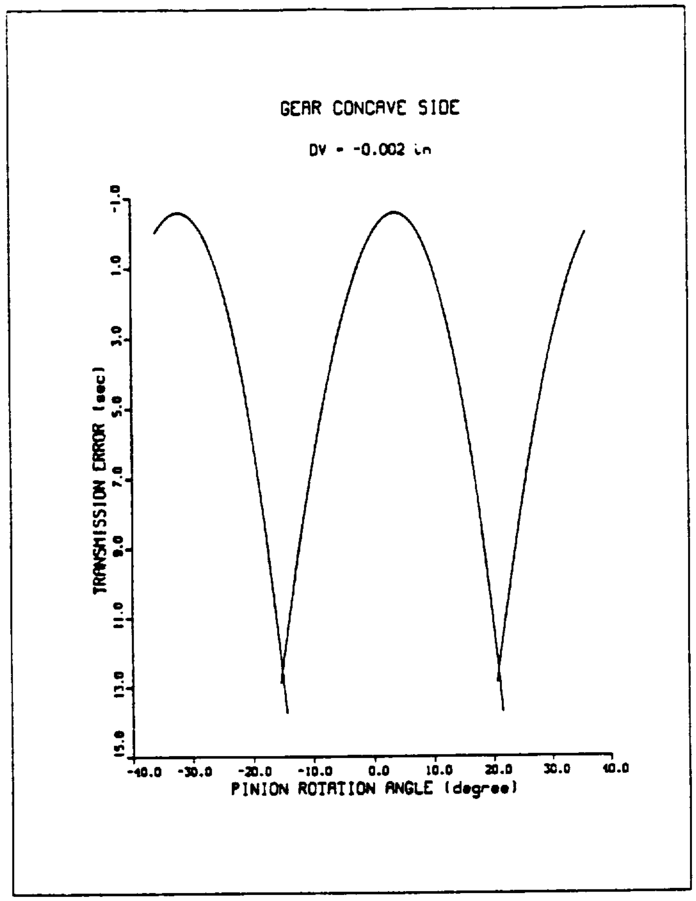


Figure 33: Straight-edged blade, gear concave side, $\triangle V = +0.002$ inches.



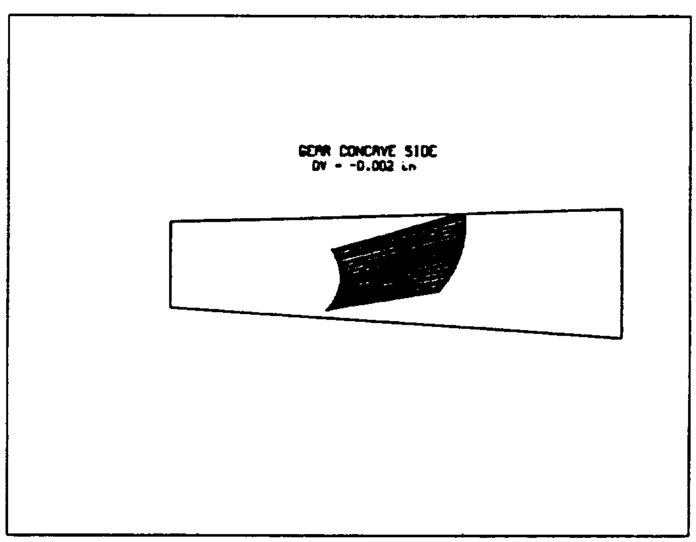
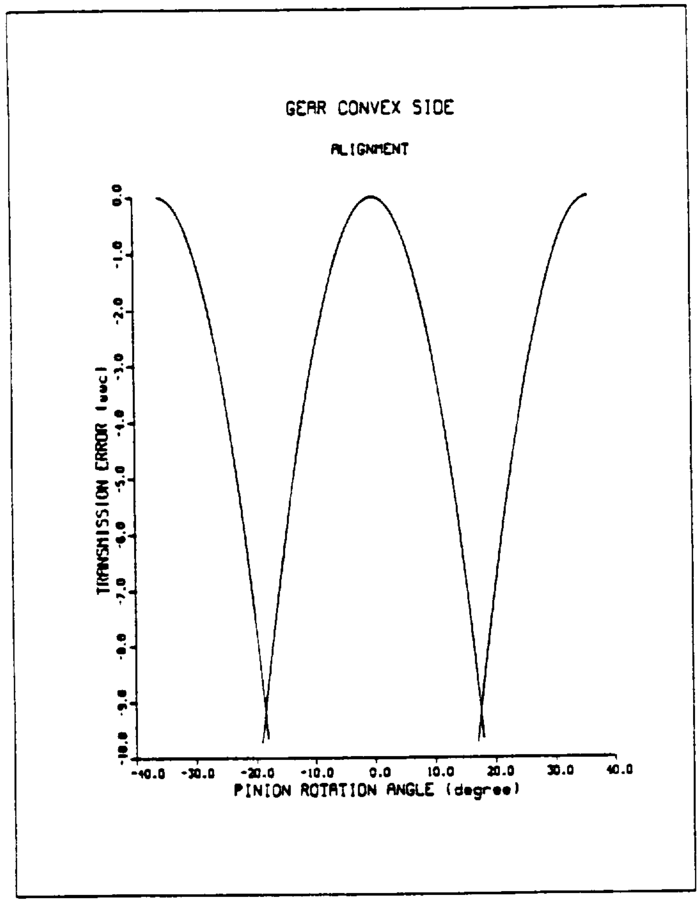


Figure 34: Straight-edged blade, gear concave side, $\triangle V = -0.002$ inches.



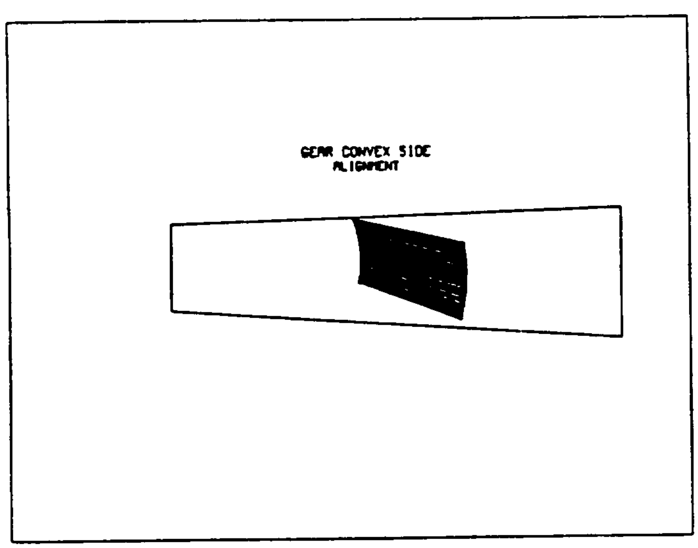
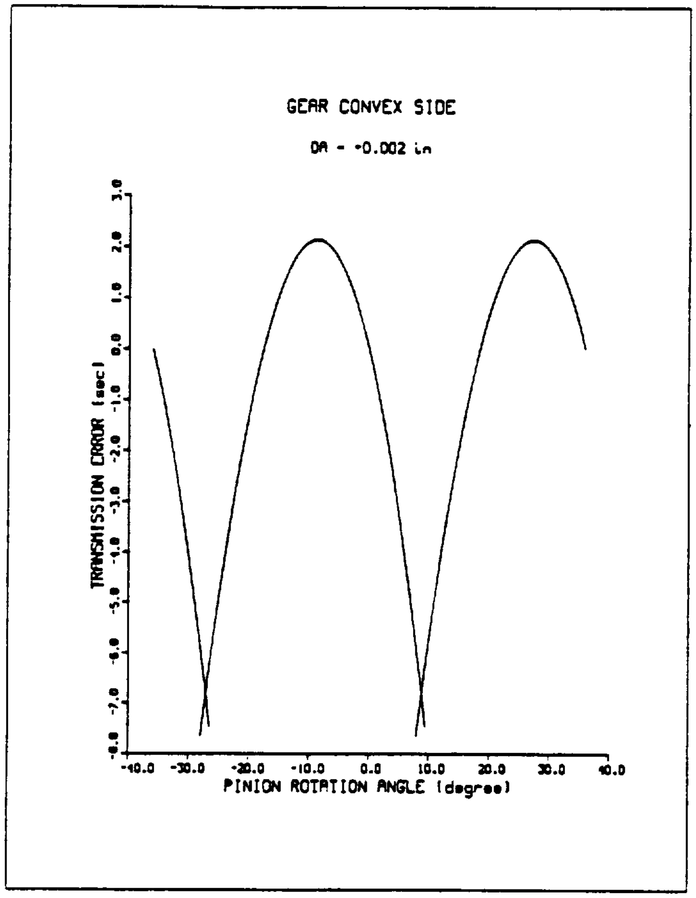


Figure 35: Curved-edged blade, gear convex side, alignment.



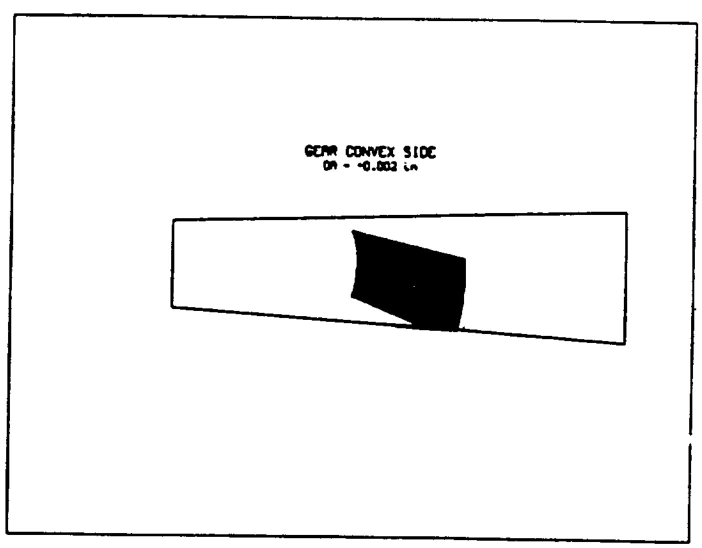
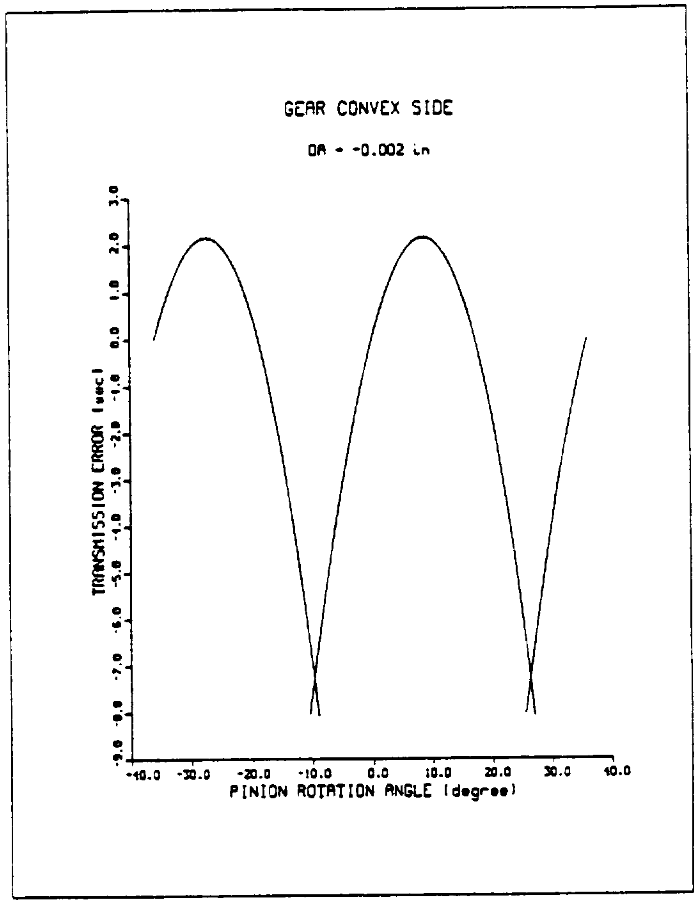


Figure 36: Curved-edged blade, gear convex side, $\triangle A = +0.002$ inches.



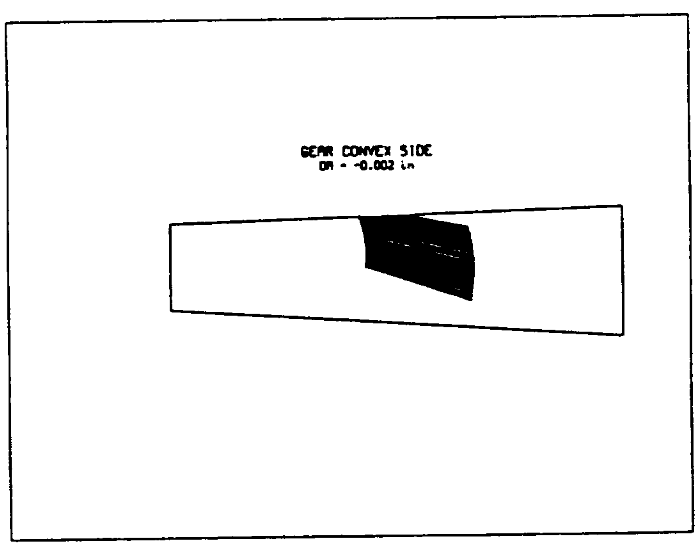
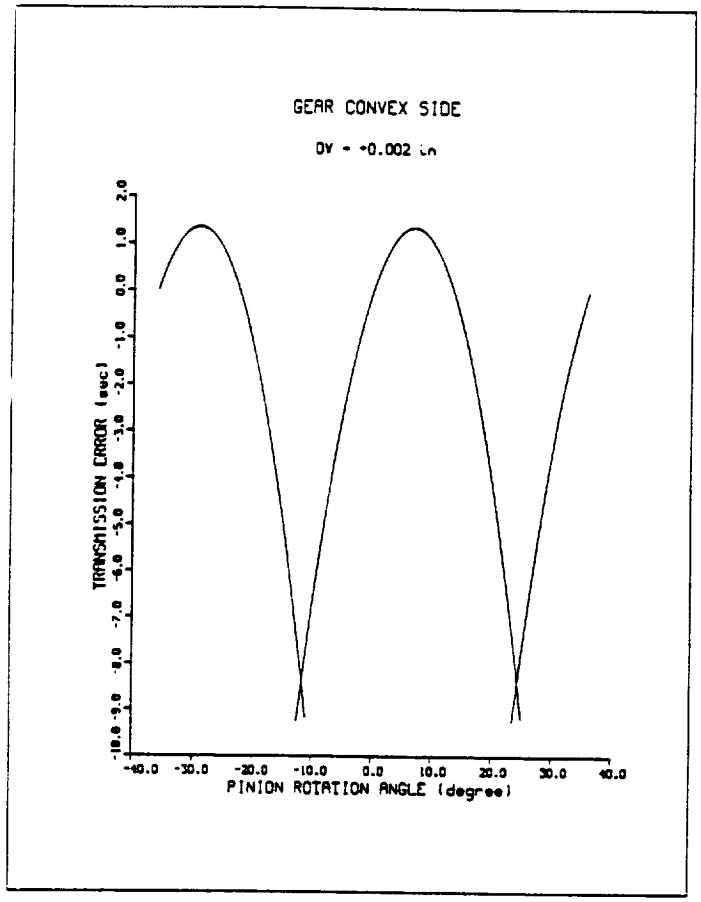


Figure 37: Curved-edged blade, gear convex side, $\triangle A = -0.002$ inches.



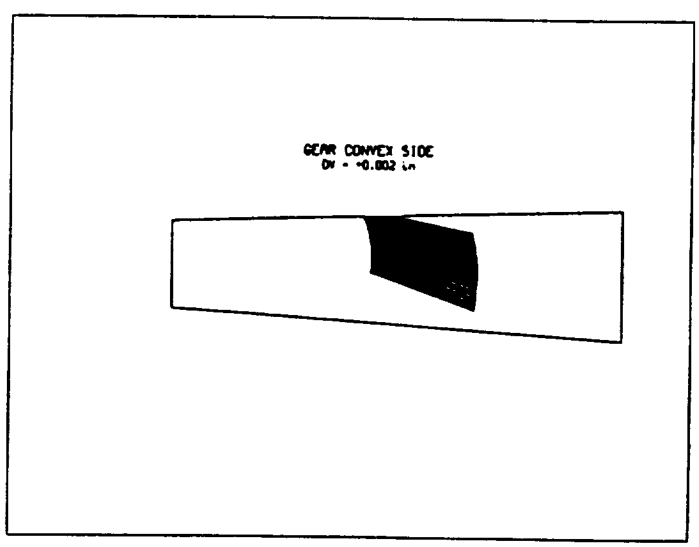
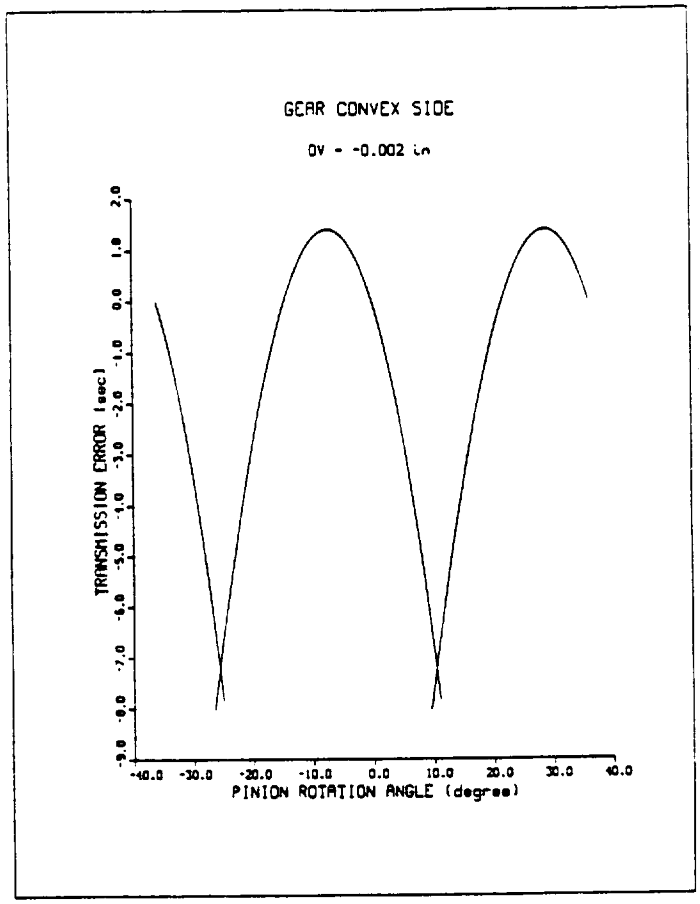


Figure 38: Curved-edged blade, gear convex side, $\triangle V = +0.002$ inches.



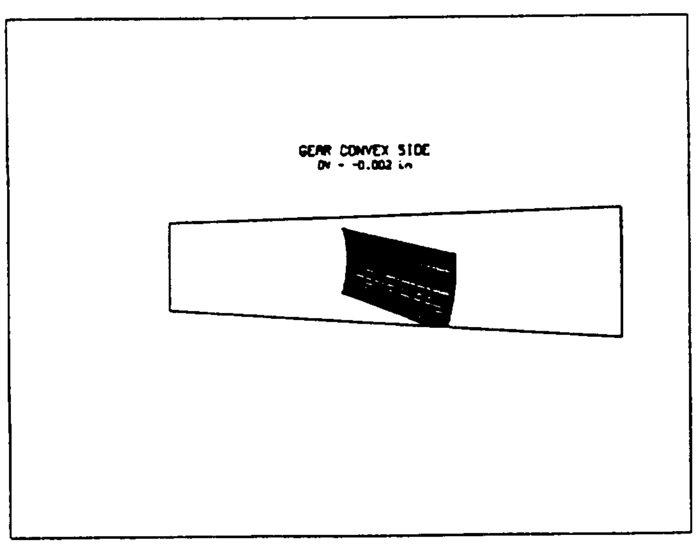
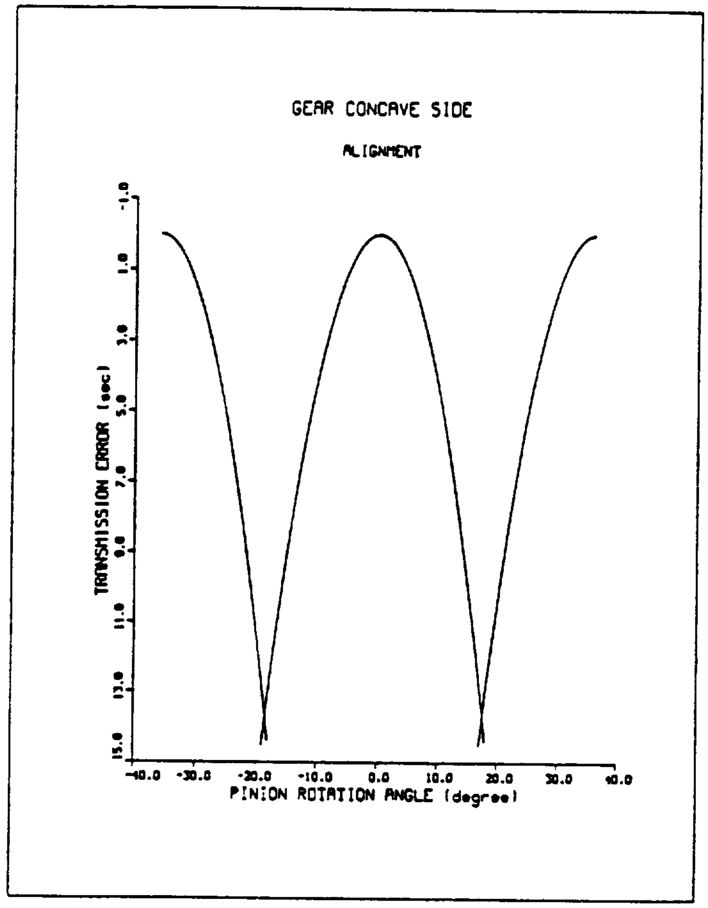


Figure 39: Curved-edged blade, gear convex side, $\triangle V = -0.002$ inches.



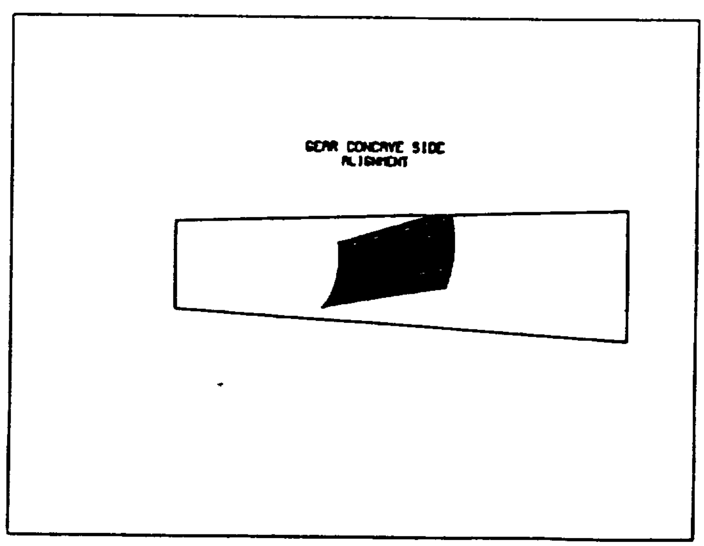
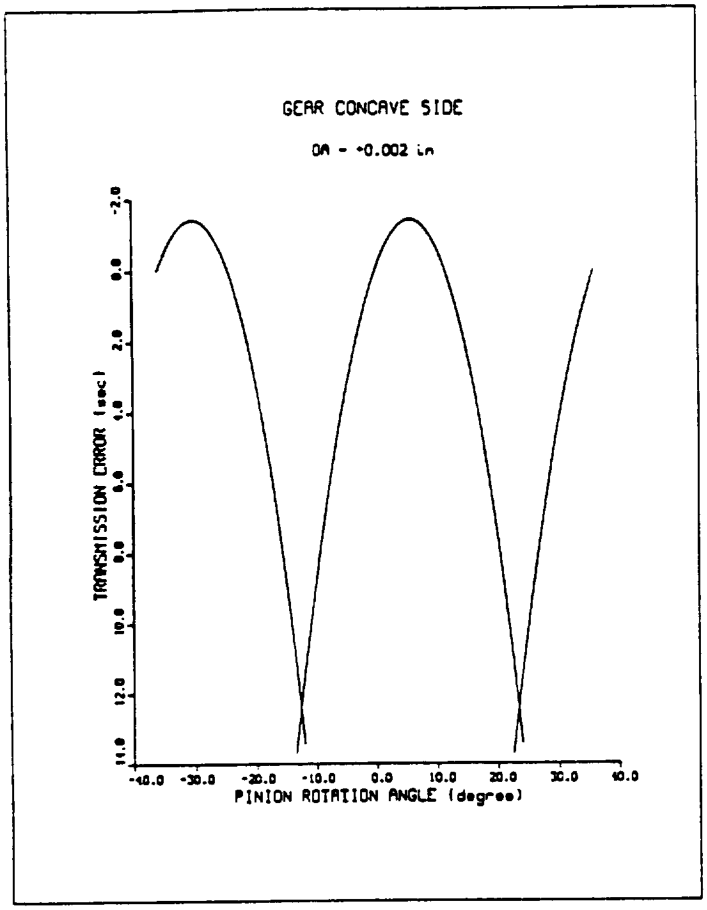


Figure 40: Curved-edged blade, gear concave side, alignment.



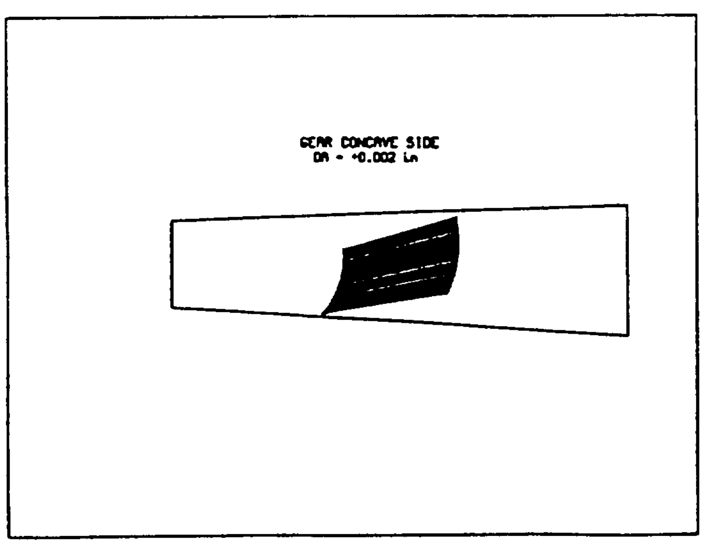
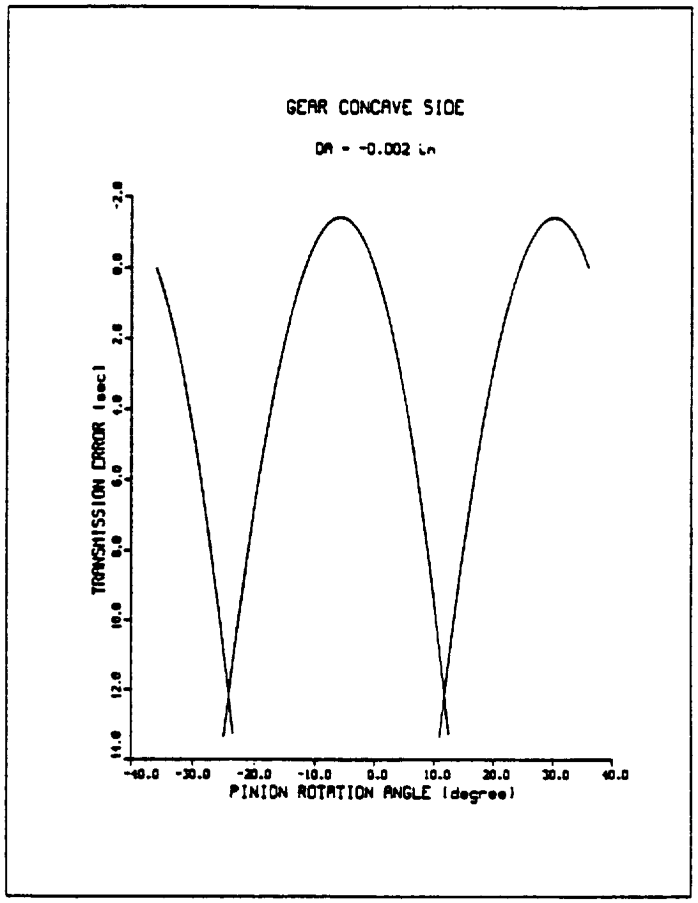


Figure 41: Curved-edged blade, gear concave side, $\triangle A = +0.002$ inches.



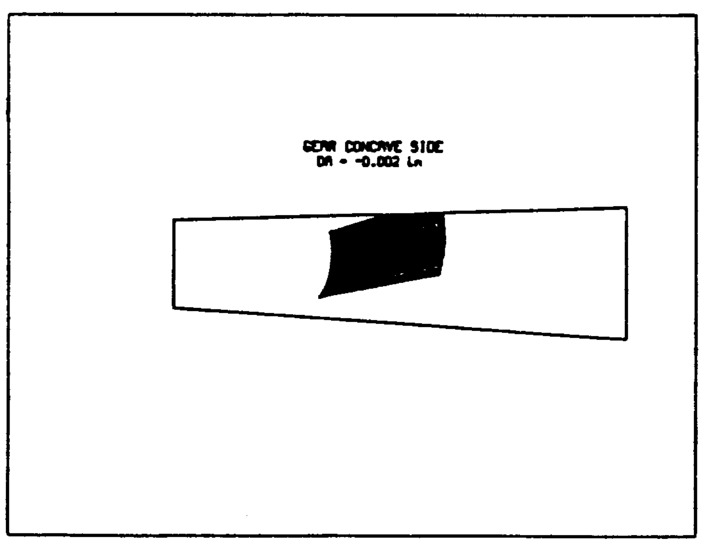
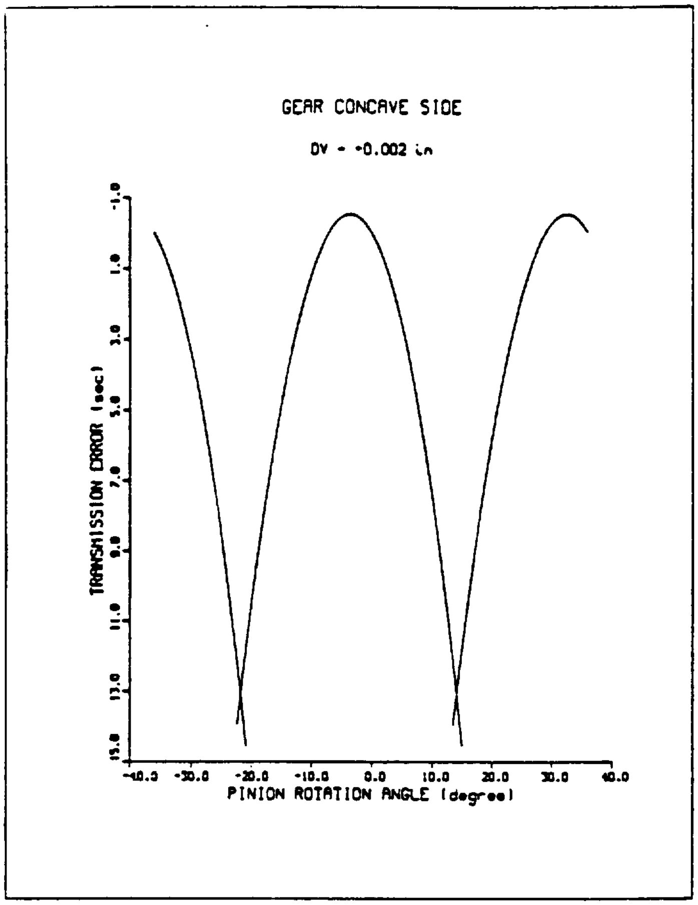


Figure 42: Curved-edged blade, gear concave side, $\triangle A = -0.002$ inches.



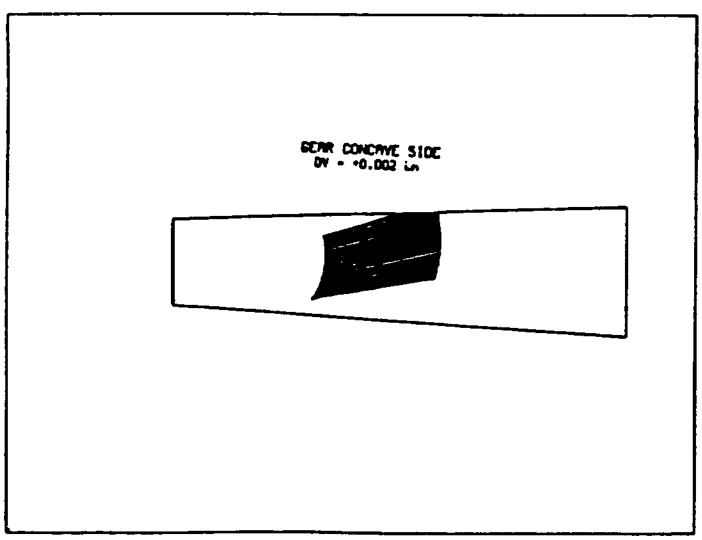
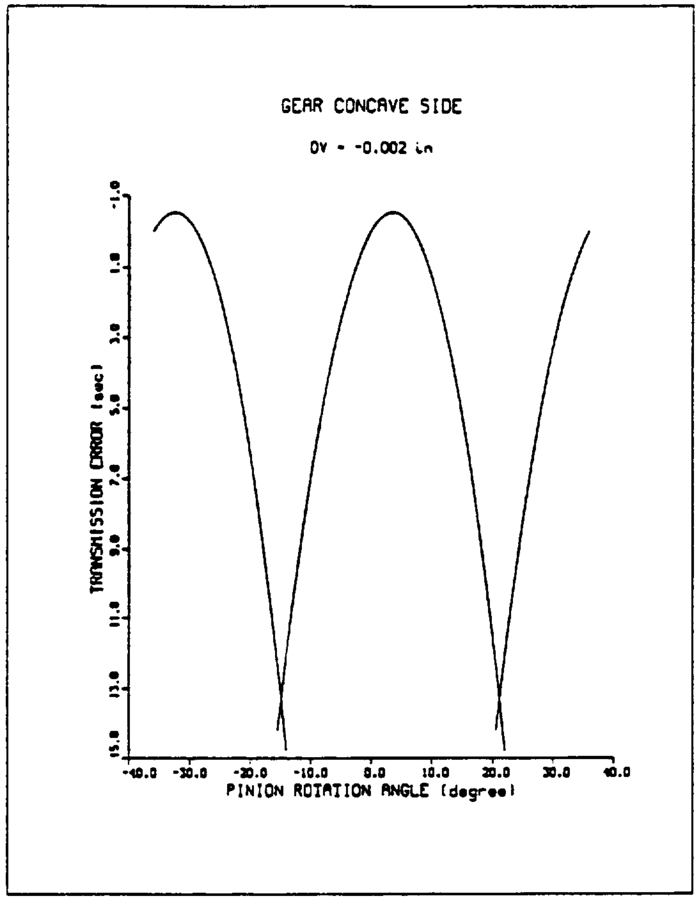


Figure 43: Curved-edged blade, gear concave side, $\triangle V = +0.002$ inches.



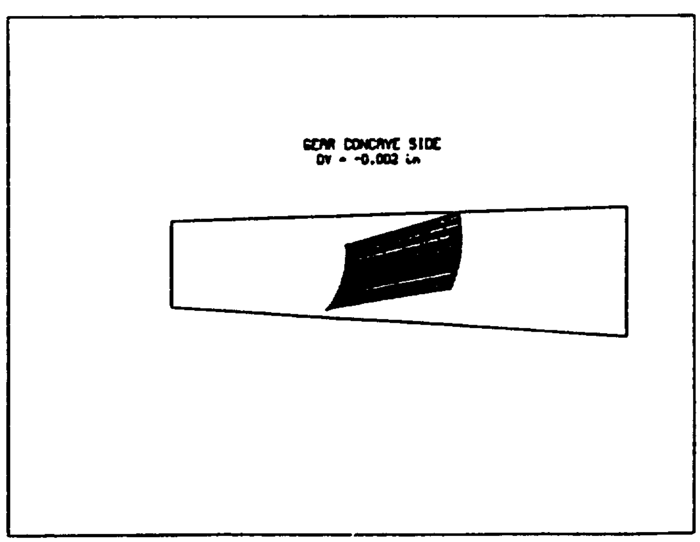


Figure 44: Curved-edged blade, gear concave side, $\triangle V = -0.002$ inches.

APPENDIX C

LISTING OF COMPUTER PROGRAMS

```
************************************************************************
*                                                                      *
*                    Gleason's Spiral Bevel Gears                      *
*                                                                      *
*        Basic Machine-Tool Settings and Tooth Contact Analysis        *
*                                                                      *
*                  Straight Blade to Cut the Pinion                    *
*                                                                      *
************************************************************************
      IMPLICIT REAL*8(A-H,K,M-Z)
      REAL*8 X(1),F(1),FI(1),PAR(6),LM,TX(5),TF(5),TF1(5),TPAR(19),
     &       AZSP(1,1),WORKP(1),AZS(5,5),WORK(5)
      CHARACTER*8 HG,HNGR
      DIMENSION IPVTP(1),IPVT(5)
      EXTERNAL PCN,TCN
      COMMON/P1/PAR
      COMMON/T1/TPAR
      COMMON/A0/HG
      COMMON/A1/p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/A5/CSQG,SNQG,THETAG
      COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSPSIP,SNPSIP
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/B2fx,B2fy,B2fz
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
      COMMON/C1/UG,CSTAUG,SNTAUG
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/D1/UP,CSTAUP,SNTAUP
      COMMON/E1/XBf,YBf,ZBf
      COMMON/F1/PHIGO
      COMMON/G1/DA1,DV1
*
* INPUT THE DESIGN DATA
*
* TN1            : number of pinion teeth
*                  ----- sec. 3.1
```

```
* TN2              : number of gear teeth
*                    ----- sec. 3.1
* RT1dg, RT1min    : root angle of pinion (degree and arc minute, respec-
*                    tively)
*                    ----- sec. 3.1
* RT2dg, RT2min    : root angle of gear (degree and arc minute, respec-
*                    tively)
*                    ----- sec. 3.1
* SHAFdg           : shaft angle (degree)
*                    ----- sec. 3.1
* BETAdg           : mean spiral angle (degree)
*                    ----- sec. 3.1
* ADIA             : average gear cutter diameter
*                    ----- sec. 3.1
* W                : point width of gear cutter
*                    ----- sec. 3.1
* A                : mean cone distance
*                    ----- sec. 3.1
* ALPHdg           : blade angle of gear cutter (degree)
*                    ----- sec. 3.1
* DLTXdg           : angle measured counterclockwise from root of gear to
*                    the tangent of the contact path (degree)
*                    gear convex side
*                    ----- fig. 19
* DLTVdg           : angle measured counterclockwise from root of gear to
*                    the tangent of the contact path (degree)
*                    gear concave side
*                    ----- fig. 19
* M21XPR           : first derivative of gear ratio
*                    gear convex side
*                    ----- sec. 3.1.1
* M21VPR           : first derivative of gear ratio
*                    gear concave side
*                    ----- sec. 3.1.1
* AXILX            : semimajor axis of contact ellipse
*                    gear convex side
*                    ----- eq. (3.76)
* AXILV            : semimajor axis of contact ellipse
*                    gear concave side
*                    ----- eq. (3.76)
* HNGR             : hand of gear ('L' or 'R')
* DA               : amount of shift along pinion axis
*                    + : pinion mounting distance being increased
*                    - : pinion mounting distance being decreased
* DV               : amount of pinion shaft offset
*                    the same sense as yf shown in fig. 18
* DEF              : elastic approach
*                    ----- eq. (3.76)
* EPS              : amount to control calculation accuracy
*
* OUTPUT OF THE BASIC MACHINE-TOOL SETTINGS
*
* PSIGdg           : gear blade angle
```

```
* PSIPdg          : pinion blade angle
* RG              : tip radius of gear cutter
* RP              : tip radius of pinion cutter
* SG              : gear radial
* SP              : pinion radial
* QGdg            : gear cradle angle
* QPdg            : pinion cradle angle
* MG2             : gear cutting ratio
* MP1             : pinion cutting ratio
* EM              : machining offset
* LM              : machine center to back + sliding base
*
      DATA TN1,TN2/10.D00,41.D00/
      DATA RT1dg,RT1min/12.D00,1.D00/
      DATA RT2dg,RT2min/72.D00,25.D00/
      DATA SHAFdg,BETAdg/90.D00,35.D00/
      DATA ADIA/6.0D00/
      DATA W/0.08D00/
      DATA A/3.226D00/
      DATA ALPHdg/20.D00/
      DATA DLTXdg/ 90.D00/
      DATA DLTVdg/ 75.D00/
      DATA M21XPR/-3.5D-03/
      DATA M21VPR/5.2D-03/
      DATA AXILX/0.1710D00/
      DATA AXILV/0.1810D00/
      DATA HNGR/'L'/
      DATA DV,DA/0.D00,0.D00/
      DATA DEF/0.00025D00/
      DATA EPS/1.D-12/
*
*
*
      DA1=DA
      DV1=DV
      HG=HNGR
*
* CONVERT DEGREES TO RADIANS
*
      CNST=4.D00*DATAN(1.D00)/180.D00
      RITAG=90.D00*CNST
      DLTX=DLTXdg*CNST
      DLTV=DLTVdg*CNST
      RT1=(RT1dg+RT1min/60.D00)*CNST
      RT2=(RT2dg+RT2min/60.D00)*CNST
      BETA=BETAdg*CNST
      PSIG=ALPHdg*CNST
      SHAFT=SHAFdg*CNST
      CSRT2=DCOS(RT2)
      SNRT2=DSIN(RT2)
      CSRT1=DCOS(RT1)
      SNRT1=DSIN(RT1)
*
```

```
* CALCULATE PITCH ANGLES
*
      MM21=TN1/TN2
c ----- eq. (3.1)
      PITCH2=DATAN(DSIN(SHAFT)/(MM21+DCOS(SHAFT)))
      IF(PITCH2 .LT. 0.D00)THEN
       PITCH2=PITCH2+180.D00
      END IF
      CSPIT2=DCOS(PITCH2)
      SNPIT2=DSIN(PITCH2)
c ----- eq. (3.2)
      PITCH1=SHAFT-PITCH2
      CSPIT1=DCOS(PITCH1)
      SNPIT1=DSIN(PITCH1)
*
* CALCULATE DEDENDUM ANGLES
*
c ----- eq. (3.3)
      D1=PITCH1-RT1
      D2=PITCH2-RT2
      CSD1=DCOS(D1)
      SND1=DSIN(D1)
      TND1=SND1/CSD1
      CSD2=DCOS(D2)
      SND2=DSIN(D2)
      TND2=SND2/CSD2
*
* CALCULATE GEAR CUTTING RATIO
*
c ----- eq. (3.7)
      MG2=DSIN(PITCH2)/CSD2
*
* FOR GEAR CONVEX SIDE I = 1, FOR GEAR CONCAVE SIDE I = 2.
*
      DO 99999 I=1,2
      IF(I .EQ. 1)THEN
       WRITE(72,*)'GEAR CONVEX SIDE'
       DLTA=DLTX
       M21PRM=M21XPR
       AXIL=AXILX
      ELSE
       WRITE(72,*)'GEAR CONCAVE SIDE'
       DLTA=DLTV
       M21PRM=M21VPR
       AXIL=AXILV
      END IF
      WRITE(72,*)
c ----- eq. (3.76)
      AXIA=DEF/(AXIL*AXIL)
*
* CALCULATE GEAR BLADE ANGLE
*
c ----- sec. 2.2
```

```
      IF(I .EQ. 2)THEN
       PSIG=180.D00*CNST-PSIG
      END IF
      CSPSIG=DCOS(PSIG)
      SNPSIG=DSIN(PSIG)
      CTPSIG=CSPSIG/SNPSIG
*
* CALCULATE CUTTER TIP RADIUS
*
c ----- eq. (3.8)
      IF(I .EQ. 1)THEN
       RG=(ADIA-W)/2.D00
      ELSE
       RG=(ADIA+W)/2.D00
      END IF
*
* CALCULATE RADIAL
*
c ----- eq. (3.9)
      IF(I .EQ. 1)THEN
       SG=DSQRT(ADIA*ADIA/4.D00+A*A*CSD2*CSD2-A*ADIA*CSD2*DSIN(BETA))
*
* CALCULATE CRADLE ANGLE
*
c ----- eq. (3.10)
       QG=DACOS((A*A*CSD2*CSD2+SG*SG-ADIA*ADIA/4.D00)/(2.D00*A*SG*CSD2))
       CSQG=DCOS(QG)
       SNQG=DSIN(QG)
      END IF
*
      PAR(1)=RG*CTPSIG*CSPSIG
      PAR(4)=RG*CTPSIG
*
* CALCULATE PHIG
*
       PHIG=0.D00
       PHIGO=PHIG
       CSPHIG=DCOS(PHIG)
       SNPHIG=DSIN(PHIG)
*
       IF(HG .EQ. 'L')THEN
        IF(I .EQ. 1)THEN
* Mmc=Mms*Msc
c ----- eq. (2.26)
        CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .   1.D00,0.D00,0.D00,0.D00,CSPHIG,SNPHIG,0.D00,-SNPHIG,CSPHIG,
     .   0.D00,0.D00,0.D00,
     .   1.D00,0.D00,0.D00,0.D00,CSQG,-SNQG,0.D00,SNQG,CSQG,
     .   0.D00,-SG*SNQG,SG*CSQG)
        END IF
* Mpc=Mpm*Mmc
c ----- eqs. (2.25), (3.13)
        CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
```

```
     .    CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .    0.D00,0.D00,0.D00,
     .    m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
*
        ELSE
*
         IF(I .EQ. 1)THEN
* Mmc=Mms*Msc
c ----- eq. (2.26)
          CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .     1.D00,0.D00,0.D00,0.D00,CSPHIG,-SNPHIG,0.D00,SNPHIG,CSPHIG,
     .     0.D00,0.D00,0.D00,
     .     1.D00,0.D00,0.D00,0.D00,CSQG,SNQG,0.D00,-SNQG,CSQG,
     .     0.D00,SG*SNQG,SG*CSQG)
         END IF
* Mpc=Mpm*Mmc
c ----- eqs. (2.25), (3.13)
        CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     .   CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .   0.D00,0.D00,0.D00,
     .   m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
       END IF
*
* DETERMINE MAIN CONTACT POINT
*
*
* CALCULATE THETAG
*
c ----- X(1) represents THETAG
       PAR(2)=(MG2-SNRT2)*CSPSIG
       IF(HG .EQ. 'L')THEN
        PAR(3)=-SNQG*CSRT2*SNPSIG
c ----- step 1 in sec. 3.2
        X(1)=QG-BETA+RITAG
       ELSE
        PAR(3)=SNQG*CSRT2*SNPSIG
c ----- step 1 in sec. 3.2
        X(1)=-(QG-BETA+RITAG)
       END IF
       CALL NONLIN(PCN,14,1,100,X,F,FI,1.D-5,AZSP,IPVTP,WORKP)
       THETAG=X(1)
       CSTHEG=DCOS(THETAG)
       SNTHEG=DSIN(THETAG)
*
* CALCULATE TAUG
*
c ----- eq. (2.38)
      IF(HG .EQ. 'L')THEN
       TAUG=THETAG-QG+PHIG
      ELSE
       TAUG=THETAG+QG-PHIG
      END IF
      CSTAUG=DCOS(TAUG)
```

```
      SNTAUG=DSIN(TAUG)
*
* CALCULATE UG
*
c ----- eq. (2.43)
      IF(HG .EQ. 'L')THEN
       UG=RG*CTPSIG*CSPSIG-SG*((MG2-SNRT2)*CSPSIG*SNTHEG-DSIN(QG-PHIG)*
     #    CSRT2*SNPSIG)/(CSRT2*SNTAUG)
      ELSE
       UG=RG*CTPSIG*CSPSIG-SG*((MG2-SNRT2)*CSPSIG*SNTHEG+DSIN(QG-PHIG)*
     #    CSRT2*SNPSIG)/(CSRT2*SNTAUG)
      END IF
*
* CALCULATE MAIN CONTACT POINT
*
c ----- eq. (2.1)
      Bcx=RG*CTPSIG-UG*CSPSIG
      Bcy=UG*SNPSIG*SNTHEG
      Bcz=UG*SNPSIG*CSTHEG
c ----- eq. (2.2)
      Ncx=SNPSIG
      Ncy=CSPSIG*SNTHEG
      Ncz=CSPSIG*CSTHEG
c ----- eq. (2.9)
      EGIcx=0.D00
      EGIcy=CSTHEG
      EGIcz=-SNTHEG
c ----- eq. (3.13)
      CALL TRCOOR(Bpx,Bpy,Bpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . Bcx,Bcy,Bcz)
c ----- eq. (3.16)
      CALL TRCOOR(Npx,Npy,Npz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,0.D00,0.D00,0.D00,
     . Ncx,Ncy,Ncz)
c ----- eq. (3.17)
      CALL TRCOOR(EGIpx,EGIpy,EGIpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,0.D00,0.D00,0.D00,
     . EGIcx,EGIcy,EGIcz)
c ----- fig. 18 & sec. 3.3
      Bfx=Bpx
      Bfy=Bpy
      Bfz=Bpz
      Nfx=Npx
      Nfy=Npy
      Nfz=Npz
      EGIfx=EGIpx
      EGIfy=EGIpy
      EGIfz=EGIpz
*
* CALCULATE PSIP
*
c ----- eq. (3.83)
```

```
         PSIP=DASIN(CSD1*Nfx-SND1*Nfz)
         IF (I .EQ. 1)THEN
          PSIP=-PSIP+180.D00*CNST
         END IF
        CSPSIP=DCOS(PSIP)
        SNPSIP=DSIN(PSIP)
*
* CALCULATE TAUP
*
c ----- eqs. (3.84)-(3.86)
        TAUP=DATAN2(Nfy/CSPSIP,(Nfx-CSD1*SNPSIP)/(-SND1*CSPSIP))
         CSTAUP=DCOS(TAUP)
         SNTAUP=DSIN(TAUP)
*
* CALCULATE PRINCIPAL CURVATURES AND DIREGTIONS OF THE GEAR CUTTER
*
c ----- eq. (2.10)
        KGI=-CTPSIG/UG
c ----- eq. (2.12)
        KGII=0.D00
c ----- the second principal direction is determined by rotating of
c ----- the first principal derection about unit normal by 90 degrees
        CALL ROTATE(EGIIfx,EGIIfy,EGIIfz,EGIfx,EGIfy,EGIfz,RITAG,
       . Nfx,Nfy,Nfz)
*
* CALCULATE W2G
*
c ----- eqs. (3.18)-(3.20)
        IF(HG .EQ. 'L')THEN
         W2fx=-SNPIT2
         WGfx=-MG2*CSD2
         W2fy=0.D00
         WGfy=0.D00
         W2fz=CSPIT2
         WGfz=-MG2*SND2
        ELSE
         W2fx=SNPIT2
         WGfx=MG2*CSD2
         W2fy=0.D00
         WGfy=0.D00
         W2fz=-CSPIT2
         WGfz=MG2*SND2
         END IF
*
        W2Gfx=W2fx-WGfx
        W2Gfy=W2fy-WGfy
        W2Gfz=W2fz-WGfz
*
* CALCULATE VT2, VTG, AND VT2G
*
c ----- eq. (3.22)
        CALL CROSS(VT2fx,VT2fy,VT2fz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
c ----- eq. (3.21)
```

```
      CALL CROSS(VTGfx,VTGfy,VTGfz,WGfx,WGfy,WGfz,Bfx,Bfy,Bfz)
c ----- eq. (3.23)
      VT2Gfx=VT2fx-VTGfx
      VT2Gfy=VT2fy-VTGfy
      VT2Gfz=VT2fz-VTGfz
*
* CALCULATE V(2G)GI AND V(2G)GII
*
c ----- eq. (3.24)
      CALL DOT(VGI,EGIfx,EGIfy,EGIfz,VT2Gfx,VT2Gfy,VT2Gfz)
c ----- eq. (3.25)
      CALL DOT(VGII,EGIIfx,EGIIfy,EGIIfz,VT2Gfx,VT2Gfy,VT2Gfz)
*
* CALCULATE A13,A23,A33
*
c ----- eq. (3.26)
      CALL DET(DETI,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIfx,EGIfy,EGIfz)
      A13=-KGI*VGI-DETI
c ----- eq. (3.27)
      CALL DET(DETII,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIIfx,EGIIfy,EGIIfz)
      A23=-KGII*VGII-DETII
c ----- eq. (3.28)
      CALL DET(DET3,Nfx,Nfy,Nfz,W2Gfx,W2Gfy,W2Gfz,VT2Gfx,VT2Gfy,VT2Gfz)
      CALL CROSS(Cx,Cy,Cz,W2fx,W2fy,W2fz,VTGfx,VTGfy,VTGfz)
      CALL CROSS(Dx,Dy,Dz,WGfx,WGfy,WGfz,VT2fx,VT2fy,VT2fz)
      CALL DOT(DET45,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KGI*VGI*VGI+KGII*VGII*VGII-DET3-DET45
*
* CALCULATE SIGMA
*
c ----- eq. (3.29)
      P=A23*A23-A13*A13+(KGI-KGII)*A33
      SIGDBL=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGDBL
*
* CALCULATE K2I AND K2II
*
c ----- eqs. (3.30)-(3.31)
      T1=P/(A33*DCOS(SIGDBL))
      T2=KGI+KGII-(A13*A13+A23*A23)/A33
      K2I=(T1+T2)/2.D00
      K2II=(T2-T1)/2.D00
*
* CALCULATE E2I AND E2II
*
c ----- description after eq. (3.29)
      CALL ROTATE(E2Ifx,E2Ify,E2Ifz,EGIfx,EGIfy,EGIfz,-SIGMA,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E2IIfx,E2IIfy,E2IIfz,E2Ifx,E2Ify,E2Ifz,RITAG,
     . Nfx,Nfy,Nfz)
c ----- eq. (3.44)
      TNETAG=DSIN(DLTA+SIGMA)/DCOS(DLTA+SIGMA)
*
```

```
* CALCULATE W2
*
c ----- eq. (3.33)
      IF(HG .EQ. 'L')THEN
       W2fx=-MM21*SNPIT2
       W2fy=0.D00
       W2fz=MM21*CSPIT2
      ELSE
       W2fx=MM21*SNPIT2
       W2fy=0.D00
       W2fz=-MM21*CSPIT2
      END IF
*
* CALCULATE W1
*
c ----- eq. (3.32)
      IF(HG .EQ. 'L')THEN
       W1fx=-SNPIT1
       W1fy=0.D00
       W1fz=-CSPIT1
      ELSE
       W1fx=SNPIT1
       W1fy=0.D00
       W1fz=CSPIT1
      END IF
*
* CALCULATE W12
*
c ----- eq. (3.34)
      W12fx=W1fx-W2fx
      W12fy=W1fy-W2fy
      W12fz=W1fz-W2fz
*
* CALCULATE VT2
*
c ----- eq. (3.36)
      CALL CROSS(VT2fx,VT2fy,VT2fz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
*
* CALCULATE VT1
*
c ----- eq. (3.35)
      CALL CROSS(VT1fx,VT1fy,VT1fz,W1fx,W1fy,W1fz,Bfx,Bfy,Bfz)
*
* CALCULATE VT12
*
c ----- eq. (3.37)
      VT12fx=VT1fx-VT2fx
      VT12fy=VT1fy-VT2fy
      VT12fz=VT1fz-VT2fz
*
* CALCULATE V2
*
c ----- eq. (3.38)
```

```
      CALL DOT(V2I,VT12fx,VT12fy,VT12fz,E2Ifx,E2Ify,E2Ifz)
c ----- eq. (3.39)
      CALL DOT(V2II,VT12fx,VT12fy,VT12fz,E2IIfx,E2IIfy,E2IIfz)
*
* CALCULATE A31
*
c ----- eq. (3.40)
      CALL DET(DET1,W12fx,W12fy,W12fz,Nfx,Nfy,Nfz,E2Ifx,E2Ify,E2Ifz)
      A31=-K2I*V2I-DET1
c ----- eq. (A.33)
      A13=A31
*
* CALCULATE A32
*
c ----- eq. (3.41)
      CALL DET(DET2,W12fx,W12fy,W12fz,Nfx,Nfy,Nfz,E2IIfx,E2IIfy,E2IIfz)
      A32=-K2II*V2II-DET2
c ----- eq. (A.35)
      A23=A32
*
* CALCULATE A33
*
c ----- eq. (3.42)
      CALL DET(DET3,Nfx,Nfy,Nfz,W12fx,W12fy,W12fz,VT12fx,VT12fy,VT12fz)
      CALL CROSS(Cx,Cy,Cz,W1fx,W1fy,W1fz,VT2fx,VT2fy,VT2fz)
      CALL CROSS(Dx,Dy,Dz,W2fx,W2fy,W2fz,VT1fx,VT1fy,VT1fz)
      CALL DOT(DOT1,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      CALL DET(DET4,Nfx,Nfy,Nfz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
      A33=K2I*V2I*V2I+K2II*V2II*V2II-DET3-DOT1+M21PRM*DET4
*
* CALCULATE ETAP
*
c ----- eq. (3.53)
      ETAP=DATAN(((A33+A31*V2I)*TNETAG-A31*V2II)/(A33+A32*
     . (V2II-V2I*TNETAG)))
      TNETAP=DSIN(ETAP)/DCOS(ETAP)
*
* CALCULATE A11, A12, AND A22
*
      N3=(1.D00+TNETAP*TNETAP)*A33
c ----- eq. (3.72)
      N1=(A13*A13-(A23*TNETAP)**2)/N3
c ----- eq. (3.73)
      N2=(A23+A13*TNETAP)*(A13+A23*TNETAP)/N3
      KS2=K2I+K2II
      G2=K2I-K2II
c ----- eqs. (3.74), (3.75)
      KS1=KS2-((4.D00*AXIA*AXIA-N1*N1-N2*N2)*(1.D00+TNETAP*TNETAP)/
     . (-2.D00*AXIA*(1.D00+TNETAP*TNETAP)+N1*(TNETAP*TNETAP-1.D00)
     . -2.D00*N2*TNETAP))
c ----- eqs. (3.66), (3.69) & description after eq. (3.60)
      A11=TNETAP*TNETAP/(1.D00+TNETAP*TNETAP)*(KS2-KS1)+N1
c ----- eqs. (3.67), (3.70) & description after eq. (3.60)
```

```
      A12=-TNETAP/(1.D00+TNETAP*TNETAP)*(KS2-KS1)+N2
c ----- eqs. (3.68), (3.71) & description after eq. (3.60)
      A22=1.D00/(1.D00+TNETAP*TNETAP)*(KS2-KS1)-N1
c ----- eq. (A.32)
      A21=A12
*
* CALCULATE SIGMA(12)
*
c ----- eq. (3.77)
      SIGDBL=DATAN(2.D00*A12/(K2I-K2II-A11+A22))
      SIGM12=.5D00*SIGDBL
*
* CALCULATE K1I AND K1II
*
c ----- eq. (3.78)
      G1=2.D00*A12/DSIN(SIGDBL)
c ----- eq. (3.79)
      K1I=.5D00*(KS1+G1)
      K1II=.5D00*(KS1-G1)
*
* CALCULATE E1I AND E1II
*
c ----- similar to description after eq. (3.29)
      CALL ROTATE(E1Ifx,E1Ify,E1Ifz,E2Ifx,E2Ify,E2Ifz,-SIGM12,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E1IIfx,E1IIfy,E1IIfz,E1Ifx,E1Ify,E1Ifz,RITAG,
     . Nfx,Nfy,Nfz)
*
* PINION
*
*
* CALCULATE PRINCIPAL DIRECTIONS OF THE PINION CUTTER
*
c ----- eq. (3.92)
      IF(HG .EQ. 'L')THEN
       EPIfx=SND1*SNTAUP
       EPIfy=CSTAUP
       EPIfz=CSD1*SNTAUP
      ELSE
       EPIfx=-SND1*SNTAUP
       EPIfy=-CSTAUP
       EPIfz=-CSD1*SNTAUP
      END IF
      IF(DACOS(EGIfx*EPIfx+EGIfy*EPIfy+EGIfz*EPIfz)/CNST .GT. 45.D00)
     .THEN
       EPIfx=-EPIfx
       EPIfy=-EPIfy
       EPIfz=-EPIfz
      END IF
      CALL ROTATE(EPIIfx,EPIIfy,EPIIfz,EPIfx,EPIfy,EPIfz,RITAG,
     . Nfx,Nfy,Nfz)
*
* CALCULATE THE ANGLE BETWEEN PRINCIPAL DIRECTIONS OF PINION AND CUTTER
```

```
*
c ----- cross product of eli and epi
      SNSIGM=(E1Ify*EPIfz-E1Ifz*EPIfy)/Nfx
c ----- dot product of eli and epi
      CSSIGM=E1Ifx*EPIfx+E1Ify*EPIfy+E1Ifz*EPIfz
      CS2SIG=2.D00*CSSIGM*CSSIGM-1.D00
      TN2SIG=2.D00*SNSIGM*CSSIGM/CS2SIG
*
* CALCULATE PRINCIPAL CURVATURES OF PINION CUTTER
*
c ----- eq. (2.12)
      KPII=0.D00
c ----- eq. (3.94)
      KPI=K1I*K1II/(K1I*SNSIGM*SNSIGM+K1II*CSSIGM*CSSIGM)
*
* CALCULATE A11, A12, AND A22
*
c ----- eq. (A.31)
      A11=KPI-K1I*CSSIGM*CSSIGM-K1II*SNSIGM*SNSIGM
c ----- eq. (A.32)
      A12=(K1I-K1II)*SNSIGM*CSSIGM
c ----- eq. (A.34)
      A22=KPII-K1I*SNSIGM*SNSIGM-K1II*CSSIGM*CSSIGM
*
* CALCULATE UP
*
c ----- eq. (3.95)
      UP=1.D00/(KPI*SNPSIP/CSPSIP)
*
* CALCULATE RP
*
c ----- eq. (3.99)
      Bmx=-Bfx*CSD1+Bfz*SND1
c ----- eq. (3.100)
      RP=(Bmx+UP*CSPSIP)*SNPSIP/CSPSIP
*
* CALCULATE MP1
*
c ----- eq. (3.107)
      C11=(Nfy*EPIfz-Nfz*EPIfy)*CSD1+(Nfy*EPIfx-Nfx*EPIfy)*SND1
      C12=(Nfz*EPIfy-Nfy*EPIfz)*SNPIT1+(Nfy*EPIfx-Nfx*EPIfy)*CSPIT1
c ----- eq. (3.108)
      C22=-(Nfy*EPIIfz-Nfz*EPIIfy)*SNPIT1+(Nfy*EPIIfx-Nfx*EPIIfy)*CSPIT1
      IF(HG .EQ. 'R')THEN
       C11=-C11
       C12=-C12
       C22=-C22
      END IF
c ----- eq. (3.119)
      T4=(Bfy*CSRT1)/(EPIIfx*CSD1-EPIIfz*SND1)
      IF(HG .EQ. 'R')THEN
       T4=-T4
      END IF
```

```
c ----- eq. (3.120)
      T1=-C11/KPI
      T2=(A11*KPII*T4+A11*C22-A12*C12)/(A12*KPI)
c ----- eq. (3.122)
      U11=T1*EPIfx
      U12=T2*EPIfx+T4*EPIIfx
      U21=T1*EPIfy
      U22=T2*EPIfy+T4*EPIIfy
      U31=T1*EPIfz
      U32=T2*EPIfz+T4*EPIIfz
c ----- eq. (3.124)
      V1=U21*Nfz*CSD1+U21*Nfx*SND1-Nfy*(U11*SND1+U31*CSD1)
c ----- eq. (3.125)
      V2=(U22*CSD1-U21*SNPIT1)*Nfz-(U11*CSPIT1+U12*SND1+U32*CSD1-U31
     *SNPIT1)*Nfy+(U21*CSPIT1+U22*SND1)*Nfx
c ----- eq. (3.126)
      V3=U22*CSPIT1*Nfx+(U32*SNPIT1-U12*CSPIT1)*Nfy-U22*SNPIT1*Nfz
      IF(HG .EQ. 'R')THEN
       V1=-V1
       V2=-V2
       V3=-V3
      END IF
c ----- eq. (3.132)
      H11=-U21*CSPIT1+SND1*(Bfz*SNPIT1-Bfx*CSPIT1)
c ----- eq. (3.134)
      H21=U11*CSPIT1-U31*SNPIT1+Bfy*SNRT1
c ----- eq. (3.136)
      H31=U21*SNPIT1+CSD1*(Bfz*SNPIT1-Bfx*CSPIT1)
c ----- eq. (3.133)
      H12=(Bfz*SNPIT1-Bfx*CSPIT1-U22)*CSPIT1
c ----- eq. (3.135)
      H22=-(Bfy-U12*CSPIT1+U32*SNPIT1)
c ----- eq. (3.137)
      H32=-(Bfz*SNPIT1-Bfx*CSPIT1-U22)*SNPIT1
      IF(HG .EQ. 'R')THEN
       H11=U21*CSPIT1+SND1*(Bfz*SNPIT1-Bfx*CSPIT1)
       H21=-U11*CSPIT1+U31*SNPIT1+Bfy*SNRT1
       H31=-U21*SNPIT1+CSD1*(Bfz*SNPIT1-Bfx*CSPIT1)
       H12=(Bfz*SNPIT1-Bfx*CSPIT1+U22)*CSPIT1
       H22=-(Bfy+U12*CSPIT1-U32*SNPIT1)
       H32=-(Bfz*SNPIT1-Bfx*CSPIT1+U22)*SNPIT1
      END IF
c ----- eq. (3.139)
      F1=Nfx*H11+Nfy*H21+Nfz*H31
c ----- eq. (3.140)
      F2=Nfx*H12+Nfy*H22+Nfz*H32
c ----- eq. (3.145)
      Y2=A12*(2.D00*KPI*T1*T2-V2-F1)
      Y3=A12*(KPI*T2*T2+KPII*T4*T4-V3-F2)-(KPI*T2+C12)*(KPII*T4+C22)
      MP1=-Y3/Y2
*
* CALCULATE EM AND LM
*
```

```
c ----- eq. (3.122)
      VT1Pfx=U11*MP1+U12
      VT1Pfy=U21*MP1+U22
      VT1Pfz=U31*MP1+U32
c ----- eq. (3.111)
      IF(HG .EQ. 'L')THEN
       EM=(Bfy*CSPIT1-VT1Pfx)/(MP1*SND1)+Bfy
       LM=(Bfx*CSPIT1-Bfz*SNPIT1+VT1Pfy)/MP1+Bfx*SND1+Bfz*CSD1
      ELSE
       EM=(-Bfy*CSPIT1-VT1Pfx)/(MP1*SND1)-Bfy
       LM=(Bfx*CSPIT1-Bfz*SNPIT1-VT1Pfy)/MP1+Bfx*SND1+Bfz*CSD1
      END IF
*
* CALCULATE SP AND QP
*
c ----- eqs. (3.150), (3.151)
      IF(HG .EQ. 'L')THEN
       Z1=-Bfy+EM-UP*SNPSIP*SNTAUP
      ELSE
       Z1=Bfy+EM+UP*SNPSIP*SNTAUP
      END IF
      Z2=Bfx*SND1+Bfz*CSD1-LM-UP*SNPSIP*CSTAUP
      SP=DSQRT(Z1*Z1+Z2*Z2)
      QP=DATAN(Z1/Z2)
      IF(HG .EQ. 'L')THEN
       THETAP=TAUP-QP
      ELSE
       THETAP=TAUP+QP
      END IF
*
* CONVERT RADIAN TO DEGREE
*
      PSIGDG=PSIG/CNST
      PSIPDG=PSIP/CNST
      TAUGDG=TAUG/CNST
      TAUPDG=TAUP/CNST
      QGDG=QG/CNST
      QPDG=QP/CNST
      THEGDG=THETAG/CNST
      THEPDG=THETAP/CNST
      PHIGDG=PHIGO/CNST
*
* OUTPUT
*
      WRITE(72,10000)PSIGDG,PSIPDG,RG,RP,TAUGDG,TAUPDG,SG,SP,QGDG,QPDG,
     . MG2,MP1,EM,LM,UG,UP,THEGDG,THEPDG,PHIGDG
10000 FORMAT(1X,'PSIGDG   =',G20.12,12X,'PSIPDG   =',G20.12,/
     .       ,1X,'RG       =',G20.12,12X,'RP       =',G20.12,/
     .       ,1X,'TAUGDG   =',G20.12,12X,'TAUPDG   =',G20.12,/
     .       ,1X,'SG       =',G20.12,12X,'SP       =',G20.12,/
     .       ,1X,'QGDG     =',G20.12,12X,'QPDG     =',G20.12,/
     .       ,1X,'MG2      =',G20.12,12X,'MP1      =',G20.12,/
     .       ,1X,'EM       =',G20.12,12X,'LM       =',G20.12,/
```

```
     .      ,1X,'UG       =',G20.12,12X,'UP       =',G20.12,/
     .      ,1X,'THETAGDG =',G20.12,12X,'THETAPDG =',G20.12,/
     .      ,1X,'PHIGODG  =',G20.12,12X,/)
*
* TCA
*
      IF(I .EQ. 1)THEN
      TPAR(1)=RG*CSPSIG/SNPSIG*CSPSIG
      TPAR(2)=(MG2-SNRT2)*CSPSIG
      TPAR(3)=CSRT2*SNPSIG
      TPAR(4)=RG*CSPSIG/SNPSIG
      TPAR(5)=CSD2*SNPSIG
      TPAR(6)=SND2*CSPSIG
      TPAR(7)=SND2*SNPSIG
      TPAR(8)=CSD2*CSPSIG
      TPAR(9)=RP*CSPSIP/SNPSIP*CSPSIP
      TPAR(10)=(MP1-SNRT1)*CSPSIP
      TPAR(11)=CSRT1*SNPSIP
      TPAR(12)=SNRT1*CSPSIP
      TPAR(13)=RP*CSPSIP/SNPSIP
      TPAR(14)=CSD1*SNPSIP
      TPAR(15)=SND1*CSPSIP
      TPAR(16)=SND1*SNPSIP
      TPAR(17)=CSD1*CSPSIP
      TPAR(18)=LM*SND1
      TPAR(19)=LM*CSD1
*
      PHIP=0.D00
      PHI21=0.D00
      PHI11=0.D00
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
*
      TX(1)=PHIP
      TX(2)=THETAP
      TX(3)=PHI21
      TX(4)=PHIGO
      TX(5)=THETAG
      CALL NONLIN(TCN,14,5,100,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
      PHIPO=TX(1)
      THEPO=TX(2)
      PHI210=TX(3)
      PHIGO=TX(4)
      THEGO=TX(5)
*
      TX(1)=PHIPO
      TX(2)=THEPO
      TX(3)=PHI210
      TX(4)=PHIGO
      TX(5)=THEGO
      D1HI11=18.D00/36.D00*CNST
*
      DO 100 IJ=1,60
```

```
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
      CALL NONLIN(TCN,14,5,100,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
      PHIP=TX(1)
      THETAP=TX(2)
      PHI21=TX(3)
      PHIG=TX(4)
      THETAG=TX(5)
      ERROR=((PHI21*36.D02-PHI210*36.D02)-PHI11*36.D02*TN1/TN2)/CNST
*
      CALL PRING2(KS2,G2,E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz)
      CALL PRINP1(KS1,G1,E1Ifx,E1Ify,E1Ifz,E1IIfx,E1IIfy,E1IIfz)
      CALL SIGAN2(E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz,E1Ifx,E1Ify,
     &            E1Ifz,CS2SIG,SN2SIG,SIGM12)
      CALL EULER(KS2,G2,KS1,G1,CS2SIG,SN2SIG,IEU)
      IF(IEU .EQ. 1)THEN
       WRITE(72,*)'THERE IS INTERFERENCE'
       GO TO 88888
      END IF
*
      CALL ELLIPS(KS2,G2,KS1,G1,CS2SIG,SN2SIG,DEF,ALFA1,
     &            AXISL,AXISS,E1Ifx,E1Ify,E1Ifz)
*
      CALL PF(B2px,B2py,B2pz,B2fx,B2fy,B2fz)
*
* XBf, YBf, and ZBf is the direction of the long axis of the ellipse
*
      CALL PF(XBp,YBp,ZBp,XBf,YBf,ZBf)
      ELB1px=B2px+XBp
      ELB1pz=B2pz+ZBp
      ELB2px=B2px-XBp
      ELB2pz=B2pz-ZBp
*
      IF(I .EQ. 1)THEN
       WRITE(79,9000)IJ,PHI11/CNST,IJ,ERROR
       WRITE(78,8000)IJ,B2pz,IJ,B2px
       WRITE(77,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
      ELSE
       WRITE(89,9000)IJ,PHI11/CNST,IJ,ERROR
       WRITE(88,8000)IJ,B2pz,IJ,B2px
       WRITE(87,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
      END IF
*
      PHI11=PHI11+D1HI11
*
100   CONTINUE
*
*
*
      PHI11=0.D00
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
*
```

```
      TX(1)=PHIPO
      TX(2)=THEPO
      TX(3)=PHI210
      TX(4)=PHIGO
      TX(5)=THEGO
      D1HI11=18.D00/36.D00*CNST
*
      DO 200 IJ=1,60
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
      CALL NONLIN(TCN,14,5,100,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
      PHIP=TX(1)
      THETAP=TX(2)
      PHI21=TX(3)
      PHIG=TX(4)
      THETAG=TX(5)
      ERROR=((PHI21*36.D02-PHI210*36.D02)-PHI11*36.D02*TN1/TN2)/CNST
*
      CALL PRING2(KS2,G2,E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz)
      CALL PRINP1(KS1,G1,E1Ifx,E1Ify,E1Ifz,E1IIfx,E1IIfy,E1IIfz)
      CALL SIGAN2(E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz,E1Ifx,E1Ify,
     .            E1Ifz,CS2SIG,SN2SIG,SIGM12)
*
      CALL EULER(KS2,G2,KS1,G1,CS2SIG,SN2SIG,IEU)
      IF(IEU .EQ. 1)THEN
       WRITE(72,*)'THERE IS INTERFERENCE'
       GO TO 88888
      END IF
*
      CALL ELLIPS(KS2,G2,KS1,G1,CS2SIG,SN2SIG,DEF,ALFA1,
     .            AXISL,AXISS,E1Ifx,E1Ify,E1Ifz)
*
      CALL PF(B2px,B2py,B2pz,B2fx,B2fy,B2fz)
*
* XBf, YBf, and ZBf is the direction of the long axis of the ellipse
*
      CALL PF(XBp,YBp,ZBp,XBf,YBf,ZBf)
      ELB1px=B2px+XBp
      ELB1pz=B2pz+ZBp
      ELB2px=B2px-XBp
      ELB2pz=B2pz-ZBp
*
      IF(I .EQ. 1)THEN
       WRITE(79,9001)IJ,PHI11/CNST,IJ,ERROR
       WRITE(78,8001)IJ,B2pz,IJ,B2px
       WRITE(77,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
      ELSE
       WRITE(89,9001)IJ,PHI11/CNST,IJ,ERROR
       WRITE(88,8001)IJ,B2pz,IJ,B2px
       WRITE(87,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
      END IF
*
      PHI11=PHI11-D1HI11
```

```
*
200   CONTINUE
*
      END IF
*
99999 CONTINUE
88888 CONTINUE

7000  FORMAT(6X,'EX(1)=',F9.6,/,6X,'EY(1)=',F9.6,/,
     .        6X,'EX(2)=',F9.6,/,6X,'EY(2)=',F9.6,/,
     .        6X,'CALL CURVE(EX,EY,2,0)')
8000  FORMAT(6X,'X0(',I2,')=',F9.6,/,6X,'Y0(',I2,')=',F9.6)
8001  FORMAT(6X,'X1(',I2,')=',F9.6,/,6X,'Y1(',I2,')=',F9.6)
9000  FORMAT(6X,'X0(',I2,')=',F7.3,/,6X,'Y0(',I2,')=',F8.3)
9001  FORMAT(6X,'X1(',I2,')=',F7.3,/,6X,'Y1(',I2,')=',F8.3)
      END
*
* FOR THE DETERMINATION OF MEAN CONTACT POINT
*
      SUBROUTINE PCN(X,F,NE)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CHARACTER*8 HG
      INTEGER NE
      REAL*8 X(NE),F(NE),PAR(6)
      COMMON/P1/PAR
      COMMON/A0/HG
      COMMON/A1/p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      THETAG=X(1)
      CSTHEG=DCOS(THETAG)
      SNTHEG=DSIN(THETAG)
      IF(HG .EQ. 'L')THEN
       UG=PAR(1)-SG*(PAR(2)*SNTHEG+PAR(3))/(CSRT2*DSIN(THETAG-QG))
      ELSE
       UG=PAR(1)-SG*(PAR(2)*SNTHEG+PAR(3))/(CSRT2*DSIN(THETAG+QG))
      END IF
      Bcx=PAR(4)-UG*CSPSIG
      Bcy=UG*SNPSIG*SNTHEG
      Bcz=UG*SNPSIG*CSTHEG
      CALL TRCOOR(Bpx,Bpy,Bpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . Bcx,Bcy,Bcz)
      XM=Bpz*(TND1-TND2)/2.D00
      F(1)=Bpx-XM
      END
*
* FOR THE DETERMINATION OF COORDINATES AND NORMALS OF CONTACT POINTS
*
      SUBROUTINE TCN(TX,TF,NE)
      IMPLICIT REAL*8(A-H,K,M-Z)
      INTEGER NE
      CHARACTER*8 HG
```

```
REAL*8 TX(NE),TF(NE),TPAR(19),LM
COMMON/A0/HG
COMMON/T1/TPAR
COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSPSIP,SNPSIP
COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
COMMON/B3/B2fx,B2fy,B2fz
COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
COMMON/C1/UG,CSTAUG,SNTAUG
COMMON/C2/N2fx,N2fy,N2fz
COMMON/D1/UP,CSTAUP,SNTAUP
COMMON/F1/PHIGO
COMMON/G1/DA,DV
PHIP=TX(1)
THETAP=TX(2)
PHI21=TX(3)
PHIG=TX(4)
THETAG=TX(5)
CSPHIP=DCOS(PHIP)
SNPHIP=DSIN(PHIP)
CSTHEP=DCOS(THETAP)
SNTHEP=DSIN(THETAP)
CSPH21=DCOS(PHI21)
SNPH21=DSIN(PHI21)
CSPHIG=DCOS(PHIG)
SNPHIG=DSIN(PHIG)
CSTHEG=DCOS(THETAG)
SNTHEG=DSIN(THETAG)
PHI2=(PHIG-PHIGO)/MG2
PHI1=PHIP/MP1
CSPH2=DCOS(PHI2)
SNPH2=DSIN(PHI2)
CSPH1=DCOS(PHI1)
SNPH1=DSIN(PHI1)
IF(HG .EQ. 'L')THEN
 TAUP=THETAP+QP-PHIP
ELSE
 TAUP=THETAP-QP+PHIP
END IF
CSTAUP=DCOS(TAUP)
SNTAUP=DSIN(TAUP)
IF(HG .EQ. 'L')THEN
 TAUG=THETAG-QG+PHIG
ELSE
 TAUG=THETAG+QG-PHIG
END IF
CSTAUG=DCOS(TAUG)
SNTAUG=DSIN(TAUG)
CSQPHP=DCOS(QP-PHIP)
SNQPHP=DSIN(QP-PHIP)
CSQPHG=DCOS(QG-PHIG)
SNQPHG=DSIN(QG-PHIG)
```

```
*
* GEAR
*
* SURFACE EQUATIONS
*
      IF(HG .EQ. 'L')THEN
       UG=TPAR(1)-SG*(TPAR(2)*SNTHEG-SNQPHG*TPAR(3))/(CSRT2*SNTAUG)
       B2py=UG*SNPSIG*SNTAUG-SG*SNQPHG
      ELSE
       UG=TPAR(1)-SG*(TPAR(2)*SNTHEG+SNQPHG*TPAR(3))/(CSRT2*SNTAUG)
       B2py=UG*SNPSIG*SNTAUG+SG*SNQPHG
      END IF
      B2px=CSD2*(TPAR(4)-UG*CSPSIG)-SND2*(UG*SNPSIG*CSTAUG+SG*CSQPHG)
      B2pz=SND2*(TPAR(4)-UG*CSPSIG)+CSD2*(UG*SNPSIG*CSTAUG+SG*CSQPHG)
      N2px=TPAR(5)-TPAR(6)*CSTAUG
      N2py=CSPSIG*SNTAUG
      N2pz=TPAR(7)+TPAR(8)*CSTAUG
*
* [Mwp]=[Mwa][Map]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .            wp1,wp2,wp3,
     .  CSPH2,SNPH2,0.D00,-SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .            wp1,wp2,wp3,
     .  CSPH2,-SNPH2,0.D00,SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B2wx,B2wy,B2wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,wp1,wp2,wp3,
     . B2px,B2py,B2pz)
      CALL TRCOOR(N2wx,N2wy,N2wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,0.D00,0.D00,0.D00,
     . N2px,N2py,N2pz)
*
**********************************************************************
*
* [Mfw]=[Mfa][Maw]
*
      fa11=CSPIT2
      fa12=0.D00
      fa13=-SNPIT2
      fa21=0.D00
      fa22=1.D00
      fa23=0.D00
      fa31=SNPIT2
```

```
      fa32=0.D00
      fa33=CSPIT2
      fa1=0.d00
      fa2=0.d00
      fa3=0.d00
      IF(HG .EQ. 'L')THEN
       CALL COMBI(fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,
     .   fw1,fw2,fw3,
     .   CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     .   0.D00,0.D00,0.D00,
     .   CSPH21,-SNPH21,0.D00,SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     .   0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,
     .   fw1,fw2,fw3,
     .   CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     .   0.D00,0.D00,0.D00,
     .   CSPH21,SNPH21,0.D00,-SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     .   0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B2fx,B2fy,B2fz,
     . fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,fw1,fw2,fw3,
     . B2wx,B2wy,B2wz)
      CALL TRCOOR(N2fx,N2fy,N2fz,
     . fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,0.D00,0.D00,0.D00,
     . N2wx,N2wy,N2wz)
*
* PINION
*
* SURFACE EQUATIONS
*
      IF(HG .EQ. 'L')THEN
       UP=TPAR(9)-(SP*(TPAR(10)*SNTHEP+SNQPHP*TPAR(11))-EM*(TPAR(11)+
     .     TPAR(12)*CSTAUP)-LM*TPAR(12)*SNTAUP)/(CSRT1*SNTAUP)
       B1py=UP*SNPSIP*SNTAUP+SP*SNQPHP-EM
      ELSE
       UP=TPAR(9)-(SP*(TPAR(10)*SNTHEP-SNQPHP*TPAR(11))+EM*(TPAR(11)+
     .     TPAR(12)*CSTAUP)-LM*TPAR(12)*SNTAUP)/(CSRT1*SNTAUP)
       B1py=UP*SNPSIP*SNTAUP-SP*SNQPHP+EM
      END IF
      B1px=CSD1*(TPAR(13)-UP*CSPSIP)-SND1*(UP*SNPSIP*CSTAUP+SP*
     .      CSQPHP)-LM*SND1
      B1pz=SND1*(TPAR(13)-UP*CSPSIP)+CSD1*(UP*SNPSIP*CSTAUP+SP*
     .      CSQPHP)+LM*CSD1
      N1px=-(TPAR(14)-TPAR(15)*CSTAUP)
      N1py=-CSPSIP*SNTAUP
      N1pz=-(TPAR(16)+TPAR(17)*CSTAUP)
*
* [Mwp]=[Mwa] [Map]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .            wp1,wp2,wp3,
```

```
     .  CSPH1,-SNPH1,0.D00,SNPH1,CSPH1,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT1,0.D00,SNPIT1,0.D00,1.D00,0.D00,-SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .             wp1,wp2,wp3,
     .  CSPH1,SNPH1,0.D00,-SNPH1,CSPH1,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT1,0.D00,SNPIT1,0.D00,1.D00,0.D00,-SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B1wx,B1wy,B1wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,wp1,wp2,wp3,
     . B1px,B1py,B1pz)
      CALL TRCOOR(N1wx,N1wy,N1wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,0.D00,0.D00,0.D00,
     . N1px,N1py,N1pz)
*
*******************************************************************
*
*  [Mpw]=[Mpa] [Maw]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,
     .  pw1,pw2,pw3,
     .  CSPIT1,0.D00,-SNPIT1,0.D00,1.D00,0.D00,SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00,
     .  CSPH11,SNPH11,0.D00,-SNPH11,CSPH11,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,
     .  pw1,pw2,pw3,
     .  CSPIT1,0.D00,-SNPIT1,0.D00,1.D00,0.D00,SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00,
     .  CSPH11,-SNPH11,0.D00,SNPH11,CSPH11,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B1px,B1py,B1pz,
     . pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,pw1,pw2,pw3,
     . B1wx,B1wy,B1wz)
      CALL TRCOOR(N1px,N1py,N1pz,
     . pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,0.D00,0.D00,0.D00,
     . N1wx,N1wy,N1wz)
      B1fx=-B1px+DA*SNPIT1
      B1fy=-B1py+DV
      B1fz=B1pz+DA*CSPIT1
      N1fx=-N1px
      N1fy=-N1py
      N1fz=N1pz
      TF(1)=B2fx-B1fx
      TF(2)=B2fy-B1fy
      TF(3)=B2fz-B1fz
```

```
      TF(4)=N2fx-N1fx
      TF(5)=N2fy-N1fy
      END
*
* FOR THE DETERMINATION OF GEAR PRINCIPAL CURVATURES AND DIRECTIONS
*
      SUBROUTINE PRING2(KS2,G2,E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CHARACTER*8 HG
      COMMON/A0/HG
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/Bfx,Bfy,Bfz
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
      COMMON/C1/UG,CSTAUG,SNTAUG
      COMMON/C2/Nfx,Nfy,Nfz
      KGI=-CSPSIG/(UG*SNPSIG)
      KGII=0.D00
      EGIfx=SND2*SNTAUG
      EGIfy=CSTAUG
      EGIfz=-CSD2*SNTAUG
      CALL ROTATE(EGIIfx,EGIIfy,EGIIfz,EGIfx,EGIfy,EGIfz,RITAG,
     . Nfx,Nfy,Nfz)

*
* CALCULATE W2G
*
      IF(HG .EQ. 'L')THEN
       W2fx=-SNPIT2
       WGfx=-MG2*CSD2
       W2fy=0.D00
       WGfy=0.D00
       W2fz=CSPIT2
       WGfz=-MG2*SND2
      ELSE
       W2fx=SNPIT2
       WGfx=MG2*CSD2
       W2fy=0.D00
       WGfy=0.D00
       W2fz=-CSPIT2
       WGfz=MG2*SND2
      END IF

      W2Gfx=W2fx-WGfx
      W2Gfy=W2fy-WGfy
      W2Gfz=W2fz-WGfz
*
* CALCULATE VT2, VTG, AND VT2G
*

      CALL CROSS(VT2fx,VT2fy,VT2fz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
```

```
      CALL CROSS(VTGfx,VTGfy,VTGfz,WGfx,WGfy,WGfz,Bfx,Bfy,Bfz)
      VT2Gfx=VT2fx-VTGfx
      VT2Gfy=VT2fy-VTGfy
      VT2Gfz=VT2fz-VTGfz
*
* CALCULATE V(2G)GI AND V(2G)GII
*
      CALL DOT(VGI,EGIfx,EGIfy,EGIfz,VT2Gfx,VT2Gfy,VT2Gfz)
      CALL DOT(VGII,EGIIfx,EGIIfy,EGIIfz,VT2Gfx,VT2Gfy,VT2Gfz)
*
* CALCULATE A13,A23,A33
*
      CALL DET(DETI,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIfx,EGIfy,EGIfz)
      A13=-KGI*VGI-DETI
      CALL DET(DETII,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIIfx,EGIIfy,EGIIfz)
      A23=-KGII*VGII-DETII
      CALL DET(DET3,Nfx,Nfy,Nfz,W2Gfx,W2Gfy,W2Gfz,VT2Gfx,VT2Gfy,VT2Gfz)
      CALL CROSS(Cx,Cy,Cz,W2fx,W2fy,W2fz,VTGfx,VTGfy,VTGfz)
      CALL CROSS(Dx,Dy,Dz,WGfx,WGfy,WGfz,VT2fx,VT2fy,VT2fz)
      CALL DOT(DET45,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KGI*VGI*VGI+KGII*VGII*VGII-DET3-DET45
*
* CALCULATE SIGMA
*
      P=A23*A23-A13*A13+(KGI-KGII)*A33
      SIGDBL=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGDBL
*
* CALCULATE K2I AND K2II
*
      T1=P/(A33*DCOS(SIGDBL))
      T2=KGI+KGII-(A13*A13+A23*A23)/A33
      K2I=(T1+T2)/2.D00
      K2II=(T2-T1)/2.D00
*
* CALCULATE E2I AND E2II
*
      CALL ROTATE(E2Ifx,E2Ify,E2Ifz,EGIfx,EGIfy,EGIfz,-SIGMA,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E2IIfx,E2IIfy,E2IIfz,E2Ifx,E2Ify,E2Ifz,RITAG,
     . Nfx,Nfy,Nfz)
      END
*
* FOR THE DETERMINATION OF PINION PRINCIPAL CURVATURES AND DIRECTIONS
*
      SUBROUTINE PRINP1(KS1,G1,E1Ifx,E1Ify,E1Ifz,E1IIfx,E1IIfy,E1IIfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      REAL*8 TPAR(19),LM
      CHARACTER*8 HG
      COMMON/T1/TPAR
      COMMON/A0/HG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSPSIP,SNPSIP
```

```
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/Bfx,Bfy,Bfz
      COMMON/C2/Nfx,Nfy,Nfz
      COMMON/D1/UP,CSTAUP,SNTAUP
      KPI=CSPSIP/(UP*SNPSIP)
      KPII=0.D00
      EPIfx=SND1*SNTAUP
      EPIfy=CSTAUP
      EPIfz=CSD1*SNTAUP
      CALL ROTATE(EPIIfx,EPIIfy,EPIIfz,EPIfx,EPIfy,EPIfz,RITAG,
     . Nfx,Nfy,Nfz)
*
* CALCULATE W1P
*
      IF(HG .EQ. 'L')THEN
       W1fx=-SNPIT1
       WPfx=-MP1*CSD1
       W1fy=0.D00
       WPfy=0.D00
       W1fz=-CSPIT1
       WPfz=MP1*SND1
      ELSE
       W1fx=SNPIT1
       WPfx=MP1*CSD1
       W1fy=0.D00
       WPfy=0.D00
       W1fz=CSPIT1
       WPfz=-MP1*SND1
      END IF
      W1Pfx=W1fx-WPfx
      W1Pfy=W1fy-WPfy
      W1Pfz=W1fz-WPfz
*
* CALCULATE VT2, VTG, AND VT2G
*
*
      CALL CROSS(VT1fx,VT1fy,VT1fz,W1fx,W1fy,W1fz,Bfx,Bfy,Bfz)
      CALL CROSS(VTP1fx,VTP1fy,VTP1fz,WPfx,WPfy,WPfz,Bfx,Bfy,Bfz)
      IF(HG .EQ. 'L')THEN
       CALL CROSS(VTP2fx,VTP2fy,VTP2fz,TPAR(18),EM,TPAR(19),
     .             WPfx,WPfy,WPfz)
      ELSE
       CALL CROSS(VTP2fx,VTP2fy,VTP2fz,TPAR(18),-EM,TPAR(19),
     .             WPfx,WPfy,WPfz)
       END IF
      VTPfx=VTP1fx+VTP2fx
      VTPfy=VTP1fy+VTP2fy
      VTPfz=VTP1fz+VTP2fz
      VT1Pfx=VT1fx-VTPfx
      VT1Pfy=VT1fy-VTPfy
      VT1Pfz=VT1fz-VTPfz
      CALL DOT(VPI,EPIfx,EPIfy,EPIfz,VT1Pfx,VT1Pfy,VT1Pfz)
      CALL DOT(VPII,EPIIfx,EPIIfy,EPIIfz,VT1Pfx,VT1Pfy,VT1Pfz)
```

```
*
* CALCULATE A13,A23,A33
*
      CALL DET(DETI,W1Pfx,W1Pfy,W1Pfz,Nfx,Nfy,Nfz,EPIfx,EPIfy,EPIfz)
      A13=-KPI*VPI-DETI
      CALL DET(DETII,W1Pfx,W1Pfy,W1Pfz,Nfx,Nfy,Nfz,
     .         EPIIfx,EPIIfy,EPIIfz)
      A23=-KPII*VPII-DETII
      CALL DET(DET3,Nfx,Nfy,Nfz,W1Pfx,W1Pfy,W1Pfz,
     .         VT1Pfx,VT1Pfy,VT1Pfz)
      CALL CROSS(Cx,Cy,Cz,W1fx,0.D00,W1fz,VTPfx,VTPfy,VTPfz)
      CALL CROSS(Dx,Dy,Dz,WPfx,0.D00,WPfz,VT1fx,VT1fy,VT1fz)
      CALL DOT(DET45,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KPI*VPI*VPI+KPII*VPII*VPII-DET3-DET45
*
* CALCULATE SIGMA
*
      P=A23*A23-A13*A13+(KPI-KPII)*A33
      SIGDBL=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGDBL
*
* CALCULATE K1I AND K1II
*
      G1=P/(A33*DCOS(SIGDBL))
      KS1=KPI+KPII-(A13*A13+A23*A23)/A33
      K1I=(KS1+G1)/2.D00
      K1II=(KS1-G1)/2.D00
*
* CALCULATE E1I AND E1II
*
      CALL ROTATE(E1Ifx,E1Ify,E1Ifz,EPIfx,EPIfy,EPIfz,-SIGMA,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E1IIfx,E1IIfy,E1IIfz,E1Ifx,E1Ify,E1Ifz,RITAG,
     . Nfx,Nfy,Nfz)
      END
*
* FOR THE DETERMINATION OF THE ANGLE BETWEEN GEAR PRINCIPAL DIRECTIONS
* AND PINION PRINCIPAL DIRECTIONS
*
      SUBROUTINE SIGAN2(E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz,E1Ifx,
     . E1Ify,E1Ifz,CS2SIG,SN2SIG,SIGM12)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CALL DOT(CSSIG,E1Ifx,E1Ify,E1Ifz,E2Ifx,E2Ify,E2Ifz)
      CALL DOT(SNSIG,E1Ifx,E1Ify,E1Ifz,-E2IIfx,-E2IIfy,-E2IIfz)
      SIGM2=4.D00*DATAN(SNSIG/(1.D00+CSSIG))
      SIGM12=.5D00*SIGM2
      CS2SIG=DCOS(SIGM2)
      SN2SIG=DSIN(SIGM2)
      END
*
* FOR THE DETERMINATION OF CONTACT ELLIPS
*
      SUBROUTINE ELLIPS(KS2,G2,KS1,G1,CS2SIG,SN2SIG,DEF,ALFA1,
```

```
                      AXISL,AXISS,E1Ifx,E1Ify,E1Ifz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      COMMON/A3/TND1,TND2,RITAG
      COMMON/C2/Nfx,Nfy,Nfz
      COMMON/E1/XBf,YBf,ZBf
      D=DSQRT(G1*G1-2.D00*G1*G2*CS2SIG+G2*G2)
      CS2AF1=(G1-G2*CS2SIG)/D
      SN2AF1=G2*SN2SIG/D
      ALFA1=DATAN(SN2AF1/(1.D00+CS2AF1))
      A=.25D00*DABS(KS1-KS2-D)
      B=.25D00*DABS(KS1-KS2+D)
      IF(KS2 .LT. KS1)THEN
      AXISL=DSQRT(DEF/A)
      AXISS=DSQRT(DEF/B)
      CALL ROTATE(XBf,YBf,ZBf,E1Ifx,E1Ify,E1Ifz,RITAG-ALFA1,Nfx,
     . Nfy,Nfz)
      ELSE
      AXISL=DSQRT(DEF/B)
      AXISS=DSQRT(DEF/A)
      CALL ROTATE(XBf,YBf,ZBf,E1Ifx,E1Ify,E1Ifz,-ALFA1,Nfx,Nfy,
     . Nfz)
      END IF
      XBf=AXISL*XBf
      YBf=AXISL*YBf
      ZBf=AXISL*ZBf
      END
*
* COORDINATE TRANSFORMATION FOR F TO P
*
      SUBROUTINE PF(B2px,B2py,B2pz,Bfx,Bfy,Bfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
*
* [Mwf]=[Mwa] [Maf]
*
      CALL COMBI(w11,w12,w13,w21,w22,w23,w31,w32,w33,w1,w2,w3,
     . CSPH21,SNPH21,0.D00,-SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     . 0.D00,0.D00,0.D00,
     . CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     . 0.D00,0.D00,0.D00)
      CALL TRCOOR(B2wx,B2wy,B2wz,
     . w11,w12,w13,w21,w22,w23,w31,w32,w33,w1,w2,w3,
     . Bfx,Bfy,Bfz)
*
* [Mpw]=[Mpa] [Maw]
*
      CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     . 0.D00,0.D00,0.D00,
     . CSPH2,-SNPH2,0.D00,SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     . 0.D00,0.D00,0.D00)
      CALL TRCOOR(B2px,B2py,B2pz,
```

```
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . B2wx,B2wy,B2wz)
      END
*
* USING EULER FORMULA TO DETERMINATION SURFACE INTERFERENCE
*
      SUBROUTINE EULER(KS2,G2,KS1,G1,CS2SIG,SN2SIG,IEU)
      IMPLICIT REAL*8(A-H,K,M-Z)
      T=KS2-KS1
      U=DSQRT((G2-G1*CS2SIG)**2+(G1*SN2SIG)**2)
      KR1=(T+U)/2.D00
      KR2=(T-U)/2.D00
      IF(KR1*KR2 .LT. 0.D00)THEN
       IEU=1
      ELSE
       IEU=0
      END IF
      END
*
* DETERMINANT
*
      SUBROUTINE DET(S,A,B,C,D,E,F,G,H,P)
      IMPLICIT REAL*8(A-H,K,M-Z)
      S=A*E*P+D*H*C+G*B*F-A*H*F-D*B*P-G*E*C
      RETURN
      END
*
* COORDINATE TRANSFORMATION
*
      SUBROUTINE TRCOOR(XN,YN,ZN,R11,R12,R13,R21,R22,R23,R31,R32,R33,
                        T1,T2,T3,XP,YP,ZP)
      IMPLICIT REAL*8(A-H,O-Z)
      XN=R11*XP+R12*YP+R13*ZP+T1
      YN=R21*XP+R22*YP+R23*ZP+T2
      ZN=R31*XP+R32*YP+R33*ZP+T3
      RETURN
      END
*
* MULTIPLICATION OF TWO TRANSFORMATION MATRICES
*
      SUBROUTINE COMBI(C11,C12,C13,C21,C22,C23,C31,C32,C33,C1,C2,C3,
     .                 A11,A12,A13,A21,A22,A23,A31,A32,A33,A1,A2,A3,
     .                 B11,B12,B13,B21,B22,B23,B31,B32,B33,B1,B2,B3)
      IMPLICIT REAL*8(A-H,M,N,O-Z)
      C11=B31*A13+B21*A12+B11*A11
      C12=B32*A13+B22*A12+B12*A11
      C13=B33*A13+B23*A12+B13*A11
      C21=B31*A23+B21*A22+B11*A21
      C22=B32*A23+B22*A22+B12*A21
      C23=B33*A23+B23*A22+B13*A21
      C31=B31*A33+B21*A32+B11*A31
      C32=B32*A33+B22*A32+B12*A31
      C33=B33*A33+B23*A32+B13*A31
```

```
      C1=B3*A13+B2*A12+B1*A11+A1
      C2=B3*A23+B2*A22+B1*A21+A2
      C3=B3*A33+B2*A32+B1*A31+A3
      RETURN
      END
*
* DOT OF TWO VECTORS
*
      SUBROUTINE DOT(V,X1,Y1,Z1,X2,Y2,Z2)
      IMPLICIT REAL*8(A-H,O-Z)
      V=X1*X2+Y1*Y2+Z1*Z2
      RETURN
      END
*
* CROSS OF TWO VECTORS
*
      SUBROUTINE CROSS(X,Y,Z,A,B,C,D,E,F)
      IMPLICIT REAL*8(A-H,O-Z)
      X=B*F-C*E
      Y=C*D-A*F
      Z=A*E-B*D
      RETURN
      END
*
* ROTATION A VECTOR ABOUT ANOTHER VECTOR
*
      SUBROUTINE ROTATE(XN,YN,ZN,XP,YP,ZP,THETA,UX,UY,UZ)
      IMPLICIT REAL*8(A-H,O-Z)
      CT=DCOS(THETA)
      ST=DSIN(THETA)
      VT=1.D00-CT
      R11=UX*UX*VT+CT
      R12=UX*UY*VT-UZ*ST
      R13=UX*UZ*VT+UY*ST
      R21=UX*UY*VT+UZ*ST
      R22=UY*UY*VT+CT
      R23=UY*UZ*VT-UX*ST
      R31=UX*UZ*VT-UY*ST
      R32=UY*UZ*VT+UX*ST
      R33=UZ*UZ*VT+CT
      CALL TRCOOR(XN,YN,ZN,R11,R12,R13,R21,R22,R23,R31,R32,R33,
     .            0.D00,0.D00,0.D00,
     .            XP,YP,ZP)
      RETURN
      END
*
*     *****   SUBROUTINE NOLIN   *****
*
      SUBROUTINE NONLIN(FUNC,NSIG,NE,NC,X,Y,Y1,DELTA,A,IPVT,WORK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(NE),Y(NE),Y1(NE),A(NE,NE),IPVT(NE),WORK(NE)
      EXTERNAL FUNC
      NDIM=NE
```

```
      EPSI=1.D00/10.D00**NSIG
      CALL NONLIO(FUNC,EPSI,NE,NC,X,DELTA,NDIM,A,Y,Y1,WORK,IPVT)
      RETURN
      END
*
*     *****    SUBROUTINE NOLINO    *****
*
      SUBROUTINE NONLIO(FUNC,EPSI,NE,NC,X,DELTA,NDIM,A,Y,Y1,WORK,IPVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(NE),Y(NE),Y1(NE),IPVT(NE),WORK(NE),A(NDIM,NE)
      EXTERNAL FUNC
* NC: # OF COUNT TIMES
      DO 5 I=1,NC
      CALL FUNC(X,Y,NE)
* NE: # OF EQUATIONS
      DO 15 J=1,NE
      IF (DABS(Y(J)).GT.EPSI) GO TO 25
   15 CONTINUE
      GO TO 105
   25 DO 35 J=1,NE
   35 Y1(J)=Y(J)
      DO 45 J=1,NE
      DIFF=DABS(X(J))*DELTA
      IF (DABS(X(J)).LT.1.D-12) DIFF=DELTA
      XMAM=X(J)
      X(J)=X(J)-DIFF
      CALL FUNC(X,Y,NE)
      X(J)=XMAM
      DO 55 K=1,NE
      A(K,J)=(Y1(K)-Y(K))/DIFF
   55 CONTINUE
   45 CONTINUE
      DO 65 J=1,NE
   65 Y(J)=-Y1(J)
      CALL DECOMP (NDIM,NE,A,COND,IPVT,WORK)
      CALL SOLVE (NDIM,NE,A,Y,IPVT)

      DO 75 J=1,NE
      X(J)=X(J)+Y(J)
   75 CONTINUE
    5 CONTINUE
  105 RETURN
      END
*
*     *****    SUBROUTINE DECOMP    *****
*
      SUBROUTINE DECOMP (NDIM,N,A,COND,IPVT,WORK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,N),WORK(N),IPVT(N)
*
*  DECOMPOSES A REAL MATRIX BY GAUSSIAN ELIMINATION,
*  AND ESTIMATES THE CONDITION OF THE MATRIX.
*
```

```
*  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
*       M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
*
*  USE SUBROUTINE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEM.
*
*  INPUT..
*
*     NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING   A
*     N    = ORDER OF THE MATRIX
*     A    = MATRIX TO BE TRIANGULARIZED
*
*  OUTPUT..
*
*     A   CONTAINS AN UPPER TRIANGULAR MATRIX  U  AND A PREMUTED
*         VERSION OF A LOWER TRIANGULAR MATRIX  I-L  SO THAT
*         (PERMUTATION MATRIX)*A=L*U
*
*     COND = AN ESTIMATE OF THE CONDITION OF  A.
*       FOR THE LINEAR SYSTEM  A*X = B , CHANGES IN   A  AND  B
*       MAY CAUSE CHANGES  COND  TIMES AS LARGE IN  X.
*       IF  COND+1.0 .EQ. COND , A IS SINGULAR TO WORKING
*       PRECISION.  COND  IS SET TO  1.0D+32  IF EXACT
*       SINGULARITY IS DETECTED.
*
*     IPVT      = THE PIVOT VECTOR
*       IPVT(K) = THE INDEX OF THE K-TH PIVOT ROW
*       IPVT(N) = (-1)**(NUMBER OF INTERCHANGES)
*
*  WORK SPACE..   THE VECTOR  WORK  MUST BE DECLARED AND INCLUDED
*       IN THE CALL.  ITS INPUT CONTENTS ARE IGNORED.
*       ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
*
*  THE DETERMINANT OF  A  CAN BE OBTAINED ON OUTPUT BY
*     DET(A) = IPVT(N) * A(1,1) * A(2,2) * ... * A(N,N) .
*
      IPVT(N)=1
      IF (N.EQ.1) GO TO 150
      NM1=N-1
*                             COMPUTE THE 1-NORM OF  A .
      ANORM=0.D0
      DO 20 J=1,N
        T=0.D0
        DO 10 I=1,N
   10   T=T+DABS(A(I,J))
        IF (T.GT.ANORM) ANORM=T
   20 CONTINUE
*                             DO GAUSSIAN ELIMINATION WITH PARTIAL
*                                  PIVOTING.
      DO 70 K=1,NM1
        KP1=K+1
*                             FIND THE PIVOT.
        M=K
        DO 30 I=KP1,N
```

```
          IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
   30   CONTINUE
        IPVT(K)=M
        IF (M.NE.K) IPVT(N)=-IPVT(N)
        T=A(M,K)
        A(M,K)=A(K,K)
        A(K,K)=T
*                                  SKIP THE ELIMINATION STEP IF PIVOT IS ZERO.
        IF (T.EQ.0.D0) GO TO 70
*
*                                  COMPUTE THE MULTIPLIERS.
        DO 40 I=KP1,N
   40   A(I,K)=-A(I,K)/T
*                                  INTERCHANGE AND ELIMINATE BY COLUMNS.
        DO 60 J=KP1,N
          T=A(M,J)
          A(M,J)=A(K,J)
          A(K,J)=T
          IF (T.EQ.0.D0) GO TO 60
          DO 50 I=KP1,N
   50     A(I,J)=A(I,J)+A(I,K)*T
   60   CONTINUE
   70 CONTINUE

*
*  COND = (1-NORM OF A)*(AN ESTIMATE OF THE 1-NORM OF A-INVERSE)
*  THE ESTIMATE IS OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
*  SMALL SINGULAR VECTOR.  THIS INVOLVES SOLVING TWO SYSTEMS
*  OF EQUATIONS,  (A-TRANSPOSE)*Y = E  AND  A*Z = Y  WHERE  E
*  IS A VECTOR OF +1 OR -1 COMPONENTS CHOSEN TO CAUSS GROWTH IN Y.
*  ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
*
*                                  SOLVE  (A-TRANSPOSE)*Y = E .
      DO 100 K=1,N
        T=0.D0
        IF (K.EQ.1) GO TO 90
        KM1=K-1
        DO 80 I=1,KM1
   80   T=T+A(I,K)*WORK(I)

   90   EK=1.D0
        IF (T.LT.0.D0) EK=-1.D0

        IF (A(K,K).EQ.0.D0) GO TO 160
        A11=A(1,1)
      WORK(K)=-(EK+T)/A(1,1)
  100 CONTINUE
      DO 120 KB=1,NM1
        K=N-KB
        T=0.D0
        KP1=K+1
        DO 110 I=KP1,N
  110   T=T+A(I,K)*WORK(K)
```

```
        WORK(K)=T
        M=IPVT(K)
        IF (M.EQ.K) GO TO 120
        T=WORK(M)
        WORK(M)=WORK(K)
        WORK(K)=T
  120 CONTINUE
*
      YNORM=0.D0
      DO 130 I=1,N
  130 YNORM=YNORM+DABS(WORK(I))
*
*                                SOLVE  A*Z = Y
      CALL SOLVE (NDIM,N,A,WORK,IPVT)
*
      ZNORM=0.D0
      DO 140 I=1,N
  140 ZNORM=ZNORM+DABS(WORK(I))
*
*                                ESTIMATE THE CONDITION.
      COND=ANORM*ZNORM/YNORM
      IF (COND.LT.1.D0) COND=1.D0
      RETURN
*                                1-BY-1 CASE..
  150 COND=1.D0
      IF (A(1,1).NE.0.D0) RETURN
*
*                                EXACT SINGULARITY
  160 COND=1.0D32
      RETURN
      END

      SUBROUTINE SOLVE (NDIM,N,A,B,IPVT)
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,N),B(N),IPVT(N)
*
*  SOLVES A LINEAR SYSTEM,  A*X = B
*  DO NOT SOLVE THE SYSTEM IF DECOMP HAS DETECTED SINGULARITY.
*
*  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
*       M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
*
*  INPUT..
*
*     NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING  A
*     N    = ORDER OF MATRIX
*     A    = TRIAN2ULARIZED MATRIX OBTAINED FROM SUBROUTINE DECOMP
*     B    = RIGHT HAND SIDE VECTOR
*     IPVT = PIVOT VECTOR OBTAINED FROM DECOMP
*
*  OUTPUT..
*
```

```
*      B = SOLUTION VECTOR,  X
*
*                                 DO THE FORWARD ELIMINATION.
       IF (N.EQ.1) GO TO 50
       NM1=N-1
       DO 20 K=1,NM1
         KP1=K+1
         M=IPVT(K)
         T=B(M)
         B(M)=B(K)
         B(K)=T
         DO 10 I=KP1,N
   10    B(I)=B(I)+A(I,K)*T
   20  CONTINUE
*                                 NOW DO THE BACK SUBSTITUTION.
       DO 40 KB=1,NM1
         KM1=N-KB
         K=KM1+1
         B(K)=B(K)/A(K,K)
         T=-B(K)
         DO 30 I=1,KM1
   30    B(I)=B(I)+A(I,K)*T
   40  CONTINUE
   50  B(1)=B(1)/A(1,1)
       RETURN
       END
```

```
******************************************************************************
*                                                                            *
*                     Gleason's Spiral Bevel Gears                           *
*                                                                            *
*         Basic Machine-Tool Settings and Tooth Contact Analysis             *
*                                                                            *
*                     Curved Blade to Cut the Pinion                         *
*                                                                            *
******************************************************************************
      IMPLICIT REAL*8(A-H,K,M-Z)
      REAL*8 X(1),F(1),FI(1),PAR(6),LM,TX(6),TF(6),TF1(6),TPAR(19),
     .       AZSP(1,1),WORKP(1),AZS(6,6),WORK(6),LANDAP,LANPDG,LANDPO
      CHARACTER*8 HG,HNGR
      DIMENSION IPVT(6),IPVTP(1)
      EXTERNAL PCN1,PCN2,TCN
      COMMON/P1/PAR
      COMMON/T1/TPAR
      COMMON/A0/HG
      COMMON/A1/p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/A5/CSQG,SNQG,THETAG
      COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSLANP,SNLANP
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/B2fx,B2fy,B2fz
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
      COMMON/B5/XCR,ZCR
      COMMON/C1/UG,CSTAUG,SNTAUG
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/D1/CSTAUP,SNTAUP
      COMMON/E1/XBf,YBf,ZBf
      COMMON/F1/PHIGO
      COMMON/G1/DA1,DV1
*
* INPUT THE DESIGN DATA
*
*
* TN1               : number of pinion teeth
*                     ----- sec. 3.1
* TN2               : number of gear teeth
*                     ----- sec. 3.1
* RT1dg, RT1min     : root angle of pinion (degree and arc minute, respec-
*                     tively)
*                     ----- sec. 3.1
* RT2dg, RT2min     : root angle of gear (degree and arc minute, respec-
*                     tively)
*                     ----- sec. 3.1
* SHAFdg            : shaft angle (degree)
*                     ----- sec. 3.1
* BETAdg            : mean spiral angle (degree)
*                     ----- sec. 3.1
* ADIA              : average gear cutter diameter
```

```
*                     ----- sec. 3.1
* W                 : point width of gear cutter
*                     ----- sec. 3.1
* A                 : mean cone distance
*                     ----- sec. 3.1
* ALPHdg            : blade angle of gear cutter (degree)
*                     ----- sec. 3.1
* RX                : radius of blade
*                     gear convex side
*                     ----- fig. 10
* RV                : radius of blade
*                     gear concave side
*                     ----- fig. 10
* DLTXdg            : angle measured counterclockwise from root of gear to
*                     the tangent of the contact path (degree)
*                     gear convex side
*                     ----- fig. 19
* DLTVdg            : angle measured counterclockwise from root of gear to
*                     the tangent of the contact path (degree)
*                     gear concave side
*                     ----- fig. 19
* M21XPR            : first derivative of gear ratio
*                     gear convex side
*                     ----- sec. 3.1.1
* M21VPR            : first derivative of gear ratio
*                     gear concave side
*                     ----- sec. 3.1.1
* AXILX             : semimajor axis of contact ellipse
*                     gear convex side
*                     ----- eq. (3.76)
* AXILV             : semimajor axis of contact ellipse
*                     gear concave side
*                     ----- eq. (3.76)
* HNGR              : hand of gear ('L' or 'R')
* DA                : amount of shift along pinion axis
*                     + : pinion mounting distance being increased
*                     - : pinion mounting distance being decreased
* DV                : amount of pinion shaft offset
*                     the same sense as yf shown in fig. 18
* DEF               : elastic approach
*                     ----- eq. (3.76)
* EPS               : amount to control calculation accuracy
*
* OUTPUT OF THE BASIC MACHINE-TOOL SETTINGS
*
* PSIGdg            : gear blade angle
* PSIPdg            : pinion blade angle
* RG                : tip radius of gear cutter
* RP                : tip radius of pinion cutter
* SG                : gear radial
* SP                : pinion radial
* QGdg              : gear cradle angle
* QPdg              : pinion cradle angle
```

```
* MG2              : gear cutting ratio
* MP1              : pinion cutting ratio
* EM               : machining offset
* LM               : machine center to back + sliding base
* XCR, ZCR         : x and z coordinates of center of blade
*
      DATA TN1,TN2/10.D00,41.D00/
      DATA RT1dg,RT1min/12.D00,1.D00/
      DATA RT2dg,RT2min/72.D00,25.D00/
      DATA SHAFdg,BETAdg/90.D00,35.D00/
      DATA ADIA/6.0D00/
      DATA W/0.08D00/
      DATA A/3.226D00/
      DATA ALPHdg/20.D00/
      DATA DLTXdg/ 90.D00/
      DATA DLTVdg/ 75.D00/
      DATA M21XPR/-3.5D-03/
      DATA M21VPR/5.2D-03/
      DATA AXILX/0.1710D00/
      DATA AXILV/0.1810D00/
      DATA RX/40.00D00/
      DATA RV/50.00D00/
      DATA HNGR/'L'/
      DATA DA,DV/0.D00,0.D00/
      DATA DEF/0.00025D00/
      DATA EPS/1.D-12/
*
*
*
      DA1=DA
      DV1=DV
      HG=HNGR
*
* CONVERT DEGREES TO RADIANS
*
      CNST=4.D00*DATAN(1.D00)/180.D00
      RITAG=90.D00*CNST
      DLTX=DLTXdg*CNST
      DLTV=DLTVdg*CNST
      RT1=(RT1dg+RT1min/60.D00)*CNST
      RT2=(RT2dg+RT2min/60.D00)*CNST
      BETA=BETAdg*CNST
      PSIG=ALPHdg*CNST
      SHAFT=SHAFdg*CNST
      CSRT2=DCOS(RT2)
      SNRT2=DSIN(RT2)
      CSRT1=DCOS(RT1)
      SNRT1=DSIN(RT1)
*
* CALCULATE PITCH ANGLES
*
      MM21=TN1/TN2
c ----- eq. (3.1)
```

```
      PITCH2=DATAN(DSIN(SHAFT)/(MM21+DCOS(SHAFT)))
      IF(PITCH2 .LT. 0.D00)THEN
       PITCH2=PITCH2+180.D00
      END IF
      CSPIT2=DCOS(PITCH2)
      SNPIT2=DSIN(PITCH2)
c ----- eq. (3.2)
      PITCH1=SHAFT-PITCH2
      CSPIT1=DCOS(PITCH1)
      SNPIT1=DSIN(PITCH1)
*
* CALCULATE DEDENDUM ANGLES
*
c ----- eq. (3.3)
      D1=PITCH1-RT1
      D2=PITCH2-RT2
      CSD1=DCOS(D1)
      SND1=DSIN(D1)
      TND1=SND1/CSD1
      CSD2=DCOS(D2)
      SND2=DSIN(D2)
      TND2=SND2/CSD2
*
* CALCULATE GEAR CUTTING RATIO
*
c ----- eq. (3.7)
      MG2=DSIN(PITCH2)/CSD2
*
* FOR GEAR CONVEX SIDE I = 1, FOR GEAR CONCAVE SIDE I = 2.
*
      DO 99999 I=1,2
      IF(I .EQ. 1)THEN
       WRITE(72,*)'GEAR CONVEX SIDE'
       DLTA=DLTX
       M21PRM=M21XPR
       AXIL=AXILX
       R=RX
      ELSE
       WRITE(72,*)'GEAR CONCAVE SIDE'
       DLTA=DLTV
       M21PRM=M21VPR
       AXIL=AXILV
       R=RV
      END IF
      WRITE(72,*)
c ----- eq. (3.76)
      AXIA=DEF/(AXIL*AXIL)
*
* CALCULATE GEAR BLADE ANGLE
*
c ----- sec. 2.2
      IF(I .EQ. 2)THEN
       PSIG=180.D00*CNST-PSIG
```

```
      END IF
      CSPSIG=DCOS(PSIG)
      SNPSIG=DSIN(PSIG)
      CTPSIG=CSPSIG/SNPSIG
*
* CALCULATE CUTTER TIP RADIUS
*
c ----- eq. (3.8)
      IF(I .EQ. 1)THEN
       RG=(ADIA-W)/2.D00
      ELSE
       RG=(ADIA+W)/2.D00
      END IF
*
* CALCULATE RADIAL
*
c ----- eq. (3.9)
      IF(I .EQ. 1)THEN
       SG=DSQRT(ADIA*ADIA/4.D00+A*A*CSD2*CSD2-A*ADIA*CSD2*DSIN(BETA))
*
* CALCULATE CRADLE ANGLE
*
c ----- eq. (3.10)
       QG=DACOS((A*A*CSD2*CSD2+SG*SG-ADIA*ADIA/4.D00)/(2.D00*A*SG*CSD2))
       CSQG=DCOS(QG)
       SNQG=DSIN(QG)
      END IF
*
      PAR(1)=RG*CTPSIG*CSPSIG
      PAR(4)=RG*CTPSIG
*
* CALCULATE PHIG
*
      PHIG=0.D00
      PHIGO=PHIG
      CSPHIG=DCOS(PHIG)
      SNPHIG=DSIN(PHIG)
*
      IF(HG .EQ. 'L')THEN
       IF(I .EQ. 1)THEN
* Mmc=Mms*Msc
c ----- eq. (2.26)
        CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .   1.D00,0.D00,0.D00,0.D00,CSPHIG,SNPHIG,0.D00,-SNPHIG,CSPHIG,
     .   0.D00,0.D00,0.D00,
     .   1.D00,0.D00,0.D00,0.D00,CSQG,-SNQG,0.D00,SNQG,CSQG,
     .   0.D00,-SG*SNQG,SG*CSQG)
       END IF
* Mpc=Mpm*Mmc
c ----- eqs. (2.25), (3.13)
       CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     .  CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .  0.D00,0.D00,0.D00,
```

```
     .  m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
*
      ELSE
*
        IF(I .EQ. 1)THEN
* Mmc=Mms*Msc
c ----- eq. (2.26)
        CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .   1.D00,0.D00,0.D00,0.D00,CSPHIG,-SNPHIG,0.D00,SNPHIG,CSPHIG,
     .   0.D00,0.D00,0.D00,
     .   1.D00,0.D00,0.D00,0.D00,CSQG,SNQG,0.D00,-SNQG,CSQG,
     .   0.D00,SG*SNQG,SG*CSQG)
        END IF
* Mpc=Mpm*Mmc
c ----- eqs. (2.25), (3.13)
        CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     .  CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .  0.D00,0.D00,0.D00,
     .  m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
       END IF
*
* DETERMINE MAIN CONTACT POINT
*
*
* CALCULATE THETAG
*
c ----- X(1) represents THETAG
       PAR(2)=(MG2-SNRT2)*CSPSIG
       IF(HG .EQ. 'L')THEN
        PAR(3)=-SNQG*CSRT2*SNPSIG
c ----- step 1 in sec. 3.2
        X(1)=QG-BETA+RITAG
       ELSE
        PAR(3)=SNQG*CSRT2*SNPSIG
c ----- step 1 in sec. 3.2
        X(1)=-(QG-BETA+RITAG)
       END IF
       CALL NONLIN(PCN1,14,1,100,X,F,FI,1.D-5,AZSP,IPVTP,WORKP)
      THETAG=X(1)
      CSTHEG=DCOS(THETAG)
      SNTHEG=DSIN(THETAG)
*
* CALCULATE TAUG
*
c ----- eq. (2.38)
      IF(HG .EQ. 'L')THEN
       TAUG=THETAG-QG+PHIG
      ELSE
       TAUG=THETAG+QG-PHIG
      END IF
      CSTAUG=DCOS(TAUG)
      SNTAUG=DSIN(TAUG)
*
```

```
* CALCULATE UG
*
c ----- eq. (2.43)
      IF(HG .EQ. 'L')THEN
       UG=RG*CTPSIG*CSPSIG-SG*((MG2-SNRT2)*CSPSIG*SNTHEG-DSIN(QG-PHIG)*
     #    CSRT2*SNPSIG)/(CSRT2*SNTAUG)
      ELSE
       UG=RG*CTPSIG*CSPSIG-SG*((MG2-SNRT2)*CSPSIG*SNTHEG+DSIN(QG-PHIG)*
     #    CSRT2*SNPSIG)/(CSRT2*SNTAUG)
      END IF
*
* CONVERT RADIAN TO DEGREE
*
      PSIGDG=PSIG/CNST
      TAUGDG=TAUG/CNST
      QGDG=QG/CNST
      THEGDG=THETAG/CNST
      PHIGDG=PHIGO/CNST
*
* OUTPUT OF GEAR SETTINGS
*
      WRITE(72,10000)PSIGDG,QGDG,RG,SG,MG2,TAUGDG,UG,THEGDG,PHIGDG
*
* CALCULATE MAIN CONTACT POINT
*
c ----- eq. (2.1)
      Bcx=RG*CTPSIG-UG*CSPSIG
      Bcy=UG*SNPSIG*SNTHEG
      Bcz=UG*SNPSIG*CSTHEG
c ----- eq. (2.2)
      Ncx=SNPSIG
      Ncy=CSPSIG*SNTHEG
      Ncz=CSPSIG*CSTHEG
c ----- eq. (2.9)
      EGIcx=0.D00
      EGIcy=CSTHEG
      EGIcz=-SNTHEG
c ----- eq. (3.13)
      CALL TRCOOR(Bpx,Bpy,Bpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . Bcx,Bcy,Bcz)
c ----- eq. (3.16)
      CALL TRCOOR(Npx,Npy,Npz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,0.D00,0.D00,0.D00,
     . Ncx,Ncy,Ncz)
c ----- eq. (3.17)
      CALL TRCOOR(EGIpx,EGIpy,EGIpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,0.D00,0.D00,0.D00,
     . EGIcx,EGIcy,EGIcz)
c ----- fig. 18 & sec. 3.3
      Bfx=Bpx
      Bfy=Bpy
      Bfz=Bpz
```

```
      Nfx=Npx
      Nfy=Npy
      Nfz=Npz
      EGIfx=EGIpx
      EGIfy=EGIpy
      EGIfz=EGIpz
*
* CALCULATE LANDAP
*
      UG0=UG
      CSTAG0=CSTAUG
      SNTAG0=SNTAUG
      PHIG0=PHIG
      THETG0=THETAG
*
      DO 99999 J=1,2
*
      CSTAUG=CSTAG0
      SNTAUG=SNTAG0
      UG=UG0
      PHIG=PHIG0
      THETAG=THETG0

      IF(J .EQ. 1)THEN
       WRITE(72,*)'BLADE CONCAVE DOWN'
      ELSE
       WRITE(72,*)'BLADE CONCAVE UP'
      END IF
      LANDAP=DACOS(CSD1*Nfx-SND1*Nfz)
      IF (I .EQ. 1)THEN
       IF (J .EQ. 1)THEN
        LANDAP=360.D00*CNST-LANDAP
        PSIP=450.D00*CNST-LANDAP
       ELSE
        LANDAP=180.D00*CNST-LANDAP
        PSIP=270.D00*CNST-LANDAP
       END IF
      ELSE
       IF(J .EQ. 1)THEN
        PSIP=90.D00*CNST-LANDAP
       ELSE
        LANDAP=180.D00*CNST+LANDAP
        PSIP=270.D00*CNST-LANDAP
       END IF
      END IF
      CSLANP=DCOS(LANDAP)
      SNLANP=DSIN(LANDAP)
*
* CALCULATE TAUP
*
      TAUP=DATAN2(Nfy/SNLANP,(Nfx-CSD1*CSLANP)/(-SND1*SNLANP))
      IF(J .EQ. 2)THEN
      TAUP=DATAN2(-Nfy/SNLANP,(-Nfx-CSD1*CSLANP)/(-SND1*SNLANP))
```

```
      END IF
      CSTAUP=DCOS(TAUP)
      SNTAUP=DSIN(TAUP)
*
* CALCULATE PRINCIPAL CURVATURES AND DIRECTIONS OF THE GEAR CUTTER
*
c ----- eq. (2.10)
      KGI=-CTPSIG/UG
c ----- eq. (2.12)
      KGII=0.D00
c ----- the second principal direction is determined by rotating of
c ----- the first principal derection about unit normal by 90 degrees
      CALL ROTATE(EGIIfx,EGIIfy,EGIIfz,EGIfx,EGIfy,EGIfz,RITAG,
     . Nfx,Nfy,Nfz)
*
* CALCULATE W2G
*
c ----- eqs. (3.18)-(3.20)
      IF(HG .EQ. 'L')THEN
       W2fx=-SNPIT2
       WGfx=-MG2*CSD2
       W2fy=0.D00
       WGfy=0.D00
       W2fz=CSPIT2
       WGfz=-MG2*SND2
      ELSE
       W2fx=SNPIT2
       WGfx=MG2*CSD2
       W2fy=0.D00
       WGfy=0.D00
       W2fz=-CSPIT2
       WGfz=MG2*SND2
       END IF
*
      W2Gfx=W2fx-WGfx
      W2Gfy=W2fy-WGfy
      W2Gfz=W2fz-WGfz
*
* CALCULATE VT2, VTG, AND VT2G
*
c ----- eq. (3.22)
      CALL CROSS(VT2fx,VT2fy,VT2fz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
c ----- eq. (3.21)
      CALL CROSS(VTGfx,VTGfy,VTGfz,WGfx,WGfy,WGfz,Bfx,Bfy,Bfz)
c ----- eq. (3.23)
      VT2Gfx=VT2fx-VTGfx
      VT2Gfy=VT2fy-VTGfy
      VT2Gfz=VT2fz-VTGfz
*
* CALCULATE V(2G)GI AND V(2G)GII
*
c ----- eq. (3.24)
      CALL DOT(VGI,EGIfx,EGIfy,EGIfz,VT2Gfx,VT2Gfy,VT2Gfz)
```

```
c ----- eq. (3.25)
      CALL DOT(VGII,EGIIfx,EGIIfy,EGIIfz,VT2Gfx,VT2Gfy,VT2Gfz)
*
* CALCULATE A13,A23,A33
*
c ----- eq. (3.26)
      CALL DET(DETI,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIfx,EGIfy,EGIfz)
      A13=-KGI*VGI-DETI
c ----- eq. (3.27)
      CALL DET(DETII,W2Gfx,W2Gfy,W2Gfz,Nfx,Nfy,Nfz,EGIIfx,EGIIfy,EGIIfz)
      A23=-KGII*VGII-DETII
c ----- eq. (3.28)
      CALL DET(DET3,Nfx,Nfy,Nfz,W2Gfx,W2Gfy,W2Gfz,VT2Gfx,VT2Gfy,VT2Gfz)
      CALL CROSS(Cx,Cy,Cz,W2fx,W2fy,W2fz,VTGfx,VTGfy,VTGfz)
      CALL CROSS(Dx,Dy,Dz,WGfx,WGfy,WGfz,VT2fx,VT2fy,VT2fz)
      CALL DOT(DET45,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KGI*VGI*VGI+KGII*VGII*VGII-DET3-DET45
*
* CALCULATE SIGMA
*
c ----- eq. (3.29)
      P=A23*A23-A13*A13+(KGI-KGII)*A33
      SIGDBL=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGDBL
*
* CALCULATE K2I AND K2II
*
c ----- eqs. (3.30)-(3.31)
      T1=P/(A33*DCOS(SIGDBL))
      T2=KGI+KGII-(A13*A13+A23*A23)/A33
      K2I=(T1+T2)/2.D00
      K2II=(T2-T1)/2.D00
*
* CALCULATE E2I AND E2II
*
c ----- description after eq. (3.29)
      CALL ROTATE(E2Ifx,E2Ify,E2Ifz,EGIfx,EGIfy,EGIfz,-SIGMA,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E2IIfx,E2IIfy,E2IIfz,E2Ifx,E2Ify,E2Ifz,RITAG,
     . Nfx,Nfy,Nfz)
c ----- eq. (3.44)
      TNETAG=DSIN(DLTA+SIGMA)/DCOS(DLTA+SIGMA)
*
* CALCULATE W2
*
c ----- eq. (3.33)
      IF(HG .EQ. 'L')THEN
       W2fx=-MM21*SNPIT2
       W2fy=0.D00
       W2fz=MM21*CSPIT2
      ELSE
       W2fx=MM21*SNPIT2
       W2fy=0.D00
```

```
        W2fz=-MM21*CSPIT2
       END IF
*
* CALCULATE W1
*
c ----- eq. (3.32)
       IF(HG .EQ. 'L')THEN
        W1fx=-SNPIT1
        W1fy=0.D00
        W1fz=-CSPIT1
       ELSE
        W1fx=SNPIT1
        W1fy=0.D00
        W1fz=CSPIT1
       END IF
*
* CALCULATE W12
*
c ----- eq. (3.34)
       W12fx=W1fx-W2fx
       W12fy=W1fy-W2fy
       W12fz=W1fz-W2fz
*
* CALCULATE VT2
*
c ----- eq. (3.36)
       CALL CROSS(VT2fx,VT2fy,VT2fz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
*
* CALCULATE VT1
*
c ----- eq. (3.35)
       CALL CROSS(VT1fx,VT1fy,VT1fz,W1fx,W1fy,W1fz,Bfx,Bfy,Bfz)
*
* CALCULATE VT12
*
c ----- eq. (3.37)
       VT12fx=VT1fx-VT2fx
       VT12fy=VT1fy-VT2fy
       VT12fz=VT1fz-VT2fz
*
* CALCULATE V2
*
c ----- eq. (3.38)
       CALL DOT(V2I,VT12fx,VT12fy,VT12fz,E2Ifx,E2Ify,E2Ifz)
c ----- eq. (3.39)
       CALL DOT(V2II,VT12fx,VT12fy,VT12fz,E2IIfx,E2IIfy,E2IIfz)
*
* CALCULATE A31
*
c ----- eq. (3.40)
       CALL DET(DET1,W12fx,W12fy,W12fz,Nfx,Nfy,Nfz,E2Ifx,E2Ify,E2Ifz)
       A31=-K2I*V2I-DET1
c ----- eq. (A.33)
```

```
      A13=A31
*
* CALCULATE A32
*
c ----- eq. (3.41)
      CALL DET(DET2,W12fx,W12fy,W12fz,Nfx,Nfy,Nfz,E2IIfx,E2IIfy,E2IIfz)
      A32=-K2II*V2II-DET2
c ----- eq. (A.35)
      A23=A32
*
* CALCULATE A33
*
c ----- eq. (3.42)
      CALL DET(DET3,Nfx,Nfy,Nfz,W12fx,W12fy,W12fz,VT12fx,VT12fy,VT12fz)
      CALL CROSS(Cx,Cy,Cz,W1fx,W1fy,W1fz,VT2fx,VT2fy,VT2fz)
      CALL CROSS(Dx,Dy,Dz,W2fx,W2fy,W2fz,VT1fx,VT1fy,VT1fz)
      CALL DOT(DOT1,Nfx,Nfy,Nfz,Cx-Dx,Cy-Dy,Cz-Dz)
      CALL DET(DET4,Nfx,Nfy,Nfz,W2fx,W2fy,W2fz,Bfx,Bfy,Bfz)
      A33=K2I*V2I*V2I+K2II*V2II*V2II-DET3-DOT1+M21PRM*DET4
*
* CALCULATE ETAP
*
c ----- eq. (3.53)
      ETAP=DATAN(((A33+A31*V2I)*TNETAG-A31*V2II)/(A33+A32*
     . (V2II-V2I*TNETAG)))
      TNETAP=DSIN(ETAP)/DCOS(ETAP)
*
* CALCULATE A11, A12, AND A22
*
      N3=(1.D00+TNETAP*TNETAP)*A33
c ----- eq. (3.72)
      N1=(A13*A13-(A23*TNETAP)**2)/N3
c ----- eq. (3.73)
      N2=(A23+A13*TNETAP)*(A13+A23*TNETAP)/N3
      KS2=K2I+K2II
      G2=K2I-K2II
c ----- eqs. (3.74), (3.75)
      KS1=KS2-((4.D00*AXIA*AXIA-N1*N1-N2*N2)*(1.D00+TNETAP*TNETAP)/
     . (-2.D00*AXIA*(1.D00+TNETAP*TNETAP)+N1*(TNETAP*TNETAP-1.D00)
     . -2.D00*N2*TNETAP))
c ----- eqs. (3.66), (3.69) & description after eq. (3.60)
      A11=TNETAP*TNETAP/(1.D00+TNETAP*TNETAP)*(KS2-KS1)+N1
c ----- eqs. (3.67), (3.70) & description after eq. (3.60)
      A12=-TNETAP/(1.D00+TNETAP*TNETAP)*(KS2-KS1)+N2
c ----- eqs. (3.68), (3.71) & description after eq. (3.60)
      A22=1.D00/(1.D00+TNETAP*TNETAP)*(KS2-KS1)-N1
c ----- eq. (A.32)
      A21=A12
*
* CALCULATE SIGMA(12)
*
c ----- eq. (3.77)
      SIGDBL=DATAN(2.D00*A12/(K2I-K2II-A11+A22))
```

```
      SIGM12=.5D00*SIGDBL
*
* CALCULATE K1I AND K1II
*
c ----- eq. (3.78)
      G1=2.D00*A12/DSIN(SIGDBL)
c ----- eq. (3.79)
      K1I=.5D00*(KS1+G1)
      K1II=.5D00*(KS1-G1)
*
* CALCULATE E1I AND E1II
*
c ----- similar to description after eq. (3.29)
      CALL ROTATE(E1Ifx,E1Ify,E1Ifz,E2Ifx,E2Ify,E2Ifz,-SIGM12,Nfx,Nfy,
     . Nfz)
      CALL ROTATE(E1IIfx,E1IIfy,E1IIfz,E1Ifx,E1Ify,E1Ifz,RITAG,
     . Nfx,Nfy,Nfz)
*
* PINION
*
*
* CALCULATE PRINCIPAL DIRECTIONS OF THE PINION CUTTER
*
c ----- eq. (3.92)
      IF(HG .EQ. 'L')THEN
       EPIfx=SND1*SNTAUP
       EPIfy=CSTAUP
       EPIfz=CSD1*SNTAUP
      ELSE
       EPIfx=-SND1*SNTAUP
       EPIfy=-CSTAUP
       EPIfz=-CSD1*SNTAUP
      END IF
      IF(DACOS(EGIfx*EPIfx+EGIfy*EPIfy+EGIfz*EPIfz)/CNST .GT. 45.D00)
     .THEN
       EPIfx=-EPIfx
       EPIfy=-EPIfy
       EPIfz=-EPIfz
      END IF
*
      CALL ROTATE(EPIIfx,EPIIfy,EPIIfz,EPIfx,EPIfy,EPIfz,RITAG,
     . Nfx,Nfy,Nfz)
*
* CALCULATE THE ANGLE BETWEEN PRINCIPAL DIRECTIONS OF PINION AND CUTTER
*
c ----- cross product of e1i and epi
      SNSIGM=(E1Ify*EPIfz-E1Ifz*EPIfy)/Nfx
c ----- dot product of e1i and epi
      CSSIGM=E1Ifx*EPIfx+E1Ify*EPIfy+E1Ifz*EPIfz
      CS2SIG=2.D00*CSSIGM*CSSIGM-1.D00
      TN2SIG=2.D00*SNSIGM*CSSIGM/CS2SIG
*
* CALCULATE PRINCIPAL CURVATURES OF PINION CUTTER
```

```
*
c ----- eq. (2.20)
      KPII=1.D00/R
      IF(J .EQ. 2)THEN
      KPII=-KPII
      END IF
c ----- eq. (3.94)
      KPI=(KPII*(K1I*CSSIGM*CSSIGM+K1II*SNSIGM*SNSIGM)-K1I*K1II)/
     . (KPII-K1I*SNSIGM*SNSIGM-K1II*CSSIGM*CSSIGM)
*
*CALCULATE A11, A12, AND A22
*
c ----- eq. (A.31)
      A11=KPI-K1I*CSSIGM*CSSIGM-K1II*SNSIGM*SNSIGM
c ----- eq. (A.32)
      A12=(K1I-K1II)*SNSIGM*CSSIGM
c ----- eq. (A.34)
      A22=KPII-K1I*SNSIGM*SNSIGM-K1II*CSSIGM*CSSIGM
*
* CALCULATE ZCR
*
c ----- eq. (3.101)
      IF(J .EQ. 1)THEN
       ZCR=(SNLANP/KPI)-R*SNLANP
      ELSE
       ZCR=-(SNLANP/KPI)-R*SNLANP
      END IF
*
* CALCULATE XCR
*
c ----- eq. (3.99)
      Bmx=-Bfx*CSD1+Bfz*SND1
      XCR=Bmx-R*CSLANP
*
* CALCULATE RP
*
c ----- eq. (3.103)
      IF(I*J .EQ. 2)THEN
       RP=ZCR+DSQRT(DABS(R*R-XCR*XCR))
      ELSE
       RP=ZCR-DSQRT(DABS(R*R-XCR*XCR))
      END IF
*
* CALCULATE MCP
*
      Z11=Nfy*EPIfz-Nfz*EPIfy
      Z12=Nfy*EPIfx-Nfx*EPIfy
      Z21=Nfy*EPIIfz-Nfz*EPIIfy
      Z22=Nfy*EPIIfx-Nfx*EPIIfy
c ----- eqs. (3.107), (3.108)
      C11=Z11*CSD1+Z12*SND1
      C12=-Z11*SNPIT1+Z12*CSPIT1
      C22=-Z21*SNPIT1+Z22*CSPIT1
```

```
       IF(HG .EQ. 'R')THEN
        C11=-C11
        C12=-C12
        C22=-C22
       END IF
c ----- eq. (3.119)
       T4=(Bfy*CSRT1)/(EPIIfx*CSD1-EPIIfz*SND1)
       IF(HG .EQ. 'R')THEN
        T4=-T4
       END IF
c ----- eq. (3.120)
       T1=-C11/KPI
       T2=(A11*KPII*T4+A11*C22-A12*C12)/(A12*KPI)
c ----- eq. (3.122)
       U11=T1*EPIfx
       U12=T2*EPIfx+T4*EPIIfx
       U21=T1*EPIfy
       U22=T2*EPIfy+T4*EPIIfy
       U31=T1*EPIfz
       U32=T2*EPIfz+T4*EPIIfz
c ----- eq. (3.124)
       V1=U21*(Nfz*CSD1+Nfx*SND1)-Nfy*(U11*SND1+U31*CSD1)
c ----- eq. (3.125)
       V2=(U22*CSD1-U21*SNPIT1)*Nfz-(U11*CSPIT1+U12*SND1+U32*CSD1-U31
     .    *SNPIT1)*Nfy+(U21*CSPIT1+U22*SND1)*Nfx
c ----- eq. (3.126)
       V3=U22*CSPIT1*Nfx+(U32*SNPIT1-U12*CSPIT1)*Nfy-U22*SNPIT1*Nfz
       IF(HG .EQ. 'R')THEN
        V1=-V1
        V2=-V2
        V3=-V3
       END IF
c ----- eq. (3.132)
       H11=-U21*CSPIT1+SND1*(Bfz*SNPIT1-Bfx*CSPIT1)
c ----- eq. (3.134)
       H21=U11*CSPIT1-U31*SNPIT1+Bfy*SNRT1
c ----- eq. (3.136)
       H31=U21*SNPIT1+CSD1*(Bfz*SNPIT1-Bfx*CSPIT1)
c ----- eq. (3.133)
       H12=(Bfz*SNPIT1-Bfx*CSPIT1-U22)*CSPIT1
c ----- eq. (3.135)
       H22=-(Bfy-U12*CSPIT1+U32*SNPIT1)
c ----- eq. (3.137)
       H32=-(Bfz*SNPIT1-Bfx*CSPIT1-U22)*SNPIT1
       IF(HG .EQ. 'R')THEN
        H11=U21*CSPIT1+SND1*(Bfz*SNPIT1-Bfx*CSPIT1)
        H21=-U11*CSPIT1+U31*SNPIT1+Bfy*SNRT1
        H31=-U21*SNPIT1+CSD1*(Bfz*SNPIT1-Bfx*CSPIT1)
        H12=(Bfz*SNPIT1-Bfx*CSPIT1+U22)*CSPIT1
        H22=-(Bfy+U12*CSPIT1-U32*SNPIT1)
        H32=-(Bfz*SNPIT1-Bfx*CSPIT1+U22)*SNPIT1
       END IF
c ----- eq. (3.139)
```

```
      F1=Nfx*H11+Nfy*H21+Nfz*H31
c ----- eq. (3.140)
      F2=Nfx*H12+Nfy*H22+Nfz*H32
c ----- eq. (3.145)
      Y2=A12*(2.D00*KPI*T1*T2-V2-F1)
      Y3=A12*(KPI*T2*T2+KPII*T4*T4-V3-F2)-(KPI*T2+C12)*(KPII*T4+C22)
      MP1=-Y3/Y2
*
* CALCULATE EM AND LM
*
c ----- eq. (3.122)
      VT1Pfx=U11*MP1+U12
      VT1Pfy=U21*MP1+U22
      VT1Pfz=U31*MP1+U32
c ----- eq. (3.111)
      IF(HG .EQ. 'L')THEN
       EM=(Bfy*CSPIT1-VT1Pfx)/(MP1*SND1)+Bfy
       LM=(Bfx*CSPIT1-Bfz*SNPIT1+VT1Pfy)/MP1+Bfx*SND1+Bfz*CSD1
      ELSE
       EM=(-Bfy*CSPIT1-VT1Pfx)/(MP1*SND1)-Bfy
       LM=(Bfx*CSPIT1-Bfz*SNPIT1-VT1Pfy)/MP1+Bfx*SND1+Bfz*CSD1
      END IF
*
* CALCULATE SP AND QP
*
c ----- eqs. (3.150), (3.151)
      IF(HG .EQ. 'L')THEN
       IF(J .EQ. 1)THEN
        Z1=-Bfy+EM-SNLANP/KPI*SNTAUP
        Z2=Bfx*SND1+Bfz*CSD1-LM-SNLANP/KPI*CSTAUP
       ELSE
        Z1=-Bfy+EM+SNLANP/KPI*SNTAUP
        Z2=Bfx*SND1+Bfz*CSD1-LM+SNLANP/KPI*CSTAUP
       END IF
      ELSE
       IF(J .EQ. 1)THEN
        Z1=Bfy+EM+SNLANP/KPI*SNTAUP
        Z2=Bfx*SND1+Bfz*CSD1-LM-SNLANP/KPI*CSTAUP
       ELSE
        Z1=Bfy+EM-SNLANP/KPI*SNTAUP
        Z2=Bfx*SND1+Bfz*CSD1-LM+SNLANP/KPI*CSTAUP
       END IF
      END IF
      SP=DSQRT(Z1*Z1+Z2*Z2)
      QP=DATAN(Z1/Z2)
      IF(HG .EQ. 'L')THEN
       THETAP=TAUP-QP
      ELSE
       THETAP=TAUP+QP
      END IF
*
* CONVERT RADIAN TO DEGREE
*
```

```
      PSIPDG=PSIP/CNST
      TAUPDG=TAUP/CNST
      QPDG=QP/CNST
      THEPDG=THETAP/CNST
      LANPDG=LANDAP/CNST
*
* OUTPUT
*
      WRITE(72,10001)PSIPDG,QPDG,RP,SP,MP1,LANPDG,XCR,ZCR,EM,LM,TAUPDG,
     .                THEPDG
10000 FORMAT(1X,'GEAR SETTINGS:',/
     .       ,1X,'PSIGDG    =',G20.12,12X,'QGDG      =',G20.12,/
     .       ,1X,'RG        =',G20.12,12X,'SG        =',G20.12,/
     .       ,1X,'MG2       =',G20.12,12X,'TAUGDG    =',G20.12,/
     .       ,1X,'UG        =',G20.12,12X,'THETAGDG =',G20.12,/
     .       ,1X,'PHIGODG  =',G20.12,//
     .       ,1X,'PINION SETTINGS:',/)
10001 FORMAT(1X,'PSIPDG    =',G20.12,12X,'QPDG      =',G20.12,/
     .       ,1X,'RP        =',G20.12,12X,'SP        =',G20.12,/
     .       ,1X,'MP1       =',G20.12,12X,'LANDAPDG =',G20.12,/
     .       ,1X,'XCR       =',G20.12,12X,'ZCR       =',G20.12,/
     .       ,1X,'EM        =',G20.12,12X,'LM        =',G20.12,/
     .       ,1X,'TAUPDG    =',G20.12,12X,'THETAPDG =',G20.12,/)
*
* TCA
*
      TPAR(1)=RG*CSPSIG/SNPSIG*CSPSIG
      TPAR(2)=(MG2-SNRT2)*CSPSIG
      TPAR(3)=CSRT2*SNPSIG
      TPAR(4)=RG*CSPSIG/SNPSIG
      TPAR(5)=CSD2*SNPSIG
      TPAR(6)=SND2*CSPSIG
      TPAR(7)=SND2*SNPSIG
      TPAR(8)=CSD2*CSPSIG
      TPAR(9)=ZCR*CSRT1
      TPAR(10)=SP*CSRT1
      TPAR(11)=EM*CSRT1
      TPAR(12)=XCR*CSRT1
      TPAR(13)=SP*(MP1-SNRT1)
      TPAR(14)=EM*SNRT1
      TPAR(15)=LM*SNRT1
      TPAR(16)=J
      TPAR(17)=R
*
      PHIP=0.D00
      PHI21=0.D00
      PHI11=0.D00
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
*
      TX(1)=PHIP
      TX(2)=THETAP
      TX(3)=LANDAP
```

```
      TX(4)=PHI21
      TX(5)=PHIG
      TX(6)=THETAG
      CALL NONLIN(TCN,14,6,200,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
      PHIP0=TX(1)
      THEP0=TX(2)
      LANDP0=TX(3)
      PHI210=TX(4)
      PHIG0=TX(5)
      THEG0=TX(6)
*
      TX(1)=PHIP0
      TX(2)=THEP0
      TX(3)=LANDP0
      TX(4)=PHI210
      TX(5)=PHIG0
      TX(6)=THEG0
      DPHI11=18.D00/36.D00*CNST
*
      DO 100 IJ=1,60
      CSPH11=DCOS(PHI11)
      SNPH11=DSIN(PHI11)
      CALL NONLIN(TCN,14,6,200,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
      PHIP=TX(1)
      THETAP=TX(2)
      LANDAP=TX(3)
      PHI21=TX(4)
      PHIG=TX(5)
      THETAG=TX(6)
      ERROR=((PHI21*36.D02-PHI210*36.D02)-PHI11*36.D02*TN1/TN2)/CNST
*
      CALL PRING2(KS2,G2,E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz)
      CALL PRINP1(KS1,G1,E1Ifx,E1Ify,E1Ifz,E1IIfx,E1IIfy,E1IIfz)
      CALL SIGAN2(E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz,E1Ifx,E1Ify,
     .            E1Ifz,CS2SIG,SN2SIG,SIGM12)
      CALL EULER(KS2,G2,KS1,G1,CS2SIG,SN2SIG,IEU)
      IF(IEU .EQ. 1)THEN
       WRITE(72,*)'THERE IS INTERFERENCE'
       GO TO 88888
      END IF
*
      CALL ELLIPS(KS2,G2,KS1,G1,CS2SIG,SN2SIG,DEF,ALFA1,
     .            AXISL,AXISS,E1Ifx,E1Ify,E1Ifz)
*
      CALL PF(B2px,B2py,B2pz,B2fx,B2fy,B2fz)
*
* XBf, YBf, and ZBf is the direction of the long axis of the ellipse
*
      CALL PF(XBp,YBp,ZBp,XBf,YBf,ZBf)
      ELB1px=B2px+XBp
      ELB1pz=B2pz+ZBp
      ELB2px=B2px-XBp
      ELB2pz=B2pz-ZBp
```

```
*
        IF(I .EQ. 1 .AND. J .EQ. 1)THEN
         WRITE(9,9000)IJ,PHI11/CNST,IJ,ERROR
         IF(IJ .LE. 37)THEN
          WRITE(8,8000)IJ,B2pz,IJ,B2px
          WRITE(7,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
         END IF
        ELSE IF(I .EQ. 1 .AND. J .EQ. 2)THEN
         WRITE(79,9000)IJ,PHI11/CNST,IJ,ERROR
         IF(IJ .LE. 37)THEN
          WRITE(78,8000)IJ,B2pz,IJ,B2px
          WRITE(77,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
         END IF
        ELSE IF(I .EQ. 2 .AND. J .EQ. 1)THEN
         WRITE(29,9000)IJ,PHI11/CNST,IJ,ERROR
         IF(IJ .LE. 37)THEN
          WRITE(28,8000)IJ,B2pz,IJ,B2px
          WRITE(27,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
         END IF
        ELSE
         WRITE(89,9000)IJ,PHI11/CNST,IJ,ERROR
         IF(IJ .LE. 37)THEN
          WRITE(88,8000)IJ,B2pz,IJ,B2px
          WRITE(87,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
         END IF
        END IF
*
        PHI11=PHI11+DPHI11
*
100     CONTINUE
*
*
*
        PHI11=0.D00
*
        TX(1)=PHIP0
        TX(2)=THEP0
        TX(3)=LANDP0
        TX(4)=PHI210
        TX(5)=PHIG0
        TX(6)=THEG0
*
        DO 200 IJ=1,60
        CSPH11=DCOS(PHI11)
        SNPH11=DSIN(PHI11)
        CALL NONLIN(TCN,14,6,200,TX,TF,TF1,1.D-5,AZS,IPVT,WORK)
        PHIP=TX(1)
        THETAP=TX(2)
        LANDAP=TX(3)
        PHI21=TX(4)
        PHIG=TX(5)
        THETAG=TX(6)
        ERROR=((PHI21*36.D02-PHI210*36.D02)-PHI11*36.D02*TN1/TN2)/CNST
```

```
*
      CALL PRING2(KS2,G2,E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz)
      CALL PRINP1(KS1,G1,E1Ifx,E1Ify,E1Ifz,E1IIfx,E1IIfy,E1IIfz)
      CALL SIGAN2(E2Ifx,E2Ify,E2Ifz,E2IIfx,E2IIfy,E2IIfz,E1Ifx,E1Ify,
     .            E1Ifz,CS2SIG,SN2SIG,SIGM12)
      CALL EULER(KS2,G2,KS1,G1,CS2SIG,SN2SIG,IEU)
      IF(IEU .EQ. 1)THEN
       WRITE(72,*)'THERE IS INTERFERENCE'
       GO TO 88888
      END IF
*
      CALL ELLIPS(KS2,G2,KS1,G1,CS2SIG,SN2SIG,DEF,ALFA1,
     .            AXISL,AXISS,E1Ifx,E1Ify,E1Ifz)
*
      CALL PF(B2px,B2py,B2pz,B2fx,B2fy,B2fz)
*
* XBf, YBf, and ZBf is the direction of the long axis of the ellipse
*
      CALL PF(XBp,YBp,ZBp,XBf,YBf,ZBf)
      ELB1px=B2px+XBp
      ELB1pz=B2pz+ZBp
      ELB2px=B2px-XBp
      ELB2pz=B2pz-ZBp
*
      IF(I .EQ. 1 .AND. J .EQ. 1)THEN
       WRITE(9,9001)IJ,PHI11/CNST,IJ,ERROR
       IF(IJ .LE. 37)THEN
        WRITE(8,8001)IJ,B2pz,IJ,B2px
        WRITE(7,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
       END IF
      ELSE IF(I .EQ. 1 .AND. J .EQ. 2)THEN
       WRITE(79,9001)IJ,PHI11/CNST,IJ,ERROR
       IF(IJ .LE. 37)THEN
        WRITE(78,8001)IJ,B2pz,IJ,B2px
        WRITE(77,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
       END IF
      ELSE IF(I .EQ. 2 .AND. J .EQ. 1)THEN
       WRITE(29,9001)IJ,PHI11/CNST,IJ,ERROR
       IF(IJ .LE. 37)THEN
        WRITE(28,8001)IJ,B2pz,IJ,B2px
        WRITE(27,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
       END IF
      ELSE
       WRITE(89,9001)IJ,PHI11/CNST,IJ,ERROR
       IF(IJ .LE. 37)THEN
        WRITE(88,8001)IJ,B2pz,IJ,B2px
        WRITE(87,7000)ELB1pz,ELB1px,ELB2pz,ELB2px
       END IF
      END IF
*
      PHI11=PHI11-DPHI11
*
200   CONTINUE
```

```
*
99999 CONTINUE
88888 CONTINUE
*
7000  FORMAT(6X,'EX(1)=',F9.6,/,6X,'EY(1)=',F9.6,/,
     .       6X,'EX(2)=',F9.6,/,6X,'EY(2)=',F9.6,/,
     .       6X,'CALL CURVE(EX,EY,2,0)')
8000  FORMAT(6X,'X0(',I2,')=',F9.6,/,6X,'Y0(',I2,')=',F15.6)
8001  FORMAT(6X,'X1(',I2,')=',F9.6,/,6X,'Y1(',I2,')=',F15.6)
9000  FORMAT(6X,'X0(',I2,')=',F7.3,/,6X,'Y0(',I2,')=',F16.4)
9001  FORMAT(6X,'X1(',I2,')=',F7.3,/,6X,'Y1(',I2,')=',F16.4)
      END
*
* FOR THE DETERMINATION OF MEAN CONTACT POINT
*
      SUBROUTINE PCN1(X,F,NE)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CHARACTER*8 HG
      INTEGER NE
      REAL*8 X(NE),F(NE),PAR(6)
      COMMON/P1/PAR
      COMMON/A0/HG
      COMMON/A1/p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      THETAG=X(1)
      CSTHEG=DCOS(THETAG)
      SNTHEG=DSIN(THETAG)
      IF(HG .EQ. 'L')THEN
       UG=PAR(1)-SG*(PAR(2)*SNTHEG+PAR(3))/(CSRT2*DSIN(THETAG-QG))
      ELSE
       UG=PAR(1)-SG*(PAR(2)*SNTHEG+PAR(3))/(CSRT2*DSIN(THETAG+QG))
      END IF
      Bcx=PAR(4)-UG*CSPSIG
      Bcy=UG*SNPSIG*SNTHEG
      Bcz=UG*SNPSIG*CSTHEG
      CALL TRCOOR(Bpx,Bpy,Bpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . Bcx,Bcy,Bcz)
      XM=Bpz*(TND1-TND2)/2.D00
      F(1)=Bpx-XM
      END
*
* FOR THE DETERMINATION OF MEAN CONTACT POINT
*
      SUBROUTINE PCN2(X,F,NE)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CHARACTER*8 HG
      INTEGER NE
      REAL*8 X(NE),F(NE),PAR(6)
      COMMON/P1/PAR
      COMMON/A0/HG
      COMMON/A1/p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3
```

```
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/A5/CSQG,SNQG,THETAG
      PHIG=X(1)
      CSPHIG=DCOS(PHIG)
      SNPHIG=DSIN(PHIG)
      IF(HG .EQ. 'L')THEN
       UG=PAR(1)-SG*(PAR(2)+PAR(3)*DSIN(QG-PHIG))/
     .     (CSRT2*DSIN(THETAG-QG+PHIG))
      ELSE
       UG=PAR(1)-SG*(PAR(2)+PAR(3)*DSIN(QG-PHIG))/
     .     (CSRT2*DSIN(THETAG+QG-PHIG))
      END IF
      Bcx=PAR(4)-UG*CSPSIG
      Bcy=UG*PAR(5)
      Bcz=UG*PAR(6)
*
* Mmc=Mms*Msc
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .  1.D00,0.D00,0.D00,0.D00,CSPHIG,SNPHIG,0.D00,-SNPHIG,CSPHIG,
     .  0.D00,0.D00,0.D00,
     .  1.D00,0.D00,0.D00,0.D00,CSQG,-SNQG,0.D00,SNQG,CSQG,
     .  0.D00,-SG*SNQG,SG*CSQG)
*
* Mpc=Mpm*Mmc
*
       CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     .  CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .  0.D00,0.D00,0.D00,
     .  m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
      ELSE
*
* Mmc=Mms*Msc
*
       CALL COMBI(m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3,
     .  1.D00,0.D00,0.D00,0.D00,CSPHIG,-SNPHIG,0.D00,SNPHIG,CSPHIG,
     .  0.D00,0.D00,0.D00,
     .  1.D00,0.D00,0.D00,0.D00,CSQG,SNQG,0.D00,-SNQG,CSQG,
     .  0.D00,SG*SNQG,SG*CSQG)
*
* Mpc=Mpm*Mmc
*
       CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     .  CSD2,0.D00,-SND2,0.D00,1.D00,0.D00,SND2,0.D00,CSD2,
     .  0.D00,0.D00,0.D00,
     .  m11,m12,m13,m21,m22,m23,m31,m32,m33,m1,m2,m3)
      END IF
      CALL TRCOOR(Bpx,Bpy,Bpz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . Bcx,Bcy,Bcz)
```

```
      XM=Bpz*(TND1-TND2)/2.D00
      F(1)=Bpx-XM
      RETURN
      END
*
* FOR THE DETERMINATION OF COORDINATES AND NORMALS OF CONTACT POINTS
*
      SUBROUTINE TCN(TX,TF,NE)
      IMPLICIT REAL*8(A-H,K,M-Z)
      REAL*8 LANDAP,LM
      CHARACTER*8 HG
      INTEGER NE
      DIMENSION TX(NE),TF(NE),TPAR(19)
      COMMON/T1/TPAR
      COMMON/A0/HG
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSLANP,SNLANP
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/B2fx,B2fy,B2fz
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
      COMMON/B5/XCR,ZCR
      COMMON/C1/UG,CSTAUG,SNTAUG
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/D1/CSTAUP,SNTAUP
      COMMON/F1/PHIGO
      COMMON/G1/DA,DV
      J=IDINT(TPAR(16))
      PHIP=TX(1)
      THETAP=TX(2)
      LANDAP=TX(3)
      PHI21=TX(4)
      PHIG=TX(5)
      THETAG=TX(6)
      CSPHIP=DCOS(PHIP)
      SNPHIP=DSIN(PHIP)
      CSTHEP=DCOS(THETAP)
      SNTHEP=DSIN(THETAP)
      CSLANP=DCOS(LANDAP)
      SNLANP=DSIN(LANDAP)
      CSPH21=DCOS(PHI21)
      SNPH21=DSIN(PHI21)
      CSPHIG=DCOS(PHIG)
      SNPHIG=DSIN(PHIG)
      CSTHEG=DCOS(THETAG)
      SNTHEG=DSIN(THETAG)
      PHI2=(PHIG-PHIGO)/MG2
      PHI1=PHIP/MP1
      CSPH2=DCOS(PHI2)
      SNPH2=DSIN(PHI2)
      CSPH1=DCOS(PHI1)
      SNPH1=DSIN(PHI1)
```

```
      IF(HG .EQ. 'L')THEN
       TAUP=THETAP+QP-PHIP
      ELSE
       TAUP=THETAP-QP+PHIP
      END IF

      CSTAUP=DCOS(TAUP)
      SNTAUP=DSIN(TAUP)

      IF(HG .EQ. 'L')THEN
       TAUG=THETAG-QG+PHIG
      ELSE
       TAUG=THETAG+QG-PHIG
      END IF

      CSTAUG=DCOS(TAUG)
      SNTAUG=DSIN(TAUG)
      CSQPHP=DCOS(QP-PHIP)
      SNQPHP=DSIN(QP-PHIP)
      CSQPHG=DCOS(QG-PHIG)
      SNQPHG=DSIN(QG-PHIG)
*
* LEFT-HAND GEAR
*
      IF(HG .EQ. 'L')THEN
       UG=TPAR(1)-SG*(TPAR(2)*SNTHEG-SNQPHG*TPAR(3))/(CSRT2*SNTAUG)
       B2py=UG*SNPSIG*SNTAUG-SG*SNQPHG
      ELSE
       UG=TPAR(1)-SG*(TPAR(2)*SNTHEG+SNQPHG*TPAR(3))/(CSRT2*SNTAUG)
       B2py=UG*SNPSIG*SNTAUG+SG*SNQPHG
      END IF
      B2px=CSD2*(TPAR(4)-UG*CSPSIG)-SND2*(UG*SNPSIG*CSTAUG+SG*CSQPHG)
      B2pz=SND2*(TPAR(4)-UG*CSPSIG)+CSD2*(UG*SNPSIG*CSTAUG+SG*CSQPHG)
      N2px=TPAR(5)-TPAR(6)*CSTAUG
      N2py=CSPSIG*SNTAUG
      N2pz=TPAR(7)+TPAR(8)*CSTAUG
*
*  [Mwp]=[Mwa][Map]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .  wp1,wp2,wp3,
     .  CSPH2,SNPH2,0.D00,-SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .            wp1,wp2,wp3,
     .  CSPH2,-SNPH2,0.D00,SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00)
```

```
      END IF
      CALL TRCOOR(B2wx,B2wy,B2wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,wp1,wp2,wp3,
     . B2px,B2py,B2pz)
      CALL TRCOOR(N2wx,N2wy,N2wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,0.D00,0.D00,0.D00,
     . N2px,N2py,N2pz)
*
*  [Mfw]=[Mfa][Maw]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,
     .  fw1,fw2,fw3,
     .  CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00,
     .  CSPH21,-SNPH21,0.D00,SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,
     .  fw1,fw2,fw3,
     .  CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     .  0.D00,0.D00,0.D00,
     .  CSPH21,SNPH21,0.D00,-SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B2fx,B2fy,B2fz,
     . fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,fw1,fw2,fw3,
     . B2wx,B2wy,B2wz)
      CALL TRCOOR(N2fx,N2fy,N2fz,
     . fw11,fw12,fw13,fw21,fw22,fw23,fw31,fw32,fw33,0.D00,0.D00,0.D00,
     . N2wx,N2wy,N2wz)
*
* PINION
*
      IF(HG .EQ. 'L')THEN
       B1py=(ZCR+TPAR(17)*SNLANP)*SNTAUP+SP*SNQPHP-EM
      ELSE
       B1py=(ZCR+TPAR(17)*SNLANP)*SNTAUP-SP*SNQPHP+EM
      END IF
      B1px=(XCR+TPAR(17)*CSLANP)*CSD1-((ZCR+TPAR(17)*SNLANP)*CSTAUP+
     .       SP*CSQPHP+LM)*SND1
      B1pz=(XCR+TPAR(17)*CSLANP)*SND1+((ZCR+TPAR(17)*SNLANP)*CSTAUP+
     .       SP*CSQPHP+LM)*CSD1
      IF(J .EQ. 2)THEN
       N1px=CSLANP*CSD1-SNLANP*SND1*CSTAUP
       N1py=SNLANP*SNTAUP
       N1pz=CSLANP*SND1+SNLANP*CSD1*CSTAUP
      ELSE
       N1px=-CSLANP*CSD1+SNLANP*SND1*CSTAUP
       N1py=-SNLANP*SNTAUP
       N1pz=-CSLANP*SND1-SNLANP*CSD1*CSTAUP
      END IF
*
```

```
*  [Mwp]=[Mwa][Map]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .             wp1,wp2,wp3,
     .  CSPH1,-SNPH1,0.D00,SNPH1,CSPH1,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT1,0.D00,SNPIT1,0.D00,1.D00,0.D00,-SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,
     .             wp1,wp2,wp3,
     .  CSPH1,SNPH1,0.D00,-SNPH1,CSPH1,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00,
     .  CSPIT1,0.D00,SNPIT1,0.D00,1.D00,0.D00,-SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B1wx,B1wy,B1wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,wp1,wp2,wp3,
     . B1px,B1py,B1pz)
      CALL TRCOOR(N1wx,N1wy,N1wz,
     . wp11,wp12,wp13,wp21,wp22,wp23,wp31,wp32,wp33,0.D00,0.D00,0.D00,
     . N1px,N1py,N1pz)
*
*  [Mpw]=[Mpa][Maw]
*
      IF(HG .EQ. 'L')THEN
       CALL COMBI(pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,
     .  pw1,pw2,pw3,
     .  CSPIT1,0.D00,-SNPIT1,0.D00,1.D00,0.D00,SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00,
     .  CSPH11,SNPH11,0.D00,-SNPH11,CSPH11,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      ELSE
       CALL COMBI(pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,
     .  pw1,pw2,pw3,
     .  CSPIT1,0.D00,-SNPIT1,0.D00,1.D00,0.D00,SNPIT1,0.D00,CSPIT1,
     .  0.D00,0.D00,0.D00,
     .  CSPH11,-SNPH11,0.D00,SNPH11,CSPH11,0.D00,0.D00,0.D00,1.D00,
     .  0.D00,0.D00,0.D00)
      END IF
      CALL TRCOOR(B1px,B1py,B1pz,
     . pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,pw1,pw2,pw3,
     . B1wx,B1wy,B1wz)
      CALL TRCOOR(N1px,N1py,N1pz,
     . pw11,pw12,pw13,pw21,pw22,pw23,pw31,pw32,pw33,0.D00,0.D00,0.D00,
     . N1wx,N1wy,N1wz)
      B1fx=-B1px+DA*SNPIT1
      B1fy=-B1py+DV
      B1fz=B1pz+DA*CSPIT1
      N1fx=-N1px
      N1fy=-N1py
      N1fz=N1pz
```

```
      IF(HG .EQ. 'L')THEN
       TF(1)=(TPAR(9)*SNTAUP+TPAR(10)*SNQPHP-TPAR(11))*CSLANP-
     .       (TPAR(12)*SNTAUP-TPAR(13)*SNTHEP+TPAR(14)*CSTAUP+TPAR(15)*
     .        SNTAUP)*SNLANP
      ELSE
       TF(1)=(TPAR(9)*SNTAUP-TPAR(10)*SNQPHP+TPAR(11))*CSLANP-
     .       (TPAR(12)*SNTAUP-TPAR(13)*SNTHEP-TPAR(14)*CSTAUP+TPAR(15)*
     .        SNTAUP)*SNLANP
      END IF
      TF(2)=B2fx-B1fx
      TF(3)=B2fy-B1fy
      TF(4)=B2fz-B1fz
      TF(5)=N2fx-N1fx
      TF(6)=N2fy-N1fy
      END
*
* FOR THE DETERMINATION OF GEAR PRINCIPAL CURVATURES AND DIRECTIONS
*
      SUBROUTINE PRING2(KSG,GG,EGIfx,EGIfy,EGIfz,EGIIfx,EGIIfy,EGIIfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      COMMON/A2/SG,CSRT2,QG,SNPSIG,CSPSIG
      COMMON/A3/TND1,TND2,RITAG
      COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/B2fx,B2fy,B2fz
      COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
      COMMON/C1/UG,CSTAUG,SNTAUG
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/F1/PHIGO
      KCI=-CSPSIG/(UG*SNPSIG)
      KCII=0.D00
      ECIfx=SND2*SNTAUG
      ECIfy=CSTAUG
      ECIfz=-CSD2*SNTAUG
      ECIIfx=-CSD2*CSPSIG-SND2*SNPSIG*CSTAUG
      ECIIfy=SNPSIG*SNTAUG
      ECIIfz=-SND2*CSPSIG+CSD2*SNPSIG*CSTAUG
      WGfx=-SNPIT2
      WGfy=0.D00
      WGfz=CSPIT2
      WCfx=-MG2*CSD2
      WCfy=0.D00
      WCfz=-MG2*SND2
      WGCfx=WGfx-WCfx
      WGCfy=WGfy-WCfy
      WGCfz=WGfz-WCfz
      CALL CROSS(VTGfx,VTGfy,VTGfz,WGfx,WGfy,WGfz,B2fx,B2fy,B2fz)
      CALL CROSS(VTCfx,VTCfy,VTCfz,WCfx,WCfy,WCfz,B2fx,B2fy,B2fz)
      VTGCfx=VTGfx-VTCfx
      VTGCfy=VTGfy-VTCfy
      VTGCfz=VTGfz-VTCfz
      CALL DOT(VCI,ECIfx,ECIfy,ECIfz,VTGCfx,VTGCfy,VTGCfz)
      CALL DOT(VCII,ECIIfx,ECIIfy,ECIIfz,VTGCfx,VTGCfy,VTGCfz)
```

```
*
* CALCULATE A13,A23,A33
*
      CALL DET(DETI,WGCfx,WGCfy,WGCfz,N2fx,N2fy,N2fz,ECIfx,ECIfy,ECIfz)
      A13=-KCI*VCI-DETI
      CALL DET(DETII,WGCfx,WGCfy,WGCfz,N2fx,N2fy,N2fz,
     .         ECIIfx,ECIIfy,ECIIfz)
      A23=-KCII*VCII-DETII
      CALL DET(DET3,N2fx,N2fy,N2fz,WGCfx,WGCfy,WGCfz,
     .         VTGCfx,VTGCfy,VTGCfz)
      CALL CROSS(Cx,Cy,Cz,WGfx,0.D00,WGfz,VTCfx,VTCfy,VTCfz)
      CALL CROSS(Dx,Dy,Dz,WCfx,0.D00,WCfz,VTGfx,VTGfy,VTGfz)
      CALL DOT(DET45,N2fx,N2fy,N2fz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KCI*VCI*VCI+KCII*VCII*VCII-DET3-DET45
*
* CALCULATE SIGMA
*
      P=A23*A23-A13*A13+(KCI-KCII)*A33
      SIGMA2=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGMA2
*
* CALCULATE KGI AND KGII
*
      GG=P/(A33*DCOS(SIGMA2))
      KSG=KCI+KCII-(A13*A13+A23*A23)/A33
      KGI=(KSG+GG)/2.D00
      KGII=(KSG-GG)/2.D00
*
* CALCULATE EGI AND EGII
*
      CALL ROTATE(EGIfx,EGIfy,EGIfz,ECIfx,ECIfy,ECIfz,-SIGMA,N2fx,N2fy,
     . N2fz)
      CALL ROTATE(EGIIfx,EGIIfy,EGIIfz,EGIfx,EGIfy,EGIfz,RITAG,
     . N2fx,N2fy,N2fz)
      END
*
* FOR THE DETERMINATION OF PINION PRINCIPAL CURVATURES AND DIRECTIONS
*
      SUBROUTINE PRINP1(KSP,GP,EPIfx,EPIfy,EPIfz,EPIIfx,EPIIfy,EPIIfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      REAL*8 LM,TPAR(19)
      COMMON/T1/TPAR
      COMMON/A3/TND1,TND2,RITAG
      COMMON/B1/CSPH11,SNPH11,SP,EM,LM,CSRT1,CSD1,SND1,CSLANP,SNLANP
      COMMON/B2/CSPIT1,SNPIT1,MP1,MG2,QP
      COMMON/B3/B2fx,B2fy,B2fz
      COMMON/B5/XCR,ZCR
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/D1/CSTAUP,SNTAUP
      J=IDINT(TPAR(16))
      R=TPAR(17)
      IF(J .EQ. 1)THEN
      KCI=SNLANP/(ZCR+R*SNLANP)
```

```
      KCII=1.D00/R
      ELSE
      KCI=-SNLANP/(ZCR+R*SNLANP)
      KCII=-1.D00/R
      END IF

      ECIfx=SND1*SNTAUP
      ECIfy=CSTAUP
      ECIfz=CSD1*SNTAUP
      ECIIfx=CSD1*SNLANP+SND1*CSLANP*CSTAUP
      ECIIfy=-CSLANP*SNTAUP
      ECIIfz=-SND1*SNLANP+CSD1*CSLANP*CSTAUP
      IF(J .EQ. 2)THEN
      ECIIfx=-ECIIfx
      ECIIfy=-ECIIfy
      ECIIfz=-ECIIfz
      END IF

      WPfx=-SNPIT1
      WPfy=0.D00
      WPfz=-CSPIT1
      WCfx=-MP1*CSD1
      WCfy=0.D00
      WCfz=MP1*SND1
      WPCfx=WPfx-WCfx
      WPCfy=WPfy-WCfy
      WPCfz=WPfz-WCfz
      CALL CROSS(VTPfx,VTPfy,VTPfz,WPfx,WPfy,WPfz,B2fx,B2fy,B2fz)
      CALL CROSS(VTC1fx,VTC1fy,VTC1fz,WCfx,WCfy,WCfz,B2fx,B2fy,B2fz)
      CALL CROSS(VTC2fx,VTC2fy,VTC2fz,LM*SND1,EM,LM*CSD1,WCfx,WCfy,WCfz)
      VTCfx=VTC1fx+VTC2fx
      VTCfy=VTC1fy+VTC2fy
      VTCfz=VTC1fz+VTC2fz
      VTPCfx=VTPfx-VTCfx
      VTPCfy=VTPfy-VTCfy
      VTPCfz=VTPfz-VTCfz
      CALL DOT(VCI,ECIfx,ECIfy,ECIfz,VTPCfx,VTPCfy,VTPCfz)
      CALL DOT(VCII,ECIIfx,ECIIfy,ECIIfz,VTPCfx,VTPCfy,VTPCfz)
*
* CALCULATE A13,A23,A33
*
      CALL DET(DETI,WPCfx,WPCfy,WPCfz,N2fx,N2fy,N2fz,ECIfx,ECIfy,ECIfz)
      A13=-KCI*VCI-DETI
      CALL DET(DETII,WPCfx,WPCfy,WPCfz,N2fx,N2fy,N2fz,
     .         ECIIfx,ECIIfy,ECIIfz)
      A23=-KCII*VCII-DETII
      CALL DET(DET3,N2fx,N2fy,N2fz,WPCfx,WPCfy,WPCfz,
     .         VTPCfx,VTPCfy,VTPCfz)
      CALL CROSS(Cx,Cy,Cz,WPfx,0.D00,WPfz,VTCfx,VTCfy,VTCfz)
      CALL CROSS(Dx,Dy,Dz,WCfx,0.D00,WCfz,VTPfx,VTPfy,VTPfz)
      CALL DOT(DET45,N2fx,N2fy,N2fz,Cx-Dx,Cy-Dy,Cz-Dz)
      A33=KCI*VCI*VCI+KCII*VCII*VCII-DET3-DET45
*
```

```
* CALCULATE SIGMA
*
      P=A23*A23-A13*A13+(KCI-KCII)*A33
      SIGMA2=DATAN(2.D00*A13*A23/P)
      SIGMA=0.5D00*SIGMA2
*
* CALCULATE KPI AND KPII
*
      GP=P/(A33*DCOS(SIGMA2))
      KSP=KCI+KCII-(A13*A13+A23*A23)/A33
      KPI=(KSP+GP)/2.D00
      KPII=(KSP-GP)/2.D00
*
* CALCULATE EPI AND EPII
*
      CALL ROTATE(EPIfx,EPIfy,EPIfz,ECIfx,ECIfy,ECIfz,-SIGMA,N2fx,N2fy,
     . N2fz)
      CALL ROTATE(EPIIfx,EPIIfy,EPIIfz,EPIfx,EPIfy,EPIfz,RITAG,
     . N2fx,N2fy,N2fz)
      END
*
* FOR THE DETERMINATION OF THE ANGLE BETWEEN GEAR PRINCIPAL DIRECTIONS
* AND PINION PRINCIPAL DIRECTIONS
*
      SUBROUTINE SIGAN2(EGIfx,EGIfy,EGIfz,EGIIfx,EGIIfy,EGIIfz,EPIfx,
     . EPIfy,EPIfz,CS2SIG,SN2SIG,SIGMPG)
      IMPLICIT REAL*8(A-H,K,M-Z)
      CALL DOT(CSSIG,EPIfx,EPIfy,EPIfz,EGIfx,EGIfy,EGIfz)
      CALL DOT(SNSIG,EPIfx,EPIfy,EPIfz,-EGIIfx,-EGIIfy,-EGIIfz)
      SIGM2=4.D00*DATAN(SNSIG/(1.D00+CSSIG))
      SIGMPG=.5D00*SIGM2
      CS2SIG=DCOS(SIGM2)
      SN2SIG=DSIN(SIGM2)
      END
*
* FOR THE DETERMINATION OF CONTACT ELLIPS
*
      SUBROUTINE ELLIPS(KSG,GG,KSP,GP,CS2SIG,SN2SIG,DEF,ALFAP,
     .                    AXISL,AXISS,EPIfx,EPIfy,EPIfz)
      IMPLICIT REAL*8(A-H,K,M-Z)
      COMMON/A3/TND1,TND2,RITAG
      COMMON/C2/N2fx,N2fy,N2fz
      COMMON/E1/XBf,YBf,ZBf
      D=DSQRT(GP*GP-2.D00*GP*GG*CS2SIG+GG*GG)
      CS2AFP=(GP-GG*CS2SIG)/D
      SN2AFP=GG*SN2SIG/D
      ALFAP=DATAN(SN2AFP/(1.D00+CS2AFP))
      A=.25D00*DABS(KSP-KSG-D)
      B=.25D00*DABS(KSP-KSG+D)
      IF(KSG .LT. KSP)THEN
      AXISL=DSQRT(DEF/A)
      AXISS=DSQRT(DEF/B)
      CALL ROTATE(XBf,YBf,ZBf,EPIfx,EPIfy,EPIfz,RITAG-ALFAP,N2fx,
```

```
     . N2fy,N2fz)
       ELSE
       AXISL=DSQRT(DEF/B)
       AXISS=DSQRT(DEF/A)
       CALL ROTATE(XBf,YBf,ZBf,EPIfx,EPIfy,EPIfz,-ALFAP,N2fx,N2fy,
     . N2fz)
       END IF
       XBf=AXISL*XBf
       YBf=AXISL*YBf
       ZBf=AXISL*ZBf
       END
*
* COORDINATE TRANSFORMATION FOR F TO P
*
       SUBROUTINE PF(B2px,B2py,B2pz,B2fx,B2fy,B2fz)
       IMPLICIT REAL*8(A-H,K,M-Z)
       COMMON/A4/CSD2,SND2,CSPIT2,SNPIT2
       COMMON/B4/CSPH2,SNPH2,CSPH21,SNPH21
*
*  [Mtf]=[Mta][Maf]
*
       CALL COMBI(t11,t12,t13,t21,t22,t23,t31,t32,t33,t1,t2,t3,
     . CSPH21,SNPH21,0.D00,-SNPH21,CSPH21,0.D00,0.D00,0.D00,1.D00,
     . 0.D00,0.D00,0.D00,
     . CSPIT2,0.D00,SNPIT2,0.D00,1.D00,0.D00,-SNPIT2,0.D00,CSPIT2,
     . 0.D00,0.D00,0.D00)
       CALL TRCOOR(B2wx,B2wy,B2wz,
     . t11,t12,t13,t21,t22,t23,t31,t32,t33,t1,t2,t3,
     . B2fx,B2fy,B2fz)
*
*  [Mpt]=[Mpa][Mpt]
*
       CALL COMBI(p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . CSPIT2,0.D00,-SNPIT2,0.D00,1.D00,0.D00,SNPIT2,0.D00,CSPIT2,
     . 0.D00,0.D00,0.D00,
     . CSPH2,-SNPH2,0.D00,SNPH2,CSPH2,0.D00,0.D00,0.D00,1.D00,
     . 0.D00,0.D00,0.D00)
       CALL TRCOOR(B2px,B2py,B2pz,
     . p11,p12,p13,p21,p22,p23,p31,p32,p33,p1,p2,p3,
     . B2wx,B2wy,B2wz)
       END
*
* USING EULER FORMULA TO DETERMINATION SURFACE INTERFERENCE
*
       SUBROUTINE EULER(KSG,GG,KSP,GP,CS2SIG,SN2SIG,IEU)
       IMPLICIT REAL*8(A-H,K,M-Z)
       A=KSG-KSP
       B=DSQRT((GG-GP*CS2SIG)**2+(GP*SN2SIG)**2)
       KR1=(A+B)/2.D00
       KR2=(A-B)/2.D00
       IF(KR1*KR2 .LT. 0.D00)THEN
       IEU=1
       ELSE
```

```
      IEU=0
      END IF
      END
*
* DETERMINANT
*
      SUBROUTINE DET(S,A,B,C,D,E,F,G,H,P)
      IMPLICIT REAL*8(A-H,K,M-Z)
      S=A*E*P+D*H*C+G*B*F-A*H*F-D*B*P-G*E*C
      RETURN
      END
*
* COORDINATE TRANSFORMATION
*
      SUBROUTINE TRCOOR(XN,YN,ZN,R11,R12,R13,R21,R22,R23,R31,R32,R33,
     .                  T1,T2,T3,XP,YP,ZP)
      IMPLICIT REAL*8(A-H,O-Z)
      XN=R11*XP+R12*YP+R13*ZP+T1
      YN=R21*XP+R22*YP+R23*ZP+T2
      ZN=R31*XP+R32*YP+R33*ZP+T3
      RETURN
      END
*
* MULTIPLICATION OF TWO TRANSFORMATION MATRICES
*
      SUBROUTINE COMBI(C11,C12,C13,C21,C22,C23,C31,C32,C33,C1,C2,C3,
     .                 A11,A12,A13,A21,A22,A23,A31,A32,A33,A1,A2,A3,
     .                 B11,B12,B13,B21,B22,B23,B31,B32,B33,B1,B2,B3)
      IMPLICIT REAL*8(A-H,M,N,O-Z)
      C11=B31*A13+B21*A12+B11*A11
      C12=B32*A13+B22*A12+B12*A11
      C13=B33*A13+B23*A12+B13*A11
      C21=B31*A23+B21*A22+B11*A21
      C22=B32*A23+B22*A22+B12*A21
      C23=B33*A23+B23*A22+B13*A21
      C31=B31*A33+B21*A32+B11*A31
      C32=B32*A33+B22*A32+B12*A31
      C33=B33*A33+B23*A32+B13*A31
      C1=B3*A13+B2*A12+B1*A11+A1
      C2=B3*A23+B2*A22+B1*A21+A2
      C3=B3*A33+B2*A32+B1*A31+A3
      RETURN
      END
*
* DOT OF TWO VECTORS
*
      SUBROUTINE DOT(V,X1,Y1,Z1,X2,Y2,Z2)
      IMPLICIT REAL*8(A-H,O-Z)
      V=X1*X2+Y1*Y2+Z1*Z2
      RETURN
      END
*
* CROSS OF TWO VECTORS
```

```
*
      SUBROUTINE CROSS(X,Y,Z,A,B,C,D,E,F)
      IMPLICIT REAL*8(A-H,O-Z)
      X=B*F-C*E
      Y=C*D-A*F
      Z=A*E-B*D
      RETURN
      END
*
* ROTATION A VECTOR ABOUT ANOTHER VECTOR
*
      SUBROUTINE ROTATE(XN,YN,ZN,XP,YP,ZP,THETA,UX,UY,UZ)
      IMPLICIT REAL*8(A-H,O-Z)
      CT=DCOS(THETA)
      ST=DSIN(THETA)
      VT=1.D00-CT
      R11=UX*UX*VT+CT
      R12=UX*UY*VT-UZ*ST
      R13=UX*UZ*VT+UY*ST
      R21=UX*UY*VT+UZ*ST
      R22=UY*UY*VT+CT
      R23=UY*UZ*VT-UX*ST
      R31=UX*UZ*VT-UY*ST
      R32=UY*UZ*VT+UX*ST
      R33=UZ*UZ*VT+CT
      CALL TRCOOR(XN,YN,ZN,R11,R12,R13,R21,R22,R23,R31,R32,R33,
     .            0.D00,0.D00,0.D00,
     .            XP,YP,ZP)
      RETURN
      END
*
*     *****   SUBROUTINE NOLIN   *****
*
      SUBROUTINE NONLIN(FUNC,NSIG,NE,NC,X,Y,Y1,DELTA,A,IPVT,WORK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(NE),Y(NE),Y1(NE),A(NE,NE),IPVT(NE),WORK(NE)
      EXTERNAL FUNC
      NDIM=NE
      EPSI=1.D00/10.D00**NSIG
      CALL NONLIO(FUNC,EPSI,NE,NC,X,DELTA,NDIM,A,Y,Y1,WORK,IPVT)
      RETURN
      END
*
*     *****   SUBROUTINE NOLINO   *****
*
      SUBROUTINE NONLIO(FUNC,EPSI,NE,NC,X,DELTA,NDIM,A,Y,Y1,WORK,IPVT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(NE),Y(NE),Y1(NE),IPVT(NE),WORK(NE),A(NDIM,NE)
      EXTERNAL FUNC
* NC: # OF COUNT TIMES
      DO 5 I=1,NC
      CALL FUNC(X,Y,NE)
* NE: # OF EQUATIONS
```

```
      DO 15 J=1,NE
      IF (DABS(Y(J)).GT.EPSI) GO TO 25
   15 CONTINUE
      GO TO 105
   25 DO 35 J=1,NE
   35 Y1(J)=Y(J)
      DO 45 J=1,NE
      DIFF=DABS(X(J))*DELTA
      IF (DABS(X(J)).LT.1.D-12) DIFF=DELTA
      XMAM=X(J)
      X(J)=X(J)-DIFF
      CALL FUNC(X,Y,NE)
      X(J)=XMAM
      DO 55 K=1,NE
      A(K,J)=(Y1(K)-Y(K))/DIFF
   55 CONTINUE
   45 CONTINUE
      DO 65 J=1,NE
   65 Y(J)=-Y1(J)
      CALL DECOMP (NDIM,NE,A,COND,IPVT,WORK)
      CALL SOLVE (NDIM,NE,A,Y,IPVT)

      DO 75 J=1,NE
      X(J)=X(J)+Y(J)
   75 CONTINUE
    5 CONTINUE
  105 RETURN
      END
*
*     *****   SUBROUTINE DECOMP   *****
*
      SUBROUTINE DECOMP (NDIM,N,A,COND,IPVT,WORK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,N),WORK(N),IPVT(N)
*
*  DECOMPOSES A REAL MATRIX BY GAUSSIAN ELIMINATION,
*  AND ESTIMATES THE CONDITION OF THE MATRIX.
*
*  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
*       M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
*
*  USE SUBROUTINE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEM.
*
*  INPUT..
*
*     NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING   A
*     N    = ORDER OF THE MATRIX
*     A    = MATRIX TO BE TRIANGULARIZED
*
*  OUTPUT..
*
*     A   CONTAINS AN UPPER TRIANGULAR MATRIX  U  AND A PREMUTED
*        VERSION OF A LOWER TRIANGULAR MATRIX  I-L  SO THAT
```

```
*          (PERMUTATION MATRIX)*A=L*U
*
*       COND = AN ESTIMATE OF THE CONDITION OF  A.
*         FOR THE LINEAR SYSTEM  A*X = B , CHANGES IN   A  AND  B
*         MAY CAUSE CHANGES  COND  TIMES AS LARGE IN  X.
*         IF  COND+1.0 .EQ. COND , A IS SINGULAR TO WORKING
*         PRECISION.  COND  IS SET TO  1.0D+32  IF EXACT
*         SINGULARITY IS DETECTED.
*
*       IPVT      = THE PIVOT VECTOR
*         IPVT(K) = THE INDEX OF THE K-TH PIVOT ROW
*         IPVT(N) = (-1)**(NUMBER OF INTERCHANGES)
*
*   WORK SPACE..   THE VECTOR  WORK  MUST BE DECLARED AND INCLUDED
*         IN THE CALL.  ITS INPUT CONTENTS ARE IGNORED.
*         ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
*
*   THE DETERMINANT OF  A  CAN BE OBTAINED ON OUTPUT BY
*      DET(A) = IPVT(N) * A(1,1) * A(2,2) * ... * A(N,N)
*
      IPVT(N)=1
      IF (N.EQ.1) GO TO 150
      NM1=N-1
*                             COMPUTE THE 1-NORM OF  A .
      ANORM=0.D0
      DO 20 J=1,N
        T=0.D0
        DO 10 I=1,N
   10   T=T+DABS(A(I,J))
        IF (T.GT.ANORM) ANORM=T
   20 CONTINUE
*                             DO GAUSSIAN ELIMINATION WITH PARTIAL
*                                 PIVOTING.
      DO 70 K=1,NM1
        KP1=K+1
*                             FIND THE PIVOT.
        M=K
        DO 30 I=KP1,N
          IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
   30   CONTINUE
        IPVT(K)=M
        IF (M.NE.K) IPVT(N)=-IPVT(N)
        T=A(M,K)
        A(M,K)=A(K,K)
        A(K,K)=T
*                             SKIP THE ELIMINATION STEP IF PIVOT IS ZERO.
        IF (T.EQ.0.D0) GO TO 70
*
*                             COMPUTE THE MULTIPLIERS.
        DO 40 I=KP1,N
   40   A(I,K)=-A(I,K)/T
*                             INTERCHANGE AND ELIMINATE BY COLUMNS.
        DO 60 J=KP1,N
```

```
        T=A(M,J)
        A(M,J)=A(K,J)
        A(K,J)=T
        IF (T.EQ.0.D0) GO TO 60
        DO 50 I=KP1,N
   50   A(I,J)=A(I,J)+A(I,K)*T
   60 CONTINUE
   70 CONTINUE
*
*  COND = (1-NORM OF A)*(AN ESTIMATE OF THE 1-NORM OF A-INVERSE)
*  THE ESTIMATE IS OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
*  SMALL SINGULAR VECTOR.  THIS INVOLVES SOLVING TWO SYSTEMS
*  OF EQUATIONS,  (A-TRANSPOSE)*Y = E  AND  A*Z = Y  WHERE  E
*  IS A VECTOR OF +1 OR -1 COMPONENTS CHOSEN TO CAUSS GROWTH IN Y.
*  ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
*
*                             SOLVE  (A-TRANSPOSE)*Y = E .
      DO 100 K=1,N
        T=0.D0
        IF (K.EQ.1) GO TO 90
        KM1=K-1
        DO 80 I=1,KM1
   80   T=T+A(I,K)*WORK(I)

   90   EK=1.D0
        IF (T.LT.0.D0) EK=-1.D0

        IF (A(K,K).EQ.0.D0) GO TO 160
        A11=A(1,1)
      WORK(K)=-(EK+T)/A(1,1)
  100 CONTINUE
      DO 120 KB=1,NM1
        K=N-KB
        T=0.D0
        KP1=K+1
        DO 110 I=KP1,N
  110   T=T+A(I,K)*WORK(K)
        WORK(K)=T
        M=IPVT(K)
        IF (M.EQ.K) GO TO 120
        T=WORK(M)
        WORK(M)=WORK(K)
        WORK(K)=T
  120 CONTINUE
*
      YNORM=0.D0
      DO 130 I=1,N
  130 YNORM=YNORM+DABS(WORK(I))
*
*                             SOLVE  A*Z = Y
      CALL SOLVE (NDIM,N,A,WORK,IPVT)
*
```

```
      ZNORM=0.D0
      DO 140 I=1,N
  140 ZNORM=ZNORM+DABS(WORK(I))
*
*                              ESTIMATE THE CONDITION.
      COND=ANORM*ZNORM/YNORM
      IF (COND.LT.1.D0) COND=1.D0
      RETURN
*                              1-BY-1 CASE..
  150 COND=1.D0
      IF (A(1,1).NE.0.D0) RETURN
*
*                              EXACT SINGULARITY
  160 COND=1.0D32
      RETURN
      END
      SUBROUTINE SOLVE (NDIM,N,A,B,IPVT)
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,N),B(N),IPVT(N)
*
*  SOLVES A LINEAR SYSTEM,  A*X = B
*  DO NOT SOLVE THE SYSTEM IF DECOMP HAS DETECTED SINGULARITY.
*
*  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
*       M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
*
*  INPUT..
*
*     NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING  A
*     N    = ORDER OF MATRIX
*     A    = TRIANGULARIZED MATRIX OBTAINED FROM SUBROUTINE DECOMP
*     B    = RIGHT HAND SIDE VECTOR
*     IPVT = PIVOT VECTOR OBTAINED FROM DECOMP
*
*  OUTPUT..
*
*     B = SOLUTION VECTOR,  X
*
*                              DO THE FORWARD ELIMINATION.
      IF (N.EQ.1) GO TO 50
      NM1=N-1
      DO 20 K=1,NM1
        KP1=K+1
        M=IPVT(K)
        T=B(M)
        B(M)=B(K)
        B(K)=T
        DO 10 I=KP1,N
   10   B(I)=B(I)+A(I,K)*T
   20 CONTINUE
*                              NOW DO THE BACK SUBSTITUTION.
      DO 40 KB=1,NM1
```

```
      KM1=N-KB
      K=KM1+1
      B(K)=B(K)/A(K,K)
      T=-B(K)
      DO 30 I=1,KM1
30    B(I)=B(I)+A(I,K)*T
40 CONTINUE
50 B(1)=B(1)/A(1,1)
   RETURN
   END
```

Bibliography

[1] Litvin, F. L., Tsung, W.-J., and Lee, H.-T.: Generation of Spiral Bevel Gears with Conjugate Tooth Surfaces and Tooth Contact Analysis. NASA-CR-4088, Aug., 1987.

[2] Litvin, F. L. and Gutman, Y.: Methods of Synthesis and Analysis for Hypoid Gear Drives of "Formate" and "Helixform", Parts 1–3, Trans. of the ASME, Journal of Mechanical Design, Vol. 103, Jan. 1981, pp. 83-113.

[3] Litvin, F. L. and Gutman, Y.: A Method of Local Synthesis of Gears Grounded on the Connections Between the Principal and Geodetic Curvatures of Surfaces. Trans. of the ASME, Journal of Mechanical Design, Vol. 103, Jan. 1981, pp. 114-125.

[4] Zhang, Y.: Synthesis and Analysis of Hypoid Drive. Ph.D. dissertation, University of Illinois at Chicago, Chicago, 1988.

[5] Zhang, J.: Analysis and Synthesis of Modified Spur and Helical Gears. Ph.D. dissertation, University of Illinois at Chicago, Chicago, 1988.

[6] Dudley, D. W.: The Evolution of the Gear Art. Washington D. C., American Gear Manufacturers Association, 1969.

[7] Colvin, F. H. and Stanley, F. A.: Gear Cutting Practice. New York, McGraw-Hill, 1950.

[8] Dudley, D. W.: Bevel and Hypoid Gear Manufacture. In Gear Handbook: The Design, Manufacture, and Application of Gears, ed. D. W. Dudley, Chapter 20. New York, McGraw-Hill, 1962. New York, McGraw-Hill, 1950.

[9] Litvin, F. L.: The Theory of Gearing. Second Edition, Moscow, Nauka, 1968. (an English Edition is in Press by NASA)

[10] Dyson, A.: A General Theory of the Kinematics and Geometry of Gears in Gears in Three Dimensions. London, Oxford University Press, 1969.

[11] Gleason Works: Understanding Tooth Contact Analysis. Gleason Machine Division, The Gleason Works. Publication No. SD3139. New York, 1981.

[12] Litvin, F. L.: Relationships Between the Curvatures of Tooth Surfaces in Three-Dimensional Gear Systems. NASA-TM-75130, Jul., 1977.

[13] Boresi, A. P. and Sidebottom, O M.: Advanced Mechanics of Materials. Fourth Edition, New York, John Wiley & Sons, 1985.

[14] Hertz, H.: <u>Gesammetlte Werke.</u> Vol. 1, Lepzig, 1895. For an English translation, see Hertz, H.: <u>Miscellaneous Papers.</u> New York, Macmillan, 1896.

[15] Litvin, F. L., Rahman P., and Goldrich, N.: Mathematical Models for the Synthesis and Optimization of Spiral Bevel Gear Tooth Surfaces. NASA-CR-3553, Jun., 1982.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-4259 AVSCOM TR-89-C-014	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Generation and Tooth Contact Analysis of Spiral Bevel Gears With Predesigned Parabolic Functions of Transmission Errors		5. Report Date November 1989
		6. Performing Organization Code
7. Author(s) Faydor L. Litvin and Hong-Tao Lee		8. Performing Organization Report No. None (E-4977)
		10. Work Unit No. 1L162209A47A 505-63-51
9. Performing Organization Name and Address University of Illinois at Chicago Department of Mechanical Engineering Chicago, Illinois		11. Contract or Grant No. NAG3-783
12. Sponsoring Agency Name and Address Propulsion Directorate U.S. Army Aviation Research and Technology Activity—AVSCOM Cleveland, Ohio 44135-3127 and NASA Lewis Research Center Cleveland, Ohio 44135-3191		13. Type of Report and Period Covered Contractor Report Final
		14. Sponsoring Agency Code

15. Supplementary Notes

Project Manager, Robert F. Handschuh, Propulsion Directorate, U.S. Army Aviation Research and Technology Activity—AVSCOM.

16. Abstract

A new approach for determination of machine-tool settings for spiral bevel gears is proposed. The proposed settings provide a predesigned parabolic function of transmission errors and the desired location and orientation of the bearing contact. The predesigned parabolic function of transmission errors is able to absorb piece-wise linear functions of transmission errors that are caused by the gear misalignment and reduce gear noise. The gears are face-milled by head cutters with conical surfaces or surfaces of revolution. A computer program for simulation of meshing, bearing contact and determination of transmission errors for misaligned gear has been developed.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Gear geometry Spiral bevel gears Tooth contact analysis Gear	Unclassified – Unlimited Subject Category 37

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	215	A10

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1989