General Disclaimer

One or more of the Following Statements may affect this Document

• This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

• This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

• This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

• This document is paginated as submitted by the original source.

• Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
FINITE ELEMENT SOLUTION OF LUBRICATION PROBLEMS

by

M. M. Reddi

Prepared for
NASA-Lewis Research Center
Cleveland, Ohio 44135
FINITE ELEMENT SOLUTION
OF
LUBRICATION PROBLEMS

by
M. M. Reddi

Prepared for
NASA-Lewis Research Center
Cleveland, Ohio 44135
This report summarizes work done under Contract No. NAS3-13480, for NASA-Lewis Research Center, during the period 2-18-70 to 2-18-71. Mr. B. J. Hamrock acted as the Project Leader for NASA-Lewis.

The work was performed at the Franklin Institute Research Laboratories as Project No. 32G-C2762-01 through 06 with Mr. W. J. Shapiro acting as the Business Manager and Dr. M. M. Reddi as the Principal Investigator.
ABSTRACT

A variational formulation of the transient lubrication problem is presented and the corresponding finite element equations derived for three and six point triangles, and, four and eight point quadrilaterals. Test solutions for a one dimensional slider bearing used in validating the computer program 'FELUB' are given. Utility of the method is demonstrated by a solution of the shrouded step bearing.

Input map, input preparation instructions and program description of 'FELUB' computer program are presented.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td></td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>ii</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>2</td>
<td>FINITE ELEMENT FORMULATION</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>The Compressible Lubrication Problem</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Variational Formulation</td>
<td>2-3</td>
</tr>
<tr>
<td>2.3</td>
<td>Temporal Discretization</td>
<td>2-5</td>
</tr>
<tr>
<td>2.4</td>
<td>Spatial Discretization</td>
<td>2-7</td>
</tr>
<tr>
<td>2.5</td>
<td>Triangular Elements</td>
<td>2-10</td>
</tr>
<tr>
<td>2.6</td>
<td>Quadrilateral Elements</td>
<td>2-13</td>
</tr>
<tr>
<td>2.7</td>
<td>Boundary Conditions</td>
<td>2-17</td>
</tr>
<tr>
<td>2.8</td>
<td>Solution Scheme</td>
<td>2-18</td>
</tr>
<tr>
<td>3</td>
<td>PROGRAM DESCRIPTION</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>Major Program Variables</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3</td>
<td>Subroutine Descriptions</td>
<td>3-4</td>
</tr>
<tr>
<td>3.4</td>
<td>Input Preparation</td>
<td>3-12</td>
</tr>
<tr>
<td>3.5</td>
<td>Validation</td>
<td>3-25</td>
</tr>
<tr>
<td>3.6</td>
<td>Shrouded Step Bearing</td>
<td>3-29</td>
</tr>
<tr>
<td>4</td>
<td>CONCLUSIONS</td>
<td>4-1</td>
</tr>
<tr>
<td>5</td>
<td>REFERENCES</td>
<td>5-1</td>
</tr>
<tr>
<td>Fig. No.</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Triangular Elements</td>
<td>2-11</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Quadrilateral Elements</td>
<td>2-14</td>
</tr>
<tr>
<td>3.2.1</td>
<td>General Flow Chart for FELUB</td>
<td>3-6</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Cylindrical and Conical Coordinate Systems</td>
<td>3-19</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Use of ILINE and NMID</td>
<td>3-21</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Comparison of Transient Solutions, Computed and Exact.</td>
<td>3-26</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Steady State Solution for One-Dimensional Slider Bearing, A = 10.</td>
<td>3-27</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Steady State Solution for One-Dimensional Slider Bearing, A = 50.</td>
<td>3-28</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Shrouded Step Bearing Configuration</td>
<td>3-30</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Polar, Shrouded Step Bearing Mesh</td>
<td>3-31</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Rectangular Shrouded Step Bearing Mesh</td>
<td>3-32</td>
</tr>
<tr>
<td>3.6.4</td>
<td>Pad Geometry</td>
<td>3-34</td>
</tr>
<tr>
<td>3.6.5</td>
<td>Polar Velocity Field</td>
<td>3-35</td>
</tr>
<tr>
<td>3.6.6</td>
<td>Steady State Pressure Distribution for Rectangular Shrouded Step Bearing.</td>
<td>3-36</td>
</tr>
</tbody>
</table>
NOMENCLATURE

B = Typical bearing dimension (in.)
C₁ = Pressure prescribed boundary segment
C₂ = Flow prescribed boundary segment
g = Coordinate (in.)
H = Dimensionless film thickness (h/h₀)
h = Film thickness (in.)
h₀ = Typical film thickness (in.)
N = Unit outer normal
P = Dimensionless pressure (p/p₀)
p = Pressure (lb. in.⁻²)
−p = Boundary pressure (lb. in.⁻²)
p₀ = Ambient pressure (lb. in.⁻²)
p₀ = Initial pressure (lb. in.⁻²)
Q = Dimensionless, lineal mass flow (see eq. 2.1.6)
q = Lineal mass flow (lb. sec⁻¹)
R = Interior of lubricant region
R = Closed and bounded lubricant region
T = Dimensionless time (wt)
U = Surface velocity vector (in. sec⁻¹)
u = Viscosity (lb. sec. in.⁻²)
w = Typical frequency (Hz.)
$X$ = Dimensionless coordinate ($x/B$)

$x$ = Coordinate (in.)

$Y$ = Dimensionless coordinate ($y/B$)

$\lambda$ = Compressibility parameter, dimensionless (see eq. 2.1.4)

$\varphi$ = Squeeze number, dimensionless (see eq. 2.1.4)
1. INTRODUCTION

With numerical procedures becoming increasingly important in lubrication analysis, both in theoretical studies and practical bearing design, the need for more flexible and unified techniques becomes more and more apparent. While finite difference techniques have been almost universally employed in this area, a review of which may be found in [1], two recent studies [2, 3] utilizing the finite element method provide an alternative approach. One of the objectives of the present work is to extend the recent finite element studies to the solution of the general, transient lubrication problem.

A major advantage of the finite element method is that it allows the analyst to discretize the spatial domain in exactly the way he chooses, that is, with more nodes in areas where pressure gradient undergoes rapid changes and less, in other areas. In addition, it encompasses all lubrication problems governed by Reynolds' equation within the framework of one approach. Thus, no special treatment is accorded to various analyses just because of discontinuities, differing boundary configurations, etc.

Another objective of the present work is to develop a general purpose computer program that will allow analysis of general lubrication and seal problems governed by the Reynolds' equation, be they transient or steady-state, compressible or incompressible, hydrodynamic or hybrid. The usefulness of such a program in the routine analysis of non-routine designs cannot be over stated.

A less obvious but more significant utility of such a program lies in the eventual possibility of obtaining true automation in optimized bearing and seal design. Indeed, with the availability of the general purpose program described in this report and a library of

1-1
subroutines for generating the geometries of several bearing configurations, the techniques of linear and non-linear mathematical programming can be readily applied within the current state-of-the-art to bring about complete automation in optimizing lubrication and seal systems. It is hoped that such an application will become a reality in the none too distant future.

While the theoretical development for the finite element equations of the transient lubrication problem is new, the remaining aspects of the theory required for the general purpose computer program is not and the reader is directed to references [2, 3] for pertinent developments of those areas and additional information relating to the advantages of the finite element method as well as its application to lubrication problems.
2. FINITE ELEMENT FORMULATION

2.1 The Compressible Lubrication Problem

Let $\overline{R}$ denote a closed and bounded lubricant region with the interior $R$ and boundary $\overline{B}$. Let $(x,y)$ be a generic point of $\overline{R}$. The compressible lubrication problem consists of finding a solution $p(x,y,t)$ on $\overline{R}$ of the isothermal Reynolds' equation

$$v \cdot (h^3 p v - 6u \underline{v} ph) = 12u (ph)_t$$

(2.1.1)

with the boundary conditions

$$p = p_0 \text{ on } C_1$$

$$p(x,y,0) = p_0 \text{ on } R$$

(2.1.2)

and

$$n \cdot (Uh - h^3 v p/6u)p = 2q \text{ on } C_2$$

(2.1.3)

where, $n$ is the unit outer normal to $C_2$, $\overline{B}$ is $C_1 + C_2$ and $q$ is the linear mass flow across the boundary $C_2$. In addition, $u$ is the viscosity of the lubricant, $U$ is the vector velocity of the moving surface, $h(x,y,t)$ is the film thickness measured normal to the moving surface and a subscript comma denotes differentiation. $p_0$ is the initial value of $p$.

Recalling the derivation of Reynolds' equation, we note that equation 2.1.1 is valid only in those cases where the moving surface is normal everywhere to one of the coordinate directions of a coordinate system in three dimensional Euclidean space; in other words, equation 2.1.1 is not a general equation applicable to all surfaces.
Equation 2.1.1 and the associated boundary and initial conditions 2.1.2 and 2.1.3 may be put in a dimensionless form by the following transformations in a cartesian coordinate system.

\[ X = x/B \]
\[ Y = y/B \]
\[ T = \omega t \]
\[ P = p/p_a \]
\[ H = h/h_o \]
\[ \Delta = 6\mu \frac{U}{B/(p_a h_o^2)} \]
\[ \sigma = 12\mu \frac{B^2 \omega/(p_a h_o^2)} \]

(2.1.4)

where \( B, h_o \) and \( \omega \) are typical bearing dimension, film thickness and frequency, respectively; and, \( p_a \) is the ambient pressure. If no typical frequency exists in a particular problem, \( \omega \) may be conveniently set equal to 1.

With the transformations 2.1.4, the compressible lubrication problem becomes: Find a solution \( P(X,Y,T) \) on \( \mathbb{R} \) such that

\[ \nabla \cdot (H^3 \nabla P - \Delta PH) = \sigma (PH)_T \]  

(2.1.5)

\[ P = \frac{p}{p_a} \text{ on } C_1 \]

\[ P(X,Y,0) = \frac{p_o}{p_a} \text{ on } R \]

and

\[ \eta \cdot (\Delta H - h^3 V P)P = Q \text{ on } C_2 \]

where, the dimensionless lineal mass flow \( Q \) is given by

\[ Q = 12 \mu \frac{Bq}{h_o^3} \frac{p_a^2}{P_a} \]  

(2.1.6)
In previous studies [1,4] on the numerical solution of the transient, compressible lubrication problem, use of $PH^n (n = 1 \text{ to } 3)$ as the dependent variable has been argued to be advantageous, because gradients of $PH$ are, in general, smaller than those of $P$, and, in addition, knowledge of $H_T^{-1}$ is not required. While these considerations may or may not hold for the finite difference method, at least the former is of little significance in applying the finite element method because magnitude of the gradient by itself is of no consequence in determining the solution accuracy. Rather, it is the spatial rate of change of gradient that is significant, because, it determines the gradation required in the element sizes for achieving an optimal solution from the viewpoint of accuracy vs. number of elements utilized in the solution. Of more fundamental importance in applying the finite element method, however, is the requirement for the dependent variable to be of class $D_1$, namely, that

1. $p(x,y)$ is continuous for all $(x,y) \in \mathbb{R}$
2. $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$ are continuous for all $(x,y) \in \mathbb{R}$, except at a finite number of points at which one-sided derivatives, not necessarily equal, exist.

In general, this requirement cannot be met by $PH$ as the dependent variable, particularly in step and grooved bearings.

Thus, the subsequent developments in the present study are based on the use of $P$ alone as the dependent variable. While this places a requirement on knowledge of $H_T^{-1}$, it may not be a major inconvenience in most applications.

2.2 Variational Formulation

The time dependent compressible lubrication problem defined by equations 2.1.5 may be put in a variational form: Using the customary notation of the calculus of variations, and defining an admissible function as a continuous function that satisfies prescribed boundary
and initial conditions, let $F$ be the set of all admissible functions $P$. Let $\delta \phi(*)$ on $F$ be given by

$$
\delta \phi(P) = \int_{T_1}^{T_2} \int_{\Gamma} \left( \nabla^3 V P \cdot \nabla \delta P - \nabla P \cdot \nabla \delta P + \sigma \nabla \delta P \cdot \nabla P \right) \, dR 
+ \int_{C_2} \nabla \cdot (\nabla^3 V P \cdot \nabla P) \, dS - \int_{C_2} \sigma \nabla \delta P \cdot \nabla P \, dS \tag{2.2.1}
$$

Then,

$$
\delta \phi(P) = 0 \text{ over } F \tag{2.2.2}
$$

at a particular function $P$ if and only if $P$ is a solution of the compressible lubrication problem.

A proof for the foregoing is as follows: Let $P + \delta P \in F$. It then follows that

$$
\delta P(x,y,t) = 0 \text{ on } C_1 \\
\delta P(x,y,0) = 0 \text{ on } \Gamma \tag{2.2.3}
$$

and in addition, by virtue of Green's Theorem,

$$
\delta \phi(P) = \int_{T_1}^{T_2} \int_{\Gamma} \left\{ \nabla \cdot (\nabla^3 V P \cdot \nabla P) - \sigma \nabla \delta P \cdot \nabla P \right\} \, dR 
+ \int_{C_2} \nabla \cdot (\nabla^3 V P \cdot \nabla P) \, dS + \int_{C_2} \sigma \nabla \delta P \cdot \nabla P \, dS \tag{2.2.4}
$$
First, if $P$ is a solution of the compressible lubrication problem, then $P$ satisfies equations 2.1.5 and hence, through 2.2.4, equation 2.2.2 holds. Conversely, if $P$ satisfies equation 2.2.2, then equation 2.2.4 implies

$$
\int_{T_1}^{T_2} \int_R \left[ \nabla \cdot (PH^3VP - APH) - \sigma (PH)_T \right] \delta P \, dR \, dt = 0 \quad (2.2.5)
$$

and

$$
\int_{T_1}^{T_2} \int_{C_2} \left\{ \nabla \cdot (PH^3VP - PHA) + Q \right\} \delta P \, ds = 0 \quad (2.2.6)
$$

for every $P$ that satisfies 2.2.2. By virtue of the fundamental lemma of the calculus of variations, $P$ then satisfies equations 2.1.5.

2.3 Temporal Discretization

Let the time dependence of $P$ be assumed to be of the form

$$
P = P_1 + \phi P_2 \quad (2.3.1)
$$

where $P_1$ and $P_2$ are spatial functions and $\phi$ is a temporal function in the interval $(T_1, T_2)$ and defined as

$$
\phi = (T - T_1)/(T_2 - T_1) \quad (2.3.2)
$$

While it is not necessary, for convenience of avoiding numerical integration in the time domain, one may suppose that $A$, $H$ and $Q$ are also of the form

$$
\begin{align*}
A &= A_1 + \phi A_2 \\
H &= H_1 + \phi H_2 \\
Q &= Q_1 + \phi Q_2
\end{align*}
\quad (2.3.3)
$$

THE FRANKLIN INSTITUTE RESEARCH LABORATORIES
This will be a reasonable supposition provided that \((T_1, T_2)\) is a small enough interval. Moreover, if the interval is small enough, \(P_2 \ll P_1\).

Noting that
\[
\int_{T_1}^{T_2} \phi dT = (T_2 - T_1)/(j + 1)
\]
(2.3.4)

and
\[
\delta P = \phi \delta P_2
\]
(2.3.5)

implying that \(P_1\) is a solution of the compressible lubrication problem at time \(T = T_1\), equation 2.2.1, with the aid of equations 2.3.1 through 2.3.5, yields

\[
\delta \phi(P) = \int_{R} \left\{ P_1 \nu P_1 \cdot \nabla \delta P_2 + f_2(P_1 \nu P_2 + P_2 \nu P_1) \cdot \nabla \delta P_2 + f_3 P_2 \nu P_2 \cdot \nabla \delta P_2
\]
\[
+ P_1 d_1 \cdot \nabla \delta P_2 + P_2 d_2 \cdot \nabla \delta P_2 + \alpha P_1 \delta P_2 + \beta P_2 \delta P_2 \right\} dR
\]
\[
+ \int_{C_2} (\frac{1}{2} Q_1 + \frac{1}{3} Q_2) \delta P_2 ds
\]
(2.3.6)

where,
\[
f_1 = (H_1^3/2) + H_1^2 + H_1^2 H_2 + (3H_1 H_2^2/4) + (H_2^3/5)
\]
\[
f_2 = (H_1^3/3) + (3H_1^2 H_2/4) + (3H_1 H_2^2/5) + (H_2^3/6)
\]
\[
f_3 = (H_1^3/4) + (3H_1^2 H_2/5) + (H_1 H_2^2/2) + (H_2^3/7)
\]
\[
\alpha = \sigma P_1 H_2/(2\Delta T)
\]
\[
\beta = \sigma (H_1 + 4H_2/3)/(2\Delta T)
\]
\[ d_1 = -(H_1 \Delta_1/2) + (H_2 \Delta_2/3) + (H_2 \Delta_1/3) + (H_2 \Delta_2/4) \]
\[ d_2 = -(H_2 \Delta_1/4) + (H_2 \Delta_2/5) + (H_1 \Delta_1/3) + (H_2 \Delta_2/4) \]
\[ \Delta T = T_2 - T_1 \quad (2.3.7) \]

Inasmuch as \( f_3 < f_2 \) and \( P_2 < P_1 \), in comparison with \( P_1 f_1 \nabla P_1 \), \( P_2 f_3 \nabla P_2 \) may be neglected. Indeed if \( P_1 \) is \( O(1) \) and \( P_2 (10^{-4}) \), error resulting from this approximation is of the same order as the round-off error with eight digit, finite arithmetic.

With the foregoing approximation, the Euler-Langrange equation of 2.3.6 is

\[ \nabla \cdot [P_1 f_2 \nabla P_2 + P_2 f_3 \nabla P_1] + P_1 f_1 \nabla P_1 - P_1 d_1 - P_2 d_2 = \sigma [P_1 H_2 + P_2 H_1 + 4 P_2 H_2/3]/(2\Delta t) \quad (2.3.8) \]

Alternatively, equation 2.3.8 can be derived from 2.1.5 with the assumed functional form 2.3.1 by applying Galerkin's Principle and neglecting \( P_2 f_3 \nabla P_2 \) in comparison with \( P_1 f_1 \nabla P_1 \).

2.4 Spatial Discretization

The procedure for discretizing the continuum problem into finite element equations is described in [2,3]. Let the spatial region of interest be subdivided into \( M \) elements of \( J \) nodes each. For a typical element \( E_m \), the pressure field \( P_{2m} \) may be assumed as

\[ P_{2m}(x,y) = f^T a \quad (2.4.1) \]

where, \( a \) is a column vector of \( J \) constants and \( f \) is a column vector of \( J \) interpolation functions defining the pressure within the element. The constants \( a \) are related to the unknown nodal pressures and are determined by evaluating 2.4.1 at the nodes.
where $F$ is the matrix of interpolation functions evaluated at the nodes and $P_{2m}$ is the column vector of nodal pressures.

From 2.4.2,

$$ g = F^{-1} P_{2m} \quad (2.4.3) $$

and 2.4.1 may be expressed as

$$ P_{2m}(x,y) = T_{m}^{T} P_{2m} \quad (2.4.4) $$

where

$$ T_{m} = f^{T} F^{-1} \quad (2.4.5) $$

Augmenting $T_{m}$ with zero elements such that the non-zero elements correspond to the nodes of element $m$, a new column vector $T_{m}$ is defined. Equation 2.4.4 may now be put in the form

$$ P_{2m} = T_{m}^{T} P_{2} \quad (2.4.6) $$

where $P_{2}$ is now the column vector of nodal pressures for the entire finite element assemblage.

A column vector of pressure gradients $g_{m}$ may be conveniently defined as

$$ g_{m} = B_{m} P_{2} \quad (2.4.7) $$

where,

$$ B_{m} = <T_{m,x}^{T}, T_{m,y}^{T}>^{T} \quad (2.4.8) $$
The variational expression 2.3.6 may now be expressed in matrix form as

\[
\delta \Phi(P) = \sum_{m=1}^{m} \int_{R_m} \delta P^T \left\{ \left[ B_m^T C_m B_m + B_m^T D_m \right] \right\} P_m^m + ST^T \right\} \delta P^T \int_{C_{2m}} T_m \gamma ds_m
\]

where,

\[
M_m = \begin{bmatrix}
-T_m \\
T_m
\end{bmatrix}
\]

\[
C_m = \text{DIAG} \left( P_1 f_2, P_1 f_2 \right)
\]

\[
D_m = \text{DIAG} \left( P_1 x f_2 + u_2, P_1 y f_2 + v_2 \right)
\]

\[
E_m = \left\langle (P_1 x f_1 + u_1)P_1, (P_1 y f_1 + v_1)P_1 \right\rangle^T
\]

\[
\gamma = (Q_1/2) + (Q_2/3)
\]

\[
u_1 = d_1 \cdot i_x
\]

\[
u_2 = d_2 \cdot i_x
\]

\[
v_1 = d_1 \cdot i_y
\]

\[
v_2 = d_2 \cdot i_y
\]

where \(i_x\) and \(i_y\) are unit vectors in the \(x\) and \(y\) directions respectively, and \(f_1, f_2, \alpha\) and \(\beta\) are defined by equations 2.3.7.

Since \(\delta P_2\) is an independent variation, from 2.4.9,

\[
- \sum_{m=1}^{m} \int_{R_m} \left\{ \left[ B_m^T C_m B_m + D_m \right] \right\} P_m^m + ST^T \right\} \delta P^T \int_{C_{2m}} T_m \gamma ds_m
\]

\[
= \sum_{m=1}^{m} \left\{ \left( B_m^T C_m B_m + D_m \right) \right\} P_m^m + \int_{C_{2m}} T_m \gamma ds_m
\]
In 2.4.11, the line integrals are evaluated only for those elements that adjoin the boundary B.

2.5 Triangular Elements

Referring to Figure 2.5.1(a) let $L_i$, $i = 1$ to 3, be linear functions in $x$ and $y$ such that

$$L_i = a_i + b_i x + c_i y$$  \hspace{1cm} (2.5.1)

where $L_i$ equals 1 at node 1 and zero at the remaining two nodes. The constants $a_i$, $b_i$, $c_i$ may now be explicitly stated as

$$a_1 = (x_3 y_2 - x_2 y_3)/2A$$
$$b_1 = (y_3 - y_2)/2A$$
$$c_1 = (x_2 - x_3)/2A$$  \hspace{1cm} (2.5.2)

where $A$ is the area of the element and the subscripts indicate element nodes. By a permutation of the indices, the remaining constants may be deduced from 2.5.2 in a straightforward manner. Thus, the column vector $T^T_m$ for the triangular element becomes

$$T^T_m = <L_1, L_2, L_3>$$  \hspace{1cm} (2.5.3)

While this element allows almost a constant gradient within it, a refined element may be set up with additional nodes. Thus, referring to Figure 2.5.1(b),

$$T^T_m = <L_1(2L_1 - 1), L_2(2L_2 - 1), L_3(2L_3 - 1), 4L_1L_2, 4L_2L_3, 4L_3L_1>$$  \hspace{1cm} (2.5.4)

where now the mid-side nodes come into effect. This element allows a linear variation of the gradient within it.
Figure 2.5.1 Triangular Elements

(a) THREE POINT TRIANGLE

(b) SIX POINT TRIANGLE
The required spatial integration over the element may be performed by a three point rule. Denoting a point \( X_i \) in the triangle by its three area coordinates,

\[
\int_{\Delta} F \, dA = \sum_{i=1}^{3} W_i \, F(X_i)
\]  

(2.5.5)

where the weights \( W_i \) and the \( X_i \) in terms of the triangular coordinates are given by [5]

\[
X_1 = (2/3, 1/6, 1/6); \quad W_1 = 1/3
\]

\[
X_2 = (1/6, 2/3, 1/6); \quad W_2 = 1/3
\]

\[
X_3 = (1/6, 1/6, 2/3); \quad W_3 = 1/3
\]  

(2.5.6)

It is also possible to evaluate the integrals in a closed form by resorting to the formula [6]

\[
\int_{\Delta} L_i L_j L_k \, dA = \frac{i! \, j! \, k!}{(n + 2)!} \, 2A
\]  

(2.5.7)

where

\[
n = i + j + k
\]  

(2.5.8)

and \( i, j, k \) are non-negative integers.

While 2.5.7 is exact for polynomials of all degrees, the numerical integration rule 2.5.5 is exact for polynomials up to the second degree. Since three point element utilizes polynomials of the first degree and the six point element polynomials of the second degree, the numerical rule 2.5.5 is completely adequate for the present purposes.
2.6 Quadrilateral Elements

Referring to Figure 2.6.1(a), two coordinates \((\xi, \eta)\), associated with a quadrilateral element are determined such that

\[
\begin{align*}
\xi &= 1 \text{ on side 23} \\
\xi &= -1 \text{ on side 14} \\
\eta &= -1 \text{ on side 12} \\
\eta &= 1 \text{ on side 43}
\end{align*}
\]

The interpolation functions \(N_i(\xi, \eta)\) are defined such that

\[
\begin{align*}
\sum_{i=1}^{J} N_i x_i &= x \\
\sum_{i=1}^{J} N_i y_i &= y
\end{align*}
\]

where \(J\) designates the number of nodes in the element and \(i\) refers to the individual nodes in the element. With this definition of \(N_i\), the functional form of the pressure \(P\) over the element may be expressed as

\[
P = \sum_{i=1}^{J} N_i P_i
\]

For a four-point quadrilateral, the functions \(N_i\) are given by [3]

\[
\begin{align*}
N_1 &= (1 - \xi)(1 - \eta)/4 \\
N_2 &= (1 + \xi)(1 - \eta)/4 \\
N_3 &= (1 + \xi)(1 + \eta)/4 \\
N_4 &= (1 - \xi)(1 + \eta)/4
\end{align*}
\]
Figure 2.6.1 Quadrilateral Elements

(a) FOUR POINT QUADRILATERAL

(b) EIGHT POINT QUADRILATERAL
and the column vector $T_m$ for the four point quadrilateral is defined by

$$T_m^T = \langle N_1, N_3, N_3, N_4 \rangle$$  \hspace{1cm} (2.6.4) \hspace{1cm} \text{(2.6.4)}

Referring to Figure 2.6.1(b), for the eight point quadrilateral, the function $N_i$ for $i = 1,4$ are defined by

$$N_i = [(1 + \xi_\circ)(1 + \eta_\circ) - (1 - \xi^2)(1 + \eta_\circ) - (1 + \xi_\circ)(1 - \eta^2)]/4$$  \hspace{1cm} (2.6.3) \hspace{1cm} \text{(2.6.3)}

where,

$$\xi_\circ = \xi \xi_i$$

$$\eta_\circ = \eta \eta_i$$  \hspace{1cm} (2.6.6) \hspace{1cm} \text{(2.6.6)}

with $\xi_i$ and $\eta_i$ being equal to either +1 or -1 depending on the node $i$.

The interpolation functions $N_i$ for mid-side nodes are defined by [7]

$$N_i = (1 - \xi^2)(1 + \eta_\circ)/2 \text{ for } i = 5 \text{ and } 7$$  \hspace{1cm} (2.6.7) \hspace{1cm} \text{(2.6.7)}

and

$$N_i = (1 + \xi_\circ)(1 - \eta^2)/2 \text{ for } i = 6 \text{ and } 8$$  \hspace{1cm} (2.6.8) \hspace{1cm} \text{(2.6.8)}

The column vector $T_m$ for the eight point quadrilateral is defined by

$$T_m^T = \langle N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8 \rangle$$  \hspace{1cm} (2.6.9) \hspace{1cm} \text{(2.6.9)}

Inasmuch as the interpolation functions are defined in terms of $\xi$ and $\eta$, a relation between the derivatives in the $(\xi,\eta)$ and $(x,y)$ coordinate systems is required and this is given by

2-15
where, $J$ is the Jacobian matrix given by

$$J = \begin{bmatrix}
N_{1,\xi} & N_{2,\xi} \\
N_{1,\eta} & N_{2,\eta}
\end{bmatrix} \begin{bmatrix} x_1 & y_1 \\
x_2 & y_2
\end{bmatrix}$$

From 2.6.10, the derivatives in $(x,y)$ system are given by

$$N_{1,x} = \begin{bmatrix} N_1, \xi \\
N_1, \eta
\end{bmatrix} J^{-1}
$$

and

$$N_{1,y} = \begin{bmatrix} N_1, \xi \\
N_1, \eta
\end{bmatrix} J^{-1}
$$

Since the $(\xi,\eta)$ coordinates range over $-1$ to $1$, it is convenient to integrate in that system. Thus, the substitution

$$dx \; dy = \det J \; d\xi \; d\eta$$

and change of limits to $-1$ and $+1$ allows numerical integration in the Gaussian coordinates $\xi, \eta$. A Gaussian quadrature scheme, based on a two point rule in each coordinate direction is given by

$$\int F(\xi,\eta) \; d\xi \; d\eta = \sum_{i=1}^{2} \sum_{j=1}^{2} F(\xi_i,\eta_j)$$

(2.6.15)
where \( \xi_i \) and \( \eta_j \) are given by \( \pm 0.57735027 \). This rule is exact for all polynomials up to the second degree.

2.7 Boundary Conditions

For those nodes that lie on the boundary segment \( C_1 \), pressure boundary conditions apply. Suppose that node \( i \) is on \( C_1 \) and the pressure at node \( i \) is a function of time given by \( \bar{P}_i(T) \). Since equations 2.4.11 have the incremental pressure \( P_2 \) as the unknown, the incremental boundary condition must be deduced. Thus if \( \Delta T \) is the time interval and \( T_1 \) is the time at which the boundary condition is to be applied, then at node \( i \),

\[
P_{2i} = \bar{P}_i(T_1 + \Delta T) - \bar{P}_i(T_1)
\]

(2.7.1)

If \( \bar{P}_i \) is a constant, then \( P_{2i} \) is clearly equal to zero. To apply the boundary condition to the system of equations, let

\[
[K] P_2 = R
\]

(2.7.2)

where \( K \) is the assembled matrix and \( R \) is the vector of right hand sides. If the \( i \)th node is to be a pressure prescribed node, then, we modify 2.7.2 in this manner

\[
R_j = K_{ji} P_{2i} + R_j \quad j \neq i
\]

\[
0. \to K_{ji}, \quad K_{ij} \text{ for } i \neq j
\]

\[
1. \to K_{ii}
\]

\[
P_{2i} = R_i
\]

(2.7.3)

where the subscripted quantities are matrix elements of 2.7.2.

In a similar fashion, for each of the finite elements that has one or more sides on the boundary segment \( C_2 \), the prescribed flow is
partitioned between the nodes on that side and simply added into the corresponding elements of the matrix $K$. Where symmetry or other considerations imply zero flow, $Q_1$ and $Q_2$ are set to zero.

2.8 Solution Scheme

The set of equations 2.4.11 for the incremental pressure $P_2$ constitute a banded, linear algebraic system that can be solved by Gaussian Elimination. Subsequent to solution, the incremental pressure vector is accumulated at each step to yield the pressure field at that time.
3. PROGRAM DESCRIPTION

3.1 Introduction

The computer program FELUR, (an acronym derived from Finite Element LUBrication Analysis), is a general purpose program written in Fortran IV for use with medium core-size computers. In general, the program can be used for analyzing any bearing, seal or pump type device in which the physical behavior is adequately predicted by Reynolds' equation.

In developing the program, while the temptation to achieve total generality was great, some restraint had to be exercised, because, at least with computer programs, generality is synonymous with maximality of input. Moreover, generality implies several options in the flow of information, and options in turn imply CPU consuming decisions. Thus, the final version of the program includes the following options which were judged to represent a good balance between generality, ease of input preparation and computation cost.

a. Fluid Films
   • Incompressible
   • Isothermal Compressible

b. Fluid Mechanics
   • Hydro-dynamic
   • Hydrostatic
   • Hybrid

c. Temporal Description
   • Steady-State
   • Transient

d. Spatial Description
   • Cartesian
• Polar
• Cylindrical
• Conical

The output from the program consists of a table of pressures within the device analyzed, the components of pressure integrated over the entire film in three mutually perpendicular directions, and flow across specified paths. In addition, input parameters, either read in, or internally generated, are also listed in the output printout to facilitate input checking.

3.2 Major Program Variables

COMMON/NAME/

<table>
<thead>
<tr>
<th>COMMON/NAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE(18)</td>
<td>Job Title</td>
</tr>
<tr>
<td>KONTRL(6)</td>
<td>Branch Control</td>
</tr>
<tr>
<td></td>
<td>&quot; (1) = 1 Manual Input for Grid</td>
</tr>
<tr>
<td></td>
<td>&quot; (1) = 2 Automatic Input for Grid</td>
</tr>
<tr>
<td></td>
<td>&quot; (2) = 1 Steady-State</td>
</tr>
<tr>
<td></td>
<td>&quot; (2) = 2 Transient</td>
</tr>
<tr>
<td></td>
<td>&quot; (3) = 1 Compressible</td>
</tr>
<tr>
<td></td>
<td>&quot; (3) = 2 Incompressible</td>
</tr>
<tr>
<td></td>
<td>&quot; (4) = 1 Hydrostatic or Hydrodynamic</td>
</tr>
<tr>
<td></td>
<td>&quot; (4) = 2 Hydrostatic and Hydrodynamic (Hybrid)</td>
</tr>
<tr>
<td></td>
<td>&quot; (5) = 1 Cartesian (X,Y)</td>
</tr>
<tr>
<td></td>
<td>&quot; (5) = 2 Polar (R,θ)</td>
</tr>
<tr>
<td></td>
<td>&quot; (5) = 3 Cylindrical (θ,2), R</td>
</tr>
<tr>
<td></td>
<td>&quot; (5) = 4 Conical (R,θ), α</td>
</tr>
<tr>
<td></td>
<td>&quot; (6) = 1 Plots required</td>
</tr>
<tr>
<td></td>
<td>&quot; (6) = 2 Plots not required</td>
</tr>
<tr>
<td>NLOOPS</td>
<td>Number of paths over which flow is to be computed</td>
</tr>
<tr>
<td>TRUNC</td>
<td>Needed only for steady state, compressible case. Stops computation when average change in pressure is less than TRUNC</td>
</tr>
</tbody>
</table>
COMMON/MESH/

NNODES Number of nodes in assemblage.

NPANS Number of elements in assemblage.

NPJTS(I,J) Table of node numbers for elements. In J = 9 is stored the number of nodes in the element. I is the element number.

ITHIK(I) Thickness clue set up by user to indicate if the Ith element is in a step or a groove. Not needed when film thickness is continuous.

COORD(J,I) Coordinates of the Ith element. After call to TRANF, these are cartesian.

COMMON/FIELD/

X(J) Element nodal coordinates x, relative.

Y(J) Element nodal coordinates y, relative.

KLUE Number of nodes in element.

IFILM Thickness clue for element obtained from ITHIK.

PSI(J) Element nodal pressures from previous step.

D(L,M) Element matrix.

RR(L) Element right hand side vector.

SIG Sigma/(2AT)

XA Absolute coordinate x of node 1 in element.

YA Absolute coordinate y of node 2 in element.

VMIN Typical film thickness.

OMEGA Typical frequency.

COMMON/EQUATN/

A(I,J) Augmented matrix of equations.

NEQ Number of equations.
NBAND
Column location of right hand side in A.

NBAND2
Bandwidth.
PSTAR(I)
Dimensionless pressure at the I-th node.

COMMON/BOUND/
NBC
Number of nodes at which pressure boundary conditions are specified.

NBCNDS(I)
Array of node numbers at which boundary conditions are specified.
P1(I)
Specified pressure at the nodes. P1(I) is in one-to-one correspondence with NBCNDS(I).

COMMON/TIMES/
TIME
Current time.

DT
Time increment.

KSTEP
Current step number.

VKSLAM
Viscosity.

ALPRAD
Fixed radius in cylindrical coordinates or half-cone angle in conical coordinates.
PAMB
Ambient pressure.

AU1
Maximum speed in 1-direction.

AU2
Maximum speed in 2-direction.

DPLA
Fractional increase in speed.

PLA
Current fraction of maximum speed.

3.3 Subroutine Descriptions

ENTRY FELUB is the main routine which determines the overall computational sequence. A general flow chart for FELUB is given in 3-4.
Figure 3.2.1. With a call to INPUT, required data for a problem is assembled. If a transformation of coordinates is required, TRANF is called. Next, influence coefficients for load computation are established by a call to INFYSYS. Required arrays are initialized and an incremental loop on NSTEPS is entered. A call to SPEED establishes operating speed at that instant of time. The finite element equation for the step are assembled by a call to SYSTEM. These are then solved by a call to SOLVE. Results are accumulated and displayed by a call to OUTPUT. After all incremental steps are completed, if ISENT has been set to zero, program execution stops; otherwise control returns to ENTRY point.

SUBROUTINE CONTRL is called by INPUT once for each problem. CONTRL reads the two command cards that follow the title card in the input deck and analyzes them to fill the vector KONTRL (6). In the first command card, it detects the first character of the first, fourth and sixth words respectively. It fills KONTRL (6) in the following manner.

1st Command Card

<table>
<thead>
<tr>
<th>Word No.</th>
<th>Character No.</th>
<th>Character</th>
<th>KONTRL(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>M</td>
<td>KONTRL(1) = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>A</td>
<td>KONTRL(1) = 2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>S</td>
<td>KONTRL(2) = 1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>T</td>
<td>KONTRL(2) = 2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>C</td>
<td>KONTRL(3) = 1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>I</td>
<td>KONTRL(3) = 2</td>
</tr>
</tbody>
</table>

2nd Command Card

<table>
<thead>
<tr>
<th>Word No.</th>
<th>Character No.</th>
<th>Character</th>
<th>KONTRL(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>S</td>
<td>KONTRL(4) = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>H</td>
<td>KONTRL(4) = 2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>E</td>
<td>KONTRL(5) = 1</td>
</tr>
</tbody>
</table>
Figure 3.2.1 General Flow Chart for FELUB
For the purposes of this routine, words are defined as groups of non-blank characters separated by one or more blank fields. The labelled common /FIELD/ is used in this routine as scratch storage and not for transfer of information to other routines.

SUBROUTINE INPUT is called by FELUB once for each problem. It assembles the input data for each problem from the input deck. Execution control is set up by a call to C\textsuperscript{ONTROL}. One of the two options is exercised for grid generation: If a user supplied routine AUTGRD is available, it is called. Otherwise MANGRD is called. These routines supply nodal coordinates and a table of node numbers and type for each finite element in the assemblage. In addition, boundary conditions for the problem are established.

The input parameters are put in a dimensionless form and the bandwidth for the problem is determined. All the input quantities are displayed via the printer.

This routine uses labelled common /NAME/, /MESH/, /FIELD/, /EQUATN/, /BOUND/, and /TIMES/. Directly or by external references, all the commons except /FIELD/ and /EQUATN/ are filled by this routine. In the latter two, the variables NEQ, NBAND, NAUG, NBAND2, SIG, HMIN and OMEGA are also filled.

SUBROUTINE MANGRD is called by INPUT once for each problem. It generates grid information and boundary conditions. Special features are provided for minimizing input when four point quadrilateral
elements are used. In reading the sequential, nodal coordinate data, omitted information is generated on the basis of equal division on a straight line connecting the first and last nodes bounding the missing information. Similarly, in reading the sequential, element data, for missing elements, the last element nodes are incremented by 1 or 2 as specified. Where missing information cannot be generated in a consistent fashion, an Error Stop is made.

For each element, in addition to the nodal data, information regarding the manner in which the film thickness is to be determined is read in. This is particularly useful, indeed necessary, in step bearings where coordinate information is insufficient to determine film thickness at boundaries of grooves and lands.

This routine fills labelled common /MESH/ and /BOUNDR/. In addition, in /TIMES/, the variable ALPRAD is set for problems in which cylindrical or conical coordinates are utilized.

SUBROUTINE TRANF is called by FELUB once for each problem in which the coordinate system is not a cartesian system. Depending on the coordinate system used, it takes one of three branches to transform the coordinates into a cartesian system. In the case of cylindrical and conical systems, the developed surface is used in addition. The transformations are as follows:

a) Polar Coordinates
   Input: R and θ for each node N in COORD(I,N), I=1,2
   Output: X = COORD(1,N) = R cos θ
          Y = COORD(2,N) = R sin θ

b) Cylindrical Coordinates
   Input: θ_d and Z for each node N in COORD(I,N), I=1,2 and radius R in ALPRAD
   Output: X = COORD(1,N) = Rθ_d
          Y = COORD(2,N) = Z
   Input θ_d is in degrees. Subroutine converts it to θ_r in radians.
c) Conical Coordinates

Input: \( R \) and \( \theta_d \) for each node \( N \) in \( \text{COORD}(I,N) \) \( I=1,2 \) and half-cone angle \( \alpha_d \) in ALPRAD

Output: \( X = \text{COORD}(1,N) = \frac{(R \cos (\sin \alpha_r \theta_r))}{\sin \alpha_r} \)
\( Y = \text{COORD}(2,N) = \frac{(R \sin (\sin \alpha_r \theta_r))}{\sin \alpha_r} \)

Input \( \theta_d \) and \( \alpha_d \) are in degrees. Subroutine converts them to \( \theta_r \) and \( \alpha_r \) in radians.

Labelled commons /TIMES/ and /MESH/ are used in this routine. Argument NCORD is input to the routine and identifies coordinate type as follows:

\( \text{NCORD} = 2 \) polar
\( = 3 \) cylindrical
\( = 4 \) conical

If \( \text{NCORD} = 1 \), this routine is not called.

SUBROUTINE SYSTEM is called by FELUB once for each increment or step in each problem. SYSTEM begins by initializing the system matrix \( A \) to zero and entering a loop on all the elements in the assemblage. For each element, it fills the labelled common /FIELD/ with element data and calls either FTCF3 or FTCF4 depending on whether the element is a triangle or a quadrilateral. These routines return the element matrix and right hand side vector in the variables \( D \) and \( RR \) respectively in /FIELD/, which are then added into the final system \( A \). The column NAUG in \( A(N, NAUG) \) is the accumulated right hand side and \( A \) may be called the augmented matrix of the system. If an element with number of nodes other than 3, 4, 6 or 8 is encountered, this routine terminates with an error stop. After the entire system is assembled, boundary conditions are applied and control is returned to FELUB.

Labelled commons /NAME/, /FIELD/, /MESH/, /EQUATN/, /BOUND/ and /TIMES/ are used by this routine.

SUBROUTINE FTCF3 is called by SYSTEM. It generates the element matrix and right hand side for triangular elements with three or
six nodes. Appropriate interpolation functions and their derivatives are obtained by a call to TXT. A three point rule is used to integrate over the area of the element. Depending on the coordinate systems used and whether the problem is compressible or incompressible, alternative computational procedures are utilized in generating the element matrix and right hand side. Labelled common /NAME/, /TIMES/ and /FIELD/ are used by this routine.

SUBROUTINE FTCF4 is called by SYSTEM for quadrilateral elements with four or eight nodes. Interpolation functions and their derivatives are obtained by a call to DXD. A two point rule giving rise to four terms is used to integrate over the area of the element. This routine is similar to FTCF3 in all other respects.

SUBROUTINE INFSYS called by FELUB, once for each problem, generates three influence vectors for load computation in each of the three coordinate directions. Scalar product of these vectors with the vector of nodal pressures yields the load components. The routine begins by initializing FORMOM to zero and entering a loop on all the elements. For each element, at each node, unit pressure is assumed while the remaining nodes in the element are assumed to be at zero pressure. This pressure field is integrated over the element and the components are accumulated in FORMOM. The computation is repeated for each node in the element. The integration is done by a call to INGRT3 for triangular elements and INGRT4 for quadrilaterals. If an element with number of nodes other than 3, 4, 6 or 8 is encountered, this routine terminates with an error stop.

This routine uses labelled commons /MESH/, /FIELD/ and /INFLNC/.

Subroutine INGRT3 is called by INFSYS once for each triangular element. It generates influence coefficients for load computation by integrating a pressure field of unity at one node and zero at the remaining and integrating over the area of the element. A three point
integration rule is used. Interpolation functions for the pressure
distribution are obtained by a call to TXT.

This routine uses labelled commons /NAME/, /TIMES/ and
/FIELD/.

SUBROUTINE INGRT4 is called by INFSYS once for each quadri-
lateral element. It is similar to INGRT3 in its function. A three
point rule giving rise to nine terms is used in the integration. A
call to DXD provides the interpolation functions for determining the
pressure distribution.

This routine uses labelled commons /NAME/, /TIMES/ and
/FIELD/.

SUBROUTINE OUTPUT is called by FELUB once for each step in
all cases except steady-state problems in which it is called only
after the final step. It computes load components and outputs them
via the printer along with the nodal pressures.

Labelled commons /NAME/, /MESH/, /EQUATN/, /TIMES/ and
/INFLNC/ are used.

SUBROUTINE SOLVE is called by FELUB once for each step. It
solves the banded system of equations of the augmented matrix A, stored
in the labelled common /EQUATN/ by Gaussian Elimination. If, during
forward elimination, a diagonal term is found to be zero, this routine
terminates with an error stop. The solution is returned in the NAUG
column of A(N,NAUG).

Labelled common /EQUATN/ is used by this routine.

SUBROUTINE FLOW called by FELUB computes flow over prescribed
paths. It calls FTCF3 or FTCF4, depending on whether triangular or quad-
rilateral elements are involved. For each element, the element matrix
is assembled and with the known solution vector, the flow across the
specified edge is obtained by appropriate multiplication of the element
matrix and nodal pressures for the element.
This routine uses /EQUATN/ for scratch storage. The other
labelled commons are /FIELD/, /MESH/ and /TIMES/.

SUBROUTINE TXT is called by FTCF3 and INGRT3. It returns
the interpolation functions and their derivatives for three and six point
triangles.

SUBROUTINE DXD is called by FTCF4 and INGRT4. It returns the
interpolation functions and their derivatives for four and eight point
quadrilaterals.

SUBROUTINE SPEED is a user supplied routine. Its input is
the variable TIME representing time. This routine returns speed of the
bearing at any instant of time and the change in speed during the next
increment of time. These values are returned as a fraction of the max-
imum speed expected. Thus PLA equals the fraction corresponding to the
current value and DPLA equals incremental fraction. The version of this
routine supplied with the program may be used for all applications ex-
cept those in which the speed is to be changed during the computation.

SUBROUTINE FILM is a user supplied routine. Its inputs are
X1, X2, TIME, IFILM where X1 and X2 are coordinates of a location in
the bearing in the untransformed coordinates, TIME is time and IFILM
is a clue indicating whether the location is in a land region or groove
region. The routine returns H which is the film thickness at time =
TIME, and HDOT, which is the rate of change of film thickness.

3.4 Input Preparation

A natural starting point for the analysis of any bearing,
seal or viscosity pump by the FELUB program is to make a list of the
parametric conditions for which the device is to be analyzed. This
may include such things as viscosity, speed, etc. At this point, if
the geometry of the device is to be modified several times, while the
basic configuration is kept unchanged, a decision should be made re-
garding the possibility of preparing a AUTGRD routine for automatically
generating the mesh layout for the device under consideration.
A drawing of the developed view of the film can be particularly useful in making a judgement regarding regions in which rapid changes of the gradient of pressure may occur. In such regions, naturally, one would wish to employ a finer mesh than elsewhere. In addition, a drawing may aid one in delineating sections across which flow computation might be desirable.

Supposing that one has assembled all the pertinent information and subdivided the fluid film into an assemblage of finite elements. The next task is to number the nodes of the elements in such a fashion that in the entire assemblage, the maximum of the difference between the largest and smallest nodal numbers appearing in any element is as small as possible. While this objective is not a crucial consideration in so far as solution accuracy is concerned, it has considerable bearing on minimizing the computation time.

At this point one is ready to prepare input and the input map given in Table 3.4 may be consulted. A description of each of the card types and variables within each card type follows.

**Card Type A**

This card contains alpha-numeric information describing the analysis. This information is printed as a title on each page of the output. If the information is centered on the card, it will appear centered on a 128 column printout.

**Card Type B**

This card is essentially a command card that selects options in the analysis. This card is analyzed in the program as free field input. Words are assumed to be groups of non-blank characters separated by blanks. The first word can begin anywhere on the card and words must be separated by one or more blanks. The first word is allowed to be one of the two words: MANUAL or AUTOMATIC. The word AUTOMATIC is to be used only when the user supplies the AUTGRD routine. The second and
Table 3.4
INPUT MAP FOR 'FELUB', (VERSION 3, REV. 1)

<table>
<thead>
<tr>
<th>CARD TYPE</th>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>DEFINITION</th>
<th>FORMAT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1-80</td>
<td>TITLE</td>
<td>Job Title</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>1-80</td>
<td>NZ</td>
<td>Type of Analysis</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>MANUAL</strong></td>
<td><strong>AUTOMATIC</strong></td>
</tr>
<tr>
<td>C</td>
<td>1-80</td>
<td>NZ</td>
<td>Type of bearing and output</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>STANDARD</strong></td>
<td><strong>HYBRID</strong></td>
</tr>
<tr>
<td>D</td>
<td>1-6</td>
<td>NODES</td>
<td>Number of nodes</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>7-12</td>
<td>NFANS</td>
<td>Number of elements</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>13-18</td>
<td>NBC</td>
<td>Number of pressure prescribed nodes</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>19-30</td>
<td>ALPRAD</td>
<td>If card type C indicates</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Cartesian - ALPRAD is not needed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Polar - ALPRAD is not needed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Cylindrical - ALPRAD is radius</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Conical - ALPRAD is half cone angle</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>5-6</td>
<td>ILINE</td>
<td>Connector</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>7-12</td>
<td>N</td>
<td>Node number</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
<td>COORD(1,N)</td>
<td>Coordinate 1 for node N =</td>
<td>E</td>
</tr>
</tbody>
</table>

*I-Format requires integer right justified in field.
E-Format can accept a decimal number anywhere in field, or an exponential number with exponent (base 10) right justified in field, or integer right justified in field.
A-Format requires Alphanumeric Data. All zeros may be input as blank.
Z-Free field with blanks as separators. One of the words in each curl bracket is mandatory.
Table 3.4

INPUT MAP FOR 'FELUB', (VERSION 3, REV. 1) (Cont'd)

<table>
<thead>
<tr>
<th>CARD TYPE</th>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>DEFINITION</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25-36</td>
<td>CORD(2,N)</td>
<td>Coordinate 2 for node N ( (Y, \theta, Z, \theta) )</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>37-42</td>
<td>NMID</td>
<td>Starting node in previous list</td>
<td>I</td>
</tr>
</tbody>
</table>

Repeat card type E until all nodes are generated. Last card is for \( N = NNODES \). Coordinates 1 and 2 are \( (X,Y) \) for Cartesian, \( (R,\theta) \) for Polar, \( (\theta,Z) \) for Cylindrical, \( (R,\theta) \) for Conical.

<table>
<thead>
<tr>
<th>F 6</th>
<th>ILINE</th>
<th>Element connector</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-12</td>
<td>N</td>
<td>Element number</td>
<td>I</td>
</tr>
<tr>
<td>13-18</td>
<td>NPJTS(1,N)</td>
<td>1st nodal number in element</td>
<td>I</td>
</tr>
<tr>
<td>19-24</td>
<td>NPJTS(2,N)</td>
<td>2nd ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>25-30</td>
<td>NPJTS(3,N)</td>
<td>3rd ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>31-36</td>
<td>NPJTS(4,N)</td>
<td>4th ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>37-42</td>
<td>NPJTS(5,N)</td>
<td>5th ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>43-48</td>
<td>NPJTS(6,N)</td>
<td>6th ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>49-54</td>
<td>NPJTS(7,N)</td>
<td>7th ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>55-60</td>
<td>NPJTS(8,N)</td>
<td>8th ( &quot; )</td>
<td>I</td>
</tr>
<tr>
<td>61-66</td>
<td>NPJTS(9,N)</td>
<td>Number of nodes in element</td>
<td>I</td>
</tr>
<tr>
<td>67-72</td>
<td>JH</td>
<td>Stepper for quadrilaterals</td>
<td>I</td>
</tr>
<tr>
<td>73-78</td>
<td>ITHIK(N)</td>
<td>Clue indicating whether on land or groove</td>
<td>I</td>
</tr>
</tbody>
</table>

Repeat card type F until all elements are generated. ILINE and JH are useful only when quadrilateral elements are used.

<table>
<thead>
<tr>
<th>G 1-6</th>
<th>NBCNDS(1)</th>
<th>Node number N1</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-18</td>
<td>P1(1)</td>
<td>Prescribed pressure at N1</td>
<td>E</td>
</tr>
<tr>
<td>19-24</td>
<td>NBCNDS(2)</td>
<td>Node number N2</td>
<td>I</td>
</tr>
<tr>
<td>25-36</td>
<td>P1(2)</td>
<td>Prescribed pressure at N2</td>
<td>E</td>
</tr>
</tbody>
</table>
Table 3.4

INPUT MAP FOR 'FELUB', (VERSION 3, REV. 1) (Cont'd)

<table>
<thead>
<tr>
<th>CARD TYPE</th>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>DEFINITION</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>37-42</td>
<td>NBCNDS(3)</td>
<td>Node number N3</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>43-54</td>
<td>P1(3)</td>
<td>Prescribed pressure at N3</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>55-60</td>
<td>NBCNDS(4)</td>
<td>Node number N4</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>61-72</td>
<td>P1(4)</td>
<td>Prescribed pressure at N4</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

Repeat card type G until all NBC nodes are listed

| H 1-9 | BISLAM | Viscosity | E |
| 10-18 | HMIN | Typical film thickness | E |
| 19-27 | WIDTH | Typical Bearing dimension | E |
| 28-36 | ØMEGA | Typical frequency | E |
| 37-45 | U1 | Typical speed in 1-direction | E |
| 46-54 | U2 | Typical speed in 2-direction | E |
| 55-63 | PAMB | Ambient pressure | E |

| I 1-6 | NSTEP | Number of computational steps | I |
| 7-18 | DT | Time increment | E |
| 19-30 | TIME | Initial time | E |
| 31-42 | TRUNC | Truncation | E |
| 43-48 | NLØOPS | Number of paths over which flow is to be completed | I |

If NLØOPS = 0, go to card type L

| J 1-6 | N | Flow loop number | I |
| 7-12 | NELM | Number of elements in loop | I |

| K 1-6 | IELM(1) | Element number | I |
| 7-12 | IED(1) | Edge number | I |
| 13-18 | IELM(2) | Element number | I |
| 19-24 | IED(2) | Edge number | I |
Table 3.4
INPUT MAP FOR 'FELUB', (VERSION 3, REV. 1) (Cont'd)

<table>
<thead>
<tr>
<th>CARD TYPE</th>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>DEFINITION</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>25-30</td>
<td>IELM(3)</td>
<td>Element number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>31-36</td>
<td>IED(3)</td>
<td>Edge number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>37-42</td>
<td>IELM(4)</td>
<td>Element number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>43-48</td>
<td>IED(4)</td>
<td>Edge number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>49-54</td>
<td>IELM(5)</td>
<td>Element number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>55-60</td>
<td>IED(5)</td>
<td>Edge number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>61-66</td>
<td>IELM(6)</td>
<td>Element number</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>67-72</td>
<td>IED(6)</td>
<td>Edge number</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

Repeat card type K until NELM elements are listed. Then repeat card types J and K until NLØPS sets are listed.

<table>
<thead>
<tr>
<th>L</th>
<th>I-6</th>
<th>ISENT</th>
<th>Sentinel to signal last problem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 another problem follows</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>≠ 0 last problem</td>
</tr>
</tbody>
</table>

3-17
third words are not significant. The fourth word must be one of the allowed two and the fifth word must also be one of the allowed two. Usage of these words is self-explanatory, the remaining words in card type B are of no significance.

**Card Type C**

This card is also a command card and usage is similar to card type B. The word HYBRID is assumed to denote a bearing in which both hydrostatic and hydrodynamic lubrication take place. Anything other than HYBRID is assumed to be called STANDARD.

**Card Type D**

The variables in this card type are self explanatory. Figure 3.4.1(a) indicates definition of ALPRAD for cylindrical coordinate system and Figure 3.4.1(b), indicates the same for a conical system.

**Card Type E**

This card type inputs the nodal coordinates, these being assumed to be given in the coordinate system specified in card type C. Meaning of coordinates 1 and 2 is assumed to be given by

<table>
<thead>
<tr>
<th>COORDINATE 1</th>
<th>COORDINATE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartesian</td>
<td>X</td>
</tr>
<tr>
<td>Polar</td>
<td>R</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>θ</td>
</tr>
<tr>
<td>Conical</td>
<td>R</td>
</tr>
</tbody>
</table>

This card type allows certain flexibility in use to minimize the number of cards required to describe nodal coordinates.

Ignoring ILINE and NMID for the moment, the nodal input is read sequentially with increasing nodal numbers. If node \( N_1 \) is read in and the next card has node \( N_2 \), where \( N_2 \) is more than one removed from \( N_1 \), then the program assumes that equal increments in coordinates 1
Figure 3.4.1 Cylindrical and Conical Coordinate Systems.

(a) CYLINDRICAL SYSTEM

\[ \text{ALPRAD} = r \]

(b) CONICAL SYSTEM

\[ \text{ALPRAD} = \alpha \]
and 2 are to be provided between nodes \( N_1 \) and \( N_2 \), the incrementing being done in the coordinate system specified in card type C.

If every node is included in the input deck, then they need not be arranged in sequence.

ILINE allows generation of coordinates for a mesh of nodes. Its use is best illustrated by considering an example: suppose a quadrilateral domain with a 3 x 3 grid shown in Fig. 3.4.2(a) is to be generated. In card type E, there need be only four cards as follows: First, a line of nodes 1 to 4 giving their coordinates. ILINE and NMID are set to zero on these cards. Next, a card for node 13 with coordinates, \( \text{ILINE} = 2 \) and \( \text{NMID} = 0 \) is read in. Finally, a card for node 16 with coordinates, \( \text{ILINE} = 0 \), \( \text{NMID} = 0 \) is read in. The program can then generate the coordinates for all the remaining nodes under the assumption that equal increments are used in each coordinate direction. In words, use of ILINE may be defined as follows: If \( N_1 \) is the last node read into core and if the next card has \( \text{ILINE} = m \) then the set of \( m \) nodes to be read in beginning with the node that has \( \text{ILINE} \) associated with it is to be connected to the corresponding set already in core by a mesh. The set of \( m \) nodes read under ILINE control need not be sequential, between each set of nodes that are not sequential, equal increments in coordinates will be assumed.

While ILINE allows a mesh connection between two sets of nodes when there can be no ambiguity in such a connection, NMID allows other cases where such ambiguity arises to be taken care of. Referring to Figure 3.4.2(b), suppose the line of nodes 50 through 56 is already generated. In order to generate the remaining nodes 57 through 72, two cards are necessary, one each for node 69 and node 72. \( \text{ILINE} = 2 \) is specified on the card for node 69 to indicate that a mesh connection is required and \( \text{NMID} = 52 \) on the same card to indicate that the connection begins at node 52 in the nodal set already generated.
(a) ARC OF I_LINE ILLUSTRATED

(b) USE OF NMID ILLUSTRATED

Figure 3.4.2 Use of I_LINE and NMID
The mesh generation feature is intended primarily for use with four point quadrilateral elements. When other elements are used, mesh generation feature of card type E must be used with caution.

Card Type F

This card type inputs the nodal numbers for each element in the assemblage. The sequence in which the nodal numbers are to be listed is shown in Figure 2.5.1, page 2-11 for triangular elements and in Figure 2.5.2, page 2-14 for quadrilateral elements. Element node number 1 can be selected to be any of the vertices, but the sequence for the remaining must follow the indicated sense. A convenient way to keep the sense same throughout, is to look at the element in the direction of the inward normal and pick the sense of rotation of 1-direction into the 2-direction.

Ignoring ILINE and JH for the moment, this card type lists the element number N, the number of nodes in the element and the node numbers. In addition, ITHIK is an integer clue selected by the user to indicate whether the element is in a groove or land. With the understanding that the film thickness is continuous within any element, this clue can be utilized in the user supplied routine FILM to select any of several function types used to describe the film thickness as a function of coordinates.

By means of JH and ILINE this card type allows certain flexibility in use to minimize the number of cards required to describe four point quadrilateral elements. First, the element input is read sequentially with increasing element numbers. If two element numbers, more than 1 removed from each other are read in, then the program fills in the missing element node numbers by incrementing the corresponding previous one in storage by one or two depending on whether JH is zero or non-zero, respectively.
The use of ILINE is slightly different from the previous card type. Here it indicates that the next element number is to be read in and connected to the previous set of elements in memory. The card immediately following the card with non-zero ILINE need not list the node numbers and ILINE is significant only as being either equal to zero or not equal to zero. For internally generated elements, the program sets ITHIK to be equal to the last one read in.

Card Type G

This card type lists the boundary conditions. Description of the variables in the input map is sufficiently self-explanatory.

Card Type H

This card type specifies the problem parameters. Where a typical frequency does not exist, it may be set equal to 1. \( U_1 \) and \( U_2 \) are maximum speeds in the 1 and 2 directions of the coordinate system selected, and have the following meaning:

- Cartesian
  \( U_1 \) - max. lineal speed in x-direction \((LT^{-1})\)
  \( U_2 \) - max. lineal speed in y-direction \((LT^{-1})\)

- Polar
  \( U_1 \) - not defined
  \( U_2 \) - max. angular velocity \((\text{Rad}.T^{-1})\)

- Cylindrical
  \( U_1 \) - max. angular velocity \((\text{Rad}.T^{-1})\)
  \( U_2 \) - max. lineal speed in z-direction \((LT^{-1})\)

- Conical
  \( U_1 \) - not defined
  \( U_2 \) - max. angular velocity \((\text{Rad}.T^{-1})\)

The remaining variables in this card type are self-explanatory.
Card Type I

The variables for this card type are self-explanatory.

Card Type J

A flow loop is defined as a continuous line joining several nodes along element edges. The line is assumed not to cross any of the elements. The elements to be listed are assumed to be on one side or the other of the line as one travels along the line. The sign of the computed flow is determined by which side is selected. If NLPS is set to zero in card type I, card type J should be omitted.

Card Type K

Edge number is defined by the element nodes that it connects as follows:

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Edge 1</th>
<th>Edge 2</th>
<th>Edge 3</th>
<th>Edge 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3PT triangle</td>
<td>1-2</td>
<td>2-3</td>
<td>3-4</td>
<td>---</td>
</tr>
<tr>
<td>6PT triangle</td>
<td>1-2</td>
<td>2-3</td>
<td>3-4</td>
<td>---</td>
</tr>
<tr>
<td>4PT quadrilateral</td>
<td>1-2</td>
<td>2-3</td>
<td>3-4</td>
<td>4-1</td>
</tr>
<tr>
<td>8PT quadrilateral</td>
<td>1-2</td>
<td>2-3</td>
<td>3-4</td>
<td>4-1</td>
</tr>
</tbody>
</table>

If NLPS in card type I is set to zero, this card type should be omitted.

Card Type L

This is a sentinel indicating whether more problems follow.

The user in addition to an input deck must supply a subroutine that defines film thickness as a function of coordinates and time. Calling sequence is

CALL FILM (x1, x2, TIME, IFILM, H, HDOT)

x1, x2, TIME, and IFILM are inputs to the subroutine and, H and HDOT are outputs of the routine. x1 and x2 are coordinates in the user specified system in card type C. TIME is current time and IFILM is the thickness clue as defined by the user in card type F.
If effects of changing the bearing speed are to be studied, the SPEED routine also must be supplied by the user. The calling sequence is

```
CALL SPEED (TIME, PLA, DPLA)
```

where TIME is current time, PLA is the current fraction of maximum speed specified in card type H and DPLA is the expected change in PLA during the time increment DT specified in card type I.

If the user wants to automate mesh generation for particular bearing configurations, subroutines may be generated for each of them. The calling sequence is

```
CALL AUTGRD
```

The routine must fill the COMMON/MESH/, COMMON/BOUNDR/ and ALPRAD in COMMON/TIMES/. It may use /EQUATN/ as scratch storage.

### 3.5 Validation

The transient lubrication problem defined by equations 2.1,5, has an exact, though trivial, one-dimensional solution. While the solution is of little consequence, in so far as engineering applications are concerned, it is useful for validation and studies of the effects of discretization and truncation.

Consider the initial and boundary conditions given by

\[
P(x,0) = (\alpha + \lambda) e^{2\alpha x} / 2\alpha \\
P(0,T) = (\alpha + \lambda) e^{2\alpha T} / 2\alpha \\
P(1,T) = (\alpha + \lambda) e^{2\alpha (1+T)} / 2\alpha
\]

where \(x\) is the one dimensional space variable ranging over \(0 \leq x \leq 1\), \(T\) is time ranging over \(0 \leq T\) and \(\alpha\) is a non-negative quantity. If the film thickness \(H\) is given by

\[
H = e^{-\alpha (x+T)}
\]

then

\[
P = (\alpha + \lambda) e^{2\alpha (x+T)} / 2\alpha
\]

3-25
Figure 3.5.1 Comparison of Transient Solutions, Computed and Exact
Figure 3.5.2 Steady State Solution for One-Dimensional Slider Bearing, $\Lambda = 10$

$H_1 = H_2 = 1$

$H_1/H_2 = 3$

$P_{\text{MAX}} = 1.40296$
Figure 3.5.3 Steady State Solution for One-Dimensional Slider Bearing, $\Lambda = 50$
is a solution to equation 2.1.5, which can be readily verified by substituting equation 3.5.5 into 2.1.5.

Referring to Fig. 3.5.1, solution for a problem with 4 spatial increments and \( a = 1, \lambda = 1, \sigma = 1 \), obtained by FELUB is illustrated. With \( \Delta T = .01 \), the computed solution is found to differ negligibly from the exact solution given by equation 3.5.5.

Fig. 3.5.2 illustrates the steady state solution for a one dimensional slider bearing obtained with 20 finite elements, \( \Lambda \) being kept constant at 10. An exact solution for this case is given by Harrison [8]. With \( \Delta T = -.1 \), and 15 steps, the computed pressure profile is found to differ negligibly from the exact solution. This may be compared with a previous solution [3] which required 20 steps for obtaining the same accuracy. The presence of \( \sigma (PH)_T \) in the present solution seems to be the major factor in obtaining the accelerated convergence.

Referring to Fig. 3.5.3, solution for the same slider bearing but with \( \Lambda = 50 \) and 60 finite elements is shown. With 20 increments, the solution is within the requirements of engineering accuracy. The solutions shown for 5 and 10 increments are seen, respectively, to be below and above the true solution. This is a stationary rather than extremal. In the incompressible case, which is an extremal, the solution always converge from below.

### 3.6 Shrouded Step Bearing

As an example of a problem with complex geometry, the steady-state pressure distribution for the shrouded step bearing is considered.

An AUTGRD routine was prepared to generate the grid for both rectangular and polar configurations of the bearing. This routine supplies card types D, E, F and G of the input maps given in Table 3.4.

Definition of the variables used in the routine are shown in Fig. 3.6.1.
Figure 3.6.1 Shrouded Step Bearing Configuration

\[ \frac{BW}{BL} = BW_{OB} \]
\[ \frac{SW}{BW} = SW_{OB} \]
\[ \frac{BS}{BL} = BS_{OB} \]

\[ \frac{RI}{RO} = R_{OR} \]
\[ \frac{\theta_S}{\theta} = BS_{OB} \]
\[ \frac{SW}{BW} = SW_{OB} \]
Figure 3.6.2 Polar, Shrouded Step Bearing Mesh
Figure 3.6.3 Rectangular Shrouded Step Bearing Mesh
CALCOMP plots of the mesh layout for typical cases in polar and rectangular configurations are shown in Fig. 3.6.2 and Fig. 3.6.3 respectively.

The particular case of a pad with the geometry shown in Fig. 3.6.4 was analyzed for an application with the velocity field shown in Fig. 3.6.5. The resulting steady-state solution obtained by FELUB is shown in Fig. 3.6.6 as isobars. The plot was made by a CALCOMP, 'GPCP' package.
Figure 3.6.4 Pad Geometry
Figure 3.6.5 Polar Velocity Field
Figure 3.6.6 Steady State Pressure Distribution for Rectangular Shrouded Step Bearing
4. CONCLUSIONS

The overall objective of developing a general purpose, finite element lubrication program has been accomplished in the main.

Validity of the transient, variational formulation is established.

The preliminary experience with the program indicates that the transient procedure is extremely stable.

The transient procedure is a more economical way to solve the steady-state problem than that reported in reference [3].
5. REFERENCES


