AC-CR-75-005

OPTIMIZATION OF SURFACE CONTOURS FOR ELASTIC BODIES IN CONTACT

by T. S. ARORA and J. S. ARORA

MAY 1975

TECHNICAL REPORT FOR JULY 1974 TO APRIL 1975

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

prepared by:
Division of Materials Engineering
College of Engineering
University of Iowa
Iowa City, Iowa 52242

Contract No.:
DAA09-74-M-2020

prepared for:
US Army Armament Command
Rock Island, Illinois 61201

Technical Report No. 17

RI-CR-75-005

OPTIMIZATION OF SURFACE CONTOURS
FOR ELASTIC BODIES IN CONTACT

by

T. S. Arora and J. S. Arora

May 1975

Project Title: MINIMUM STRESS CONTOUR DESIGN

Contract No.: DAAA09-74-M-2020

This report is the M. S. thesis of T. S. Arora done under the supervision of Professor Edward J. Haug, Jr.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 17	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): OPTIMIZATION OF SURFACE CONTOURS FOR ELASTIC BODIES IN CONTACT	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report July 1974 to April 1975
	6. PERFORMING ORG. REPORT NUMBER: 17
7. AUTHOR(s): T. S. Arora and J. S. Arora	8. CONTRACT OR GRANT NUMBER(s): DAAA09-74-M-2020
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Division of Materials Engineering, College of Engineering, The University of Iowa, Iowa City, Iowa 52242	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Armament Command, AMSAR-RDT, Rock Island, Illinois 61201	12. REPORT DATE: May 1975
	13. NUMBER OF PAGES: 99
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassifed
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: NA

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Contact Stress, Surface Design, Minimum Stress, Linear Programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this report, design of surface contours of contacting elastic bodies to minimize the peak stress is considered. The initial contacting arc for the bodies is obtained by analyzing the problem through quadratic programming. The problem of minimizing the peak contact stress is then posed as a sequential linear program (LP). Convergence to the optimal solution is proved by the fact that LP solutions form a non-increasing monotone sequence which is bounded below by zero. Effectiveness of the technique is demonstrated for the problem of beam on an elastic foundation for which the peak stress is reduced considerably.

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

LIST OF TABLES

LIST OF SYMBOLS

Symbol	
B	Matrix formed from influence coefficients of the two bodies.
A	Affine transformation accounting for rigid body displacement of Body 1.
a	Constant vector that depends on the externally applied loads to the two bodies and the initial gap between them at the potential contact region.
b	The vector of contour modifications.
C	The external load column vector for externally applied loads.
ε	Final gap size vector.
t	Total applied load.
w	Work done.
S	Contact stress vector for potential contact points.
J	Cost function.
b_{n+1}	Upper bound on contact stresses
P	Projection matrix = $[0 \mid I_{n-m}]$
I_{n-m}	An identity matrix of dimension (n-m).
t^1	External force vector on Body 1.
$\hat{I}$	Index set for points in contact.
$\tilde{I}$	Index set for points not in contact.
NC	Number of points in contact

Symbol	
NNC	Number of points not in contact.
b^o	Lower bound on contour modification.
b^1	Upper bound on contour modification.
H	Matrix that gives rigid body displacement at the points of application of t^1.
X_i	X-coordinate of ith point.
q	Rigid body coordinate vector for Body 1.
d_i	Initial gap between ith potential contact points.
n	Number of potential contact points.
m	Number of degrees of freedom.

Chapter I

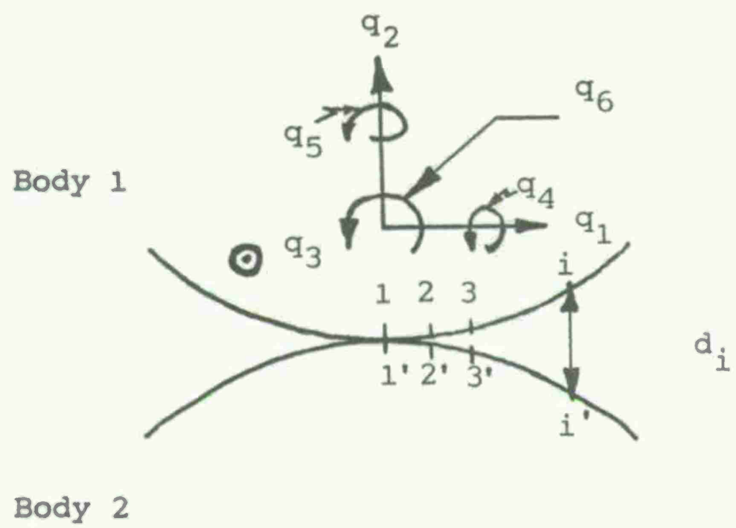
INTRODUCTION

Many problems in continuum mechanics and mechanical system design involve elastic bodies that come into contact with each other, under applied loads. Such problems, called contact problems of elasticity, are nonclassical in the sense that one does not initially know the contacting region. Considerable research has been pursued in recent years to develop constructive methods of determining the contact region and contact stress distribution [1-4].

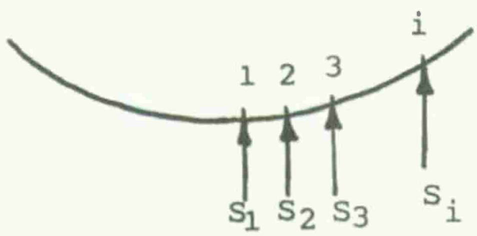
The geometry of two bodies that will come into contact under applied load is shown schematically in Figure 1(a). A numerical method of solution, by quadratic programming [5] has been developed, in which one selects a set of potential contact points, as indicated in Figure 1(a). The selection of potential contact points is explained in detail in [6]. The initial gap between the ith potential contact points is denoted by d_i. The rigid body displacement coordinates of Body One are indicated in Figure 1(a) and are represented by a generalized coordinate vector q, of dimension one to six. Once the potential contact points are defined on the two bodies, contact stress may be replaced by equivalent point loads, or contact forces, denoted as S_i in Figure 1(b).

When the bodies come into contact, a force distribution over the surfaces arises and high contact stresses may occur over some parts of the contact region. This is undesirable, since plastic deformation of the bodies may occur, or high normal forces may lead to wear of machine parts that move relative to each other. This can be quite important in precision machines. The objective of this work is to develop a technique for adjusting the contour of one or both of the bodies in order to achieve a minimum peak contact stress between the bodies, under a given load.

A related problem was solved by Conry [6], in which the design objective was to select the contour to achieve a constant stress over the contact arc. A linear programming method was presented in [7,8] to implement this design objective. In general, one cannot achieve a constant stress distribution over the contact arc when kinematic constraints define limits on modification of the contours of the bodies. This will be particularly critical in precision cams and fasteners that require precise location of parts. The design objective selected in this work is to minimize the peak contact stress, subject to constraints on the extent of modification of surface contours.



1(a).



1(b).

Figure 1. Geometry of Contacting Bodies.

Chapter II

THE UNBONDED CONTACT PROBLEM FOR ELASTIC BODIES IN CONTACT

2.1 Analytical Formulation of the Problem

The elastic contact problem formulation of [6] will be employed as the model of the elastic contact problem. Only a brief summary of this analytical formulation will be presented here. The gap variable ε, due to a stress distribution S, is given by the equation

$$\varepsilon = BS + Aq + a + b \qquad (2.1.1)$$

where B is a matrix formed from influence coefficients of the two bodies, A is an affine transformation accounting for rigid body displacement of Body 1, a is a constant vector that depends on the externally applied loads to the two bodies and the initial gap between them at the potential contact region, and b is the vector of contour modification to be selected. Formulae for B, A, and a are given in Appendix A.

Equilibrium of Body 1 is obtained through direct application of the principle of virtual work, which results in the linear equation

$$A^TS = C \qquad (2.1.2)$$

where the column vector C depends on the externally applied load, and is given in Appendix A.

Compatibility conditions between the two contacting bodies require that the product of each gap variable and contact stress be zero. Analytically, this is

$$\varepsilon_i S_i = 0 \, , \; i = 1,2, ----, n \qquad (2.1.3)$$

where n is the number of potential contact points. This condition may be interpreted as stating that either the gap or the contact stress must be zero at each point on the contacting bodies. Further, the gap and contact stress must be non-negative. Analytically these conditions are

$$S_i \geq 0, \; \varepsilon_i \geq 0, \; i = 1,2, ----, n \qquad (2.1.4)$$

As is shown in [6], Equations (2.1.1)-(2.1.4) are the Kuhn-Tucker necessary conditions for solution of a convex quadratic programming problem, which may be stated as

$$\left.\begin{array}{ll} \text{minimize} & A^TS + \frac{1}{2} S^TBS \\ \text{subject to constraints} & A^TS = C \\ & S \geq 0 \end{array}\right\} \qquad (2.1.5)$$

This problem may be solved through direct application of the simplex technique of quadratic programming [5]. It is also shown in [5] that if a solution exists, it is unique and may be reliably determined, through application of a simplex technique for quadratic programming [6].

The cost function to be minimized in this design problem is the maximum contact stress,

$$J = \max_i S_i \qquad (2.1.6)$$

Since this maximum value function is difficult to treat analytically, it may be replaced by defining an upper bound b_{n+1} on the contact stress, through the set of inequalities

$$\Phi_i = S_i - b_{n+1} \leq 0 \qquad (2.1.7)$$

The cost function J of equation (2.1.6) may now be replaced by an equivalent problem of minimizing the upper bound on contact stress

$$\bar{J} = b_{n+1} \qquad (2.1.8)$$

subject to constraints (2.1.7).

Finally, it is often required that both upper and lower bounds be placed on modifications of the surface contours of the two bodies. Analytically, these conditions may be stated as

$$\Phi_{n+i} = b^{o} - b_i \leqslant 0 \text{ , } i = 1, ----, n \qquad (2.1.9)$$

and

$$\Phi_{2n+i} = b_i - b^{1} \leqslant 0 \text{ , } i = 1, ----, n \qquad (2.1.10)$$

where b^o and b^1 are the lower and upper bounds on contour modification, respectively.

The design problem may now be stated as follows: Determine the design variable vector b and the upper bound b_{n+1} to minimize $\bar{J}$, subject to constraints (2.1.1) - (2.1.4) , (2.1.7), (2.1.9) and (2.1.10).

2.2 Sequential Linear Programming Solution Technique

With an initial estimate of the contour (normally b = 0), one may solve the contact-analysis problem of equations (2.1.1) - (2.1.4) to obtain the contact arc and contact stresses. Using this solution, define the index sets

$$\hat{I} = \left\{ i_1, i_2, -----, i_{NC} \right\}$$

of points in contact and

$$\tilde{I} = \left\{ j_1, j_2, -----, j_{NNC} \right\}$$

of points not in contact, where NNC is the number of points not in contact. The stresses at points in contact will be denoted as the NC dimensional vector $\hat{S}$.

Define $\hat{B}$ as the matrix B with rows and columns corresponding to points not in contact removed, $\hat{A}$ as the matrix A with rows corresponding to the points not in contact removed, and the vectors $\hat{a}$ and $\hat{b}$ as the vectors a and b with components corresponding to points not in contact removed.

One may now state the set of conditions that must be satisfied if the contact arc is not changed; namely

$$\hat{\varepsilon} \equiv \hat{B}\hat{S} + \hat{A}q + \hat{a} + \hat{b} = 0 \qquad (2.2.1)$$

The equilibrium equations (2.1.2) must still be satisfied with contact loads applied only on the fixed contact region, so one obtains

$$\hat{A}^T \hat{S} = C \qquad (2.2.2)$$

The objective now is to determine the reduced design variable $\hat{b}$ so that peak contact stresses are reduced as much as possible, keeping the same contact region and satisfying conditions (2.1.7) - 2.1.10).

The gap variable outside the prescribed contact region must remain non-negative, so

$$\tilde{\varepsilon} \equiv \tilde{B}\hat{S} + \tilde{A}q + \tilde{a} + \tilde{b} \geqslant 0 \qquad (2.2.3)$$

where $\tilde{B}$ is the matrix B with columns corresponding to points not in contact and rows corresponding to the points in contact removed, $\tilde{A}$ is the matrix A with rows corresponding to points in contact removed, and vectors $\tilde{a}$ and $\tilde{b}$ are the vectors a and b with rows corresponding to points in contact removed. Here, the reduced vector $\tilde{b}$ represents the estimates of the contour design variable from the preceding iteration. Finally, it is necessary to impose the condition,

$$\hat{S} \geqslant 0 \qquad (2.2.4)$$

The design modification problem is now to determine the reduced vector $\hat{b}$ and b_{n+1} to minimize

$$\bar{J} = b_{n+1}$$

subject to: (2.2.5)

$$\hat{S} \leq b_{n+1} \qquad (2.2.5, con'd.)$$

$$b^{o} - \hat{b} \leq 0$$

$$\hat{b} - b^{1} \leq 0$$

$$\hat{B}\hat{S} + \hat{A}q + \hat{a} + \hat{b} = 0$$

$$\hat{A}^{T}\hat{S} = C$$

$$\tilde{\mathcal{E}} = \tilde{B}\hat{S} + \tilde{A}q + \tilde{a} + \tilde{b} \geq 0$$

$$\hat{S} \geq 0$$

This formulation for determining the modified contour variable is simply a linear programming problem in the variables $\hat{s}$, q, and $\hat{b}$ which can be solved with standard linear programming codes.

Following each solution of the linear programming problem, the contact stresses and gap variables must be evaluated to determine the contact surface to be employed in the next iteration. Any potential contact point for which $\tilde{\mathcal{E}}_j = 0$ is included in the contact surface for the next iteration. At any points in the preceding potential contact surface for which $\hat{S}_j = 0$, it is presumed that separation will occur in the subsequent iteration, so these points are deleted from the vectors $\hat{\mathcal{E}}$ and $\hat{S}$. These modification rules form the basis for definition of the next linear programming problem to be solved for the optimum contour on an adjusted contact arc. This process is continued until no new points come into contact and no points that were previously in contact are released.

Each linear programming solution will reduce the value of the peak stress, until the process stops. Analytically, this condition can be written as follows

$$\left.\begin{array}{l} b_{n+1}^{(o)} \geqslant b_{n+1}^{(1)} \\ b_{n+1}^{(1)} \geqslant b_{n+1}^{(2)} \\ \quad\vdots \\ b^{(k-1)} \geqslant b_{n+1}^{(k)} \end{array}\right\} \qquad (2.2.6)$$

where the superscript on b_{n+1} denotes the iteration number. Combining (2.2.6) one obtains

$$b_{n+1}^{(o)} \geqslant b_{n+1}^{(1)} \geqslant \text{------} \geqslant b_{n+1}^{(k)} \qquad (2.2.7)$$

This gives a sequence of non-increasing real numbers, called a monotone sequence, that is bounded below by zero. Any non-increasing sequence that is bounded below is convergent [9], so the sequence of solutions of linear programming problems (2.2.5) must converge.

It was observed from preliminary calculations that as the value of U is increased, peak stress decreases, but a stage is reached when stress at all the points in the contact region is the same. The peak stress obtained in this case is a local minimum and no contour design variable reaches its allowable limit. This situation can be avoided by adding a penalty function to the cost function that is intended to broaden the contact region. The penalty function used here is as follows

$$J = b_{n+1} + \delta_1 \sum_{j=1}^{NNC} \tilde{\varepsilon}_j \qquad (2.2.8)$$

where δ_1 is a small constant greater than zero. The linear programming with the augmented cost function (2.2.8) finds an absolute minimum of the peak contact stress.

The process described above is summarized in the following algorithm:

Step 1. Estimate the design variables (normally $b^{(o)} = 0$).

Step 2. Solve for $\hat{S}$ and ε by quadratic programming, using the necessary conditions (2.1.1)-(2.1.4).

Step 3. Form index sets $\hat{I}$ and $\tilde{I}$ for points of contact ($S_i \geqslant 0$, $\varepsilon_i = 0$) and for points not in contact ($\varepsilon_j \geqslant 0$, $S_j = 0$), respectively.

Step 4. Construct $\hat{B}$, $\hat{A}$, $\hat{a}$, and $\hat{b}$ for $i \in I$ and A, B, a, and b for $j \in I$.

Step 5. Solve the linear programming problem (2.2.5) for $\hat{S}$, q, and $\hat{b}$. (The cost function may be changed as given in (2.2.8) if necessary).

Step 6. Evaluate $\tilde{\varepsilon}$ from (2.2.3).

Step 7. From new $\hat{I}$ to include points for which $\varepsilon_j = 0$ and deleting points for which $S_j = 0$ and form $\tilde{I}$ of points for which $\varepsilon_j \geqslant 0$ and $S_j = 0$.

Step 8. If $\hat{I}$ is unchanged, terminate; otherwise return to Step 4.

Chapter III

CONTACT SURFACE DESIGN PROBLEMS FOR BEAMS ON ELASTIC FOUNDATIONS

Examples are presented to demonstrate the technique developed in Chapter II. Two cases of the design problem with a beam on an elastic foundation are considered: (1) with no initial gap; and (2) with an initial gap.

A point load is applied to the center of the beam, as shown in Figure 2(b). Only vertical displacement of the center of the beam is chosen to represent its rigid body displacement. Formulation of the matrices B, A, a, b, and C for this problem is explained in detail in [2]. A quadratic programming problem is solved to obtain the contact arc as input to the linear programming problem.

The design variables b_i are limited by the constraint $b_i \leq U\,(1 - \frac{x_i^2}{L^2})$, where x_i is the horizontal distance of the ith potential contact point from the center of the beam, L is half the potential contact length (as only half of the beam is considered due to symmetry), and U is a constant greater than zero.

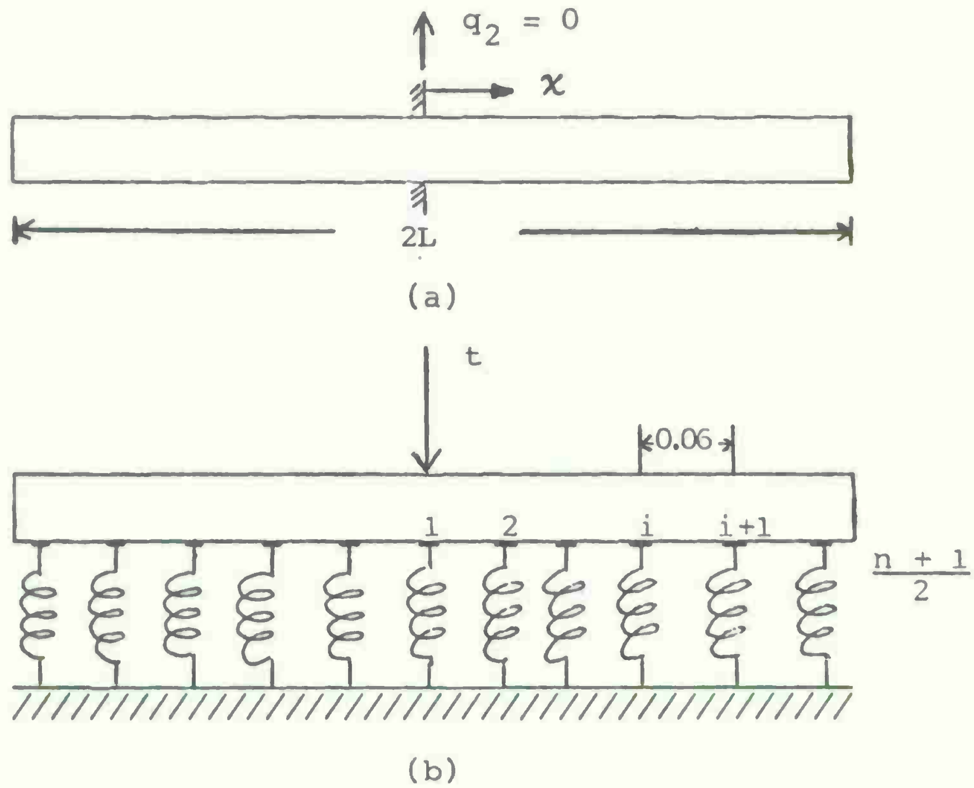


Figure 2. Potential Contact Points for the Beam on Elastic Foundation.

3.1 Beam on Elastic Foundation with No Initial Gap

The following material properties are used: Young's modulus of the beam material = 340,000 psi; diagonal element of the flexibility matrix for the elastic foundation = 0.00005 in./lb.; height of the beam = 0.5 in.; and width of the beam = 1.0 in.

The quadratic programming problem was solved for 25 potential contact points on half of the beam, with an interval size of 0.06 in. Potential contact length, as shown in Figure 3(a) is taken as 2.88 in. The contact length obtained numerically was 1.38 in., with loads of 1000 lbs. and 2000 lbs. A quadratic programming code called ZORILLA [10] was used to solve this quadratic programming problem. Results are presented in Figure 3(b).

At this stage, one has the contact arc to be employed in the linear programming problem of (2.2.5). Equations (2.2.5) can now be written as:

$$
\begin{aligned}
&\bar{J} = b_{n+1} \\
&\hat{S} - b_{n+1} \leq 0 \\
&\hat{b} \geq b^{o} \\
&\hat{b} \leq b^{1} \\
&\hat{B}\hat{S} + \hat{A}q + \hat{b} = -\hat{a} \\
&\hat{A}^{T}\hat{S} = C \\
&\tilde{B}\hat{S} + \tilde{A}q \geq -\tilde{a}-\tilde{b} \\
&\hat{S} \geq 0
\end{aligned}
\qquad (3.1.1)
$$

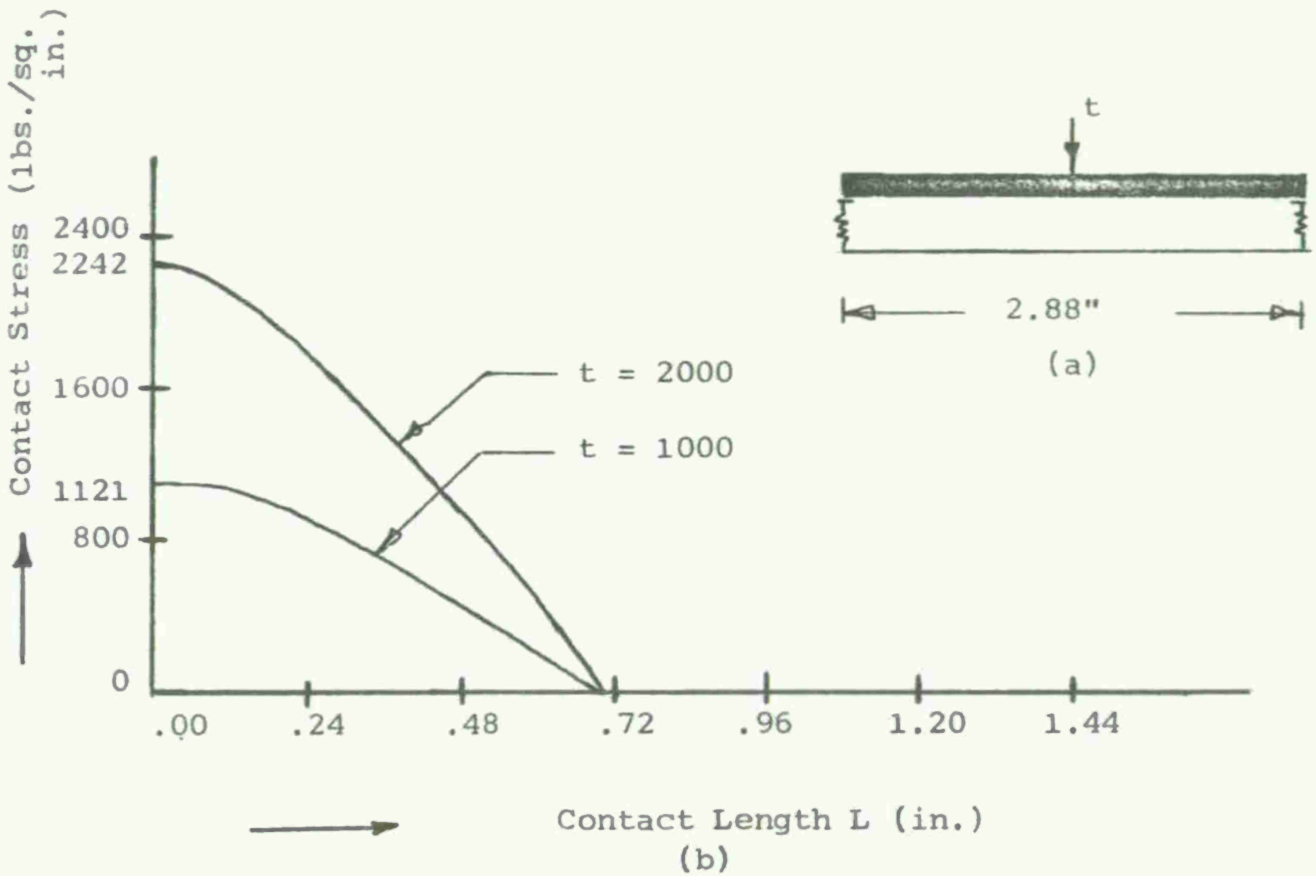


Figure 3. Contact Stress vs. Contact Length from Quadratic Programming Solution.

The linear programming subroutine used to solve (3.1.1) is explained in [11] and is summarized in Appendix B. This subroutine requires that the right hand side values of constraints must be given as a separate vector, all elements of which must be positive. If $b^o = -U[1-\frac{x^2}{L^2}]$ and $b^1 = U[1 - \frac{x^2}{L^2}]$, then Equations (3.1.1) can be re-written as

$$\left.\begin{array}{l} \bar{J} = b_{n+1} \\ \hat{S} - b_{n+1} \leq 0 \\ -\hat{b} \qquad\qquad \leq -b^o \\ \hat{b} \qquad\qquad \leq b^1 \\ \hat{B}\hat{S} + \hat{A}q + \hat{b} = 0, \text{ as } \hat{a} = 0 \\ \hat{A}^T \hat{S} = C \\ -\tilde{B}\hat{S} - \tilde{A}q \leq \tilde{a} + \tilde{b} \end{array}\right\} \qquad (3.1.2)$$

The linear programming subroutine used solves only maximization problems, so the cost function is replaced by $J = -b_{n+1}$. Since q_r can be positive or negative, the vector q is decomposed into positive and negative parts as follows:

$$\begin{array}{llll} q & = q^+ - q^- & & \\ q^+ & = q & \text{when } q \geq 0 & \\ & = 0 & \quad q \leq 0 & (3.1.3) \\ q^- & = q & \text{when } q \leq 0 & \\ & = 0 & \quad q \geq 0 & \end{array}$$

Now an LP tableau is formed as follows:

$$
\begin{array}{c}
\\ 1 \\ NC \\ NC \\ NC \\ NC \\ m \\ NNC
\end{array}
\begin{array}{c}
\begin{array}{ccccc} m & m & 1 & NC & NC \end{array} \\
\left[\begin{array}{c:c:c:c:c}
0 & 0 & -1 & 0 & 0 \\ \hdashline
0 & 0 & -I & 0 & I \\ \hdashline
0 & 0 & 0 & I & 0 \\ \hdashline
0 & 0 & 0 & -I & 0 \\ \hdashline
\hat{A} & -\hat{A} & 0 & I & \hat{B} \\ \hdashline
0 & 0 & 0 & 0 & \hat{A} \\ \hdashline
-\breve{A} & \breve{A} & 0 & 0 & \breve{B}
\end{array}\right]
\end{array}
\left[\begin{array}{c}
q^{+} \\ q^{-} \\ b_{n+1} \\ \hat{b} \\ \hat{S}
\end{array}\right]
=
\left[\begin{array}{c}
\\ \hdashline
0 \\ \hdashline
-b^{0} \\ \hdashline
b^{1} \\ \hdashline
0 \\ \hdashline
C \\ \hdashline
\breve{a}+\breve{b}
\end{array}\right]
\qquad (3.1.4)
$$

where the vector NBP $[\ 0 \ \vdots\ 0 \ \vdots\ -1 \ \vdots\ 0 \ \vdots\ 0\ ]$ of dimension $(1,\ 2m + 1 + 2 * NC + NGE)$ represents the cost function and NGE is the number of constraints of the form $\geqslant 0$, and m is the number of degrees of freedom. The Matrix $\bar{B}$, defined as

$$
\bar{B} \equiv
\begin{array}{c}
\\ NC \\ NC \\ NC \\ NC \\ m \\ NNC
\end{array}
\begin{array}{c}
\begin{array}{ccccc}
m & m & 1 & NC & NC
\end{array}
\\
\left[\begin{array}{c:c:c:c:c}
0 & 0 & -I & 0 & I \\ \hdashline
0 & 0 & 0 & I & 0 \\ \hdashline
0 & 0 & 0 & -I & 0 \\ \hdashline
\hat{A} & -\hat{A} & 0 & I & \hat{B} \\ \hdashline
0 & 0 & 0 & 0 & \hat{A} \\ \hdashline
-\tilde{A} & \tilde{A} & 0 & 0 & \tilde{B}
\end{array}\right]
\end{array}
$$

contains constraint coefficients. Its dimension is (4 * NC + m + NNC , 2 * m + 1 + 2 * NC + NGE) and the vector $RQ^T = [\ 0 \ \vdots \ -b^o \ \vdots \ b^1 \ \vdots \ o \ \vdots \ c \ \vdots \ \tilde{a}+\tilde{b}\ ]$, of dimension (1, 4 * NC + m + NNC), is the vector of right hand side values of the constraints.

The design problem was solved with loads t = 1000 lbs. and t = 2000 lbs., with a range of values of contour modification limit U. Results are presented in Figures (4 - 9). All the calculations were done on an IBM 360/65 computer. Table 1 shows the comparison of results for different loads and values of U ranging from 0.01 to 0.05. It is observed that the contact length for the optimum design, with a fixed value of U, varies with applied load. However, when the value of U is changed in the same proportion as the loads, it is noted that the contact length is the same for both cases (and peak contact stress are proportional). As the value of U is increased, the contact length appears to increase and the value of peak stress decreases. A stage is reached when stress at each point in the contact region is the same, which stops the iterative process. In the case of U = 0.025, t = 1000 lbs., and U = 0.05, t = 2000 lbs., no contour design variable's constraint reaches its tolerance limit. The cost function was augmented with penalty function and the linear programming problem was solved for the above case. The results are presented in Figures (8 - 9). It is also observed that the contour design variable constraint at the center point of the contact region is always tight.

Table 1 presents peak stress, computing time and contact lengths for several numerical examples. Solutions of the linear programming problem converge in at most 8

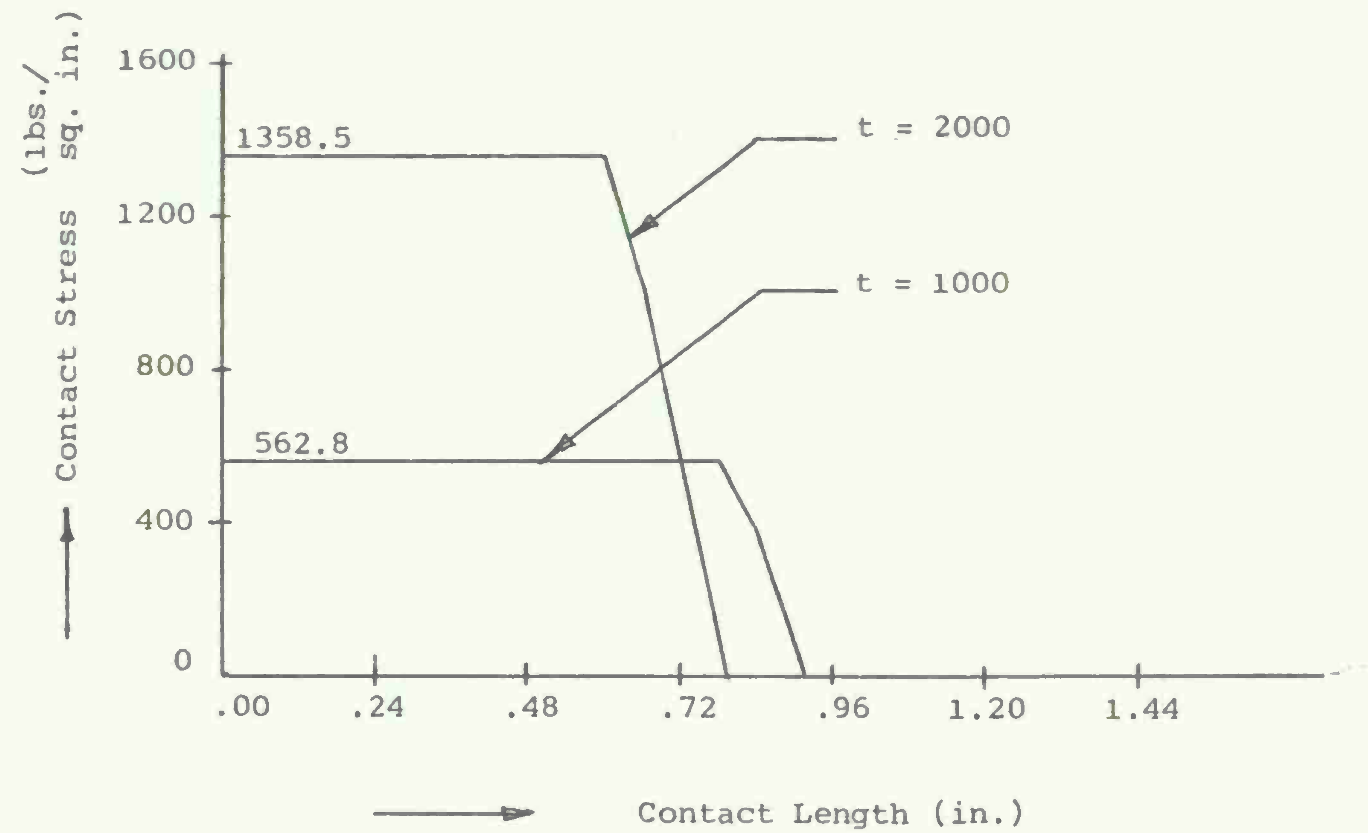


Figure 4. Contact Stress vs. Contact Length with U = 0.01.

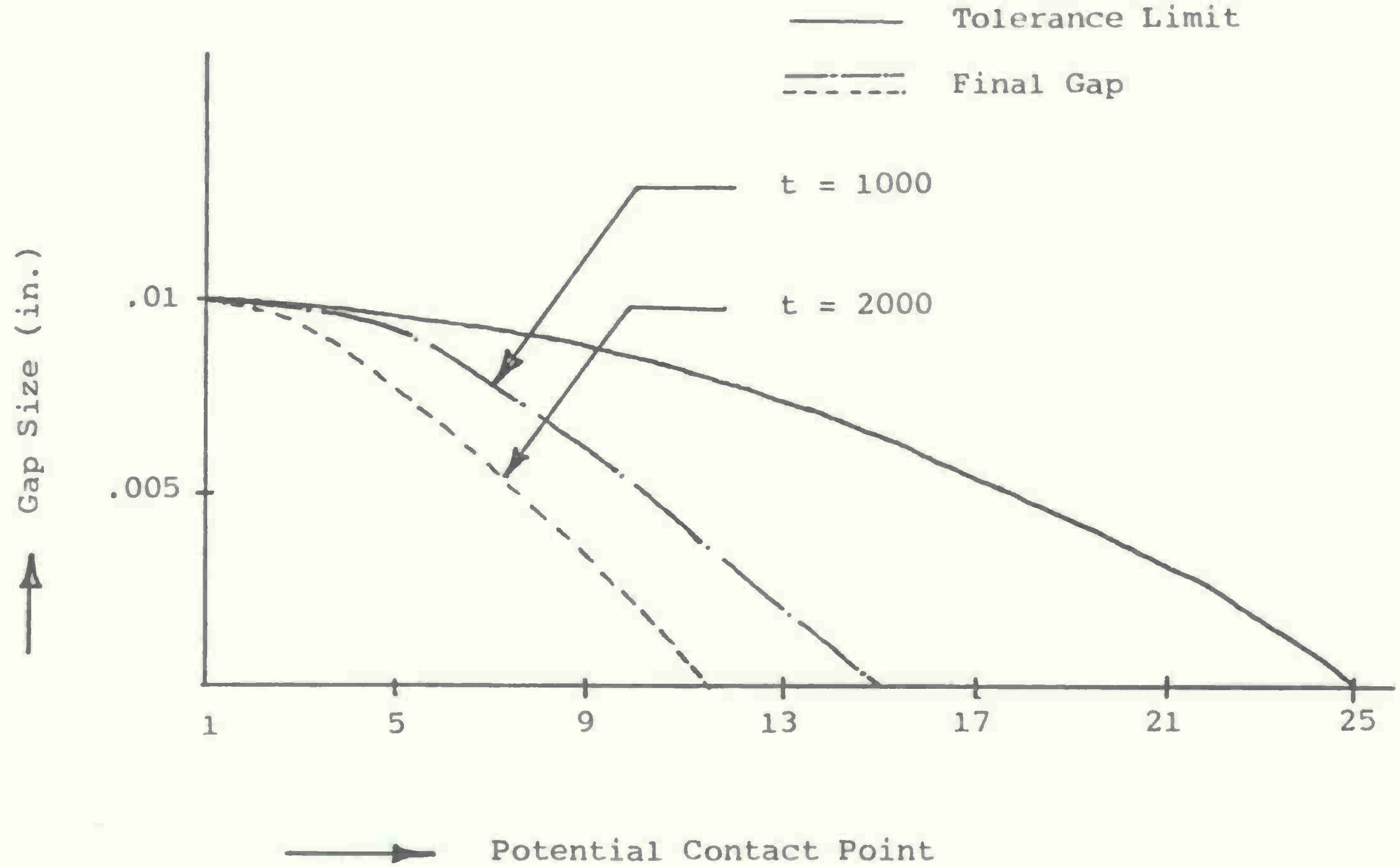


Figure 5. Gap Size vs. Potential Contact Point with U = 0.01.

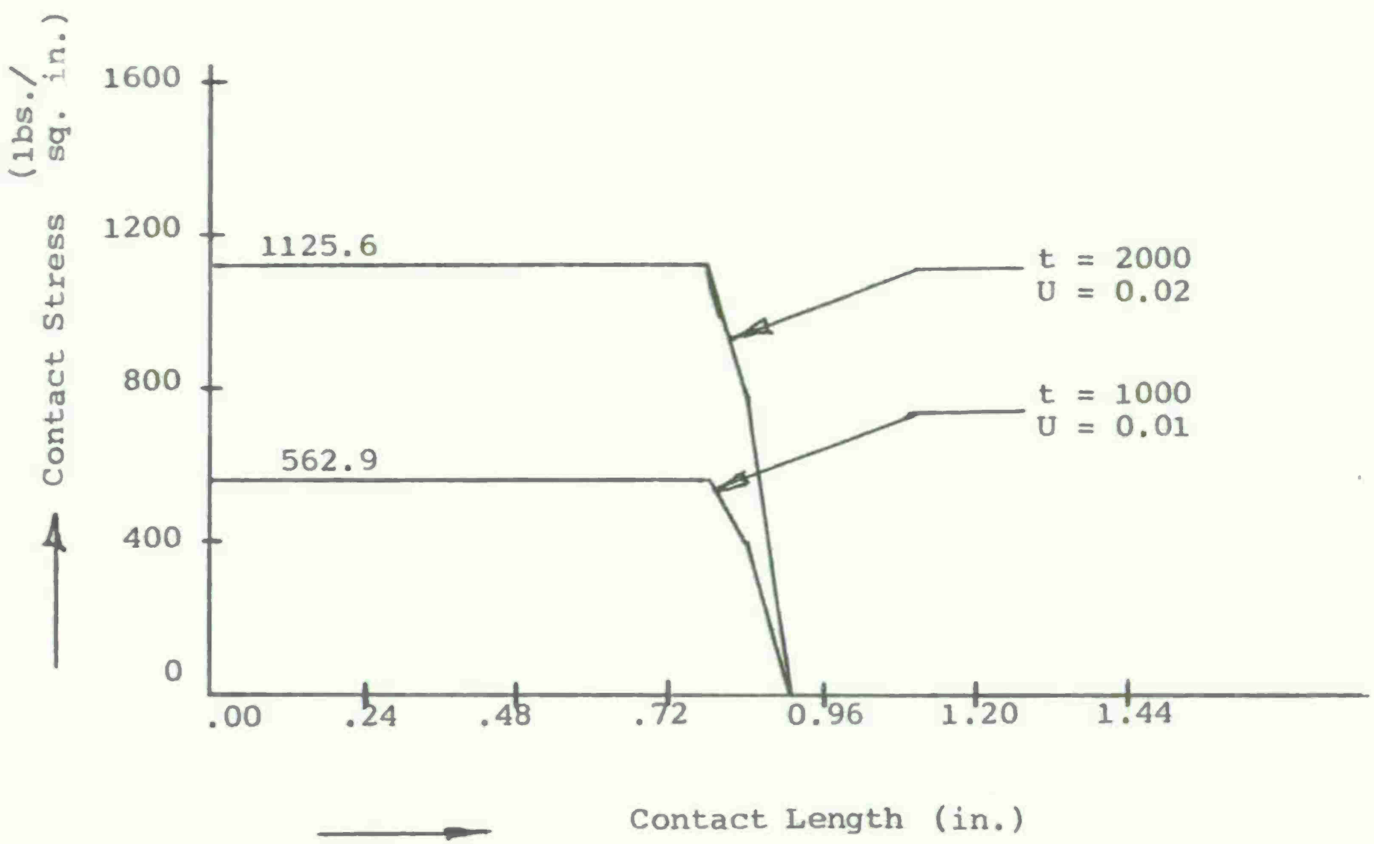


Figure 6. Contact Stress vs. Contact Length.

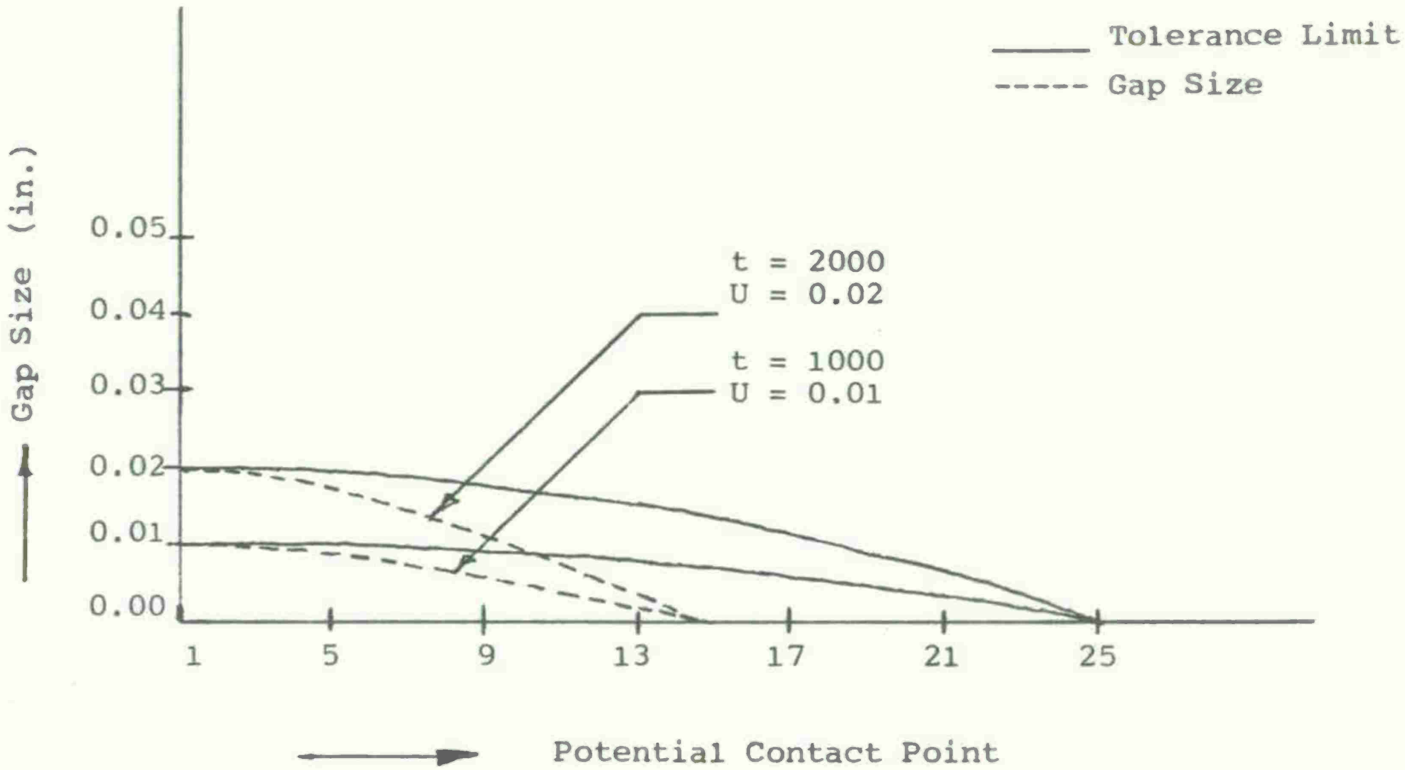


Figure 7. Gap Size vs. Potential Contact Point.

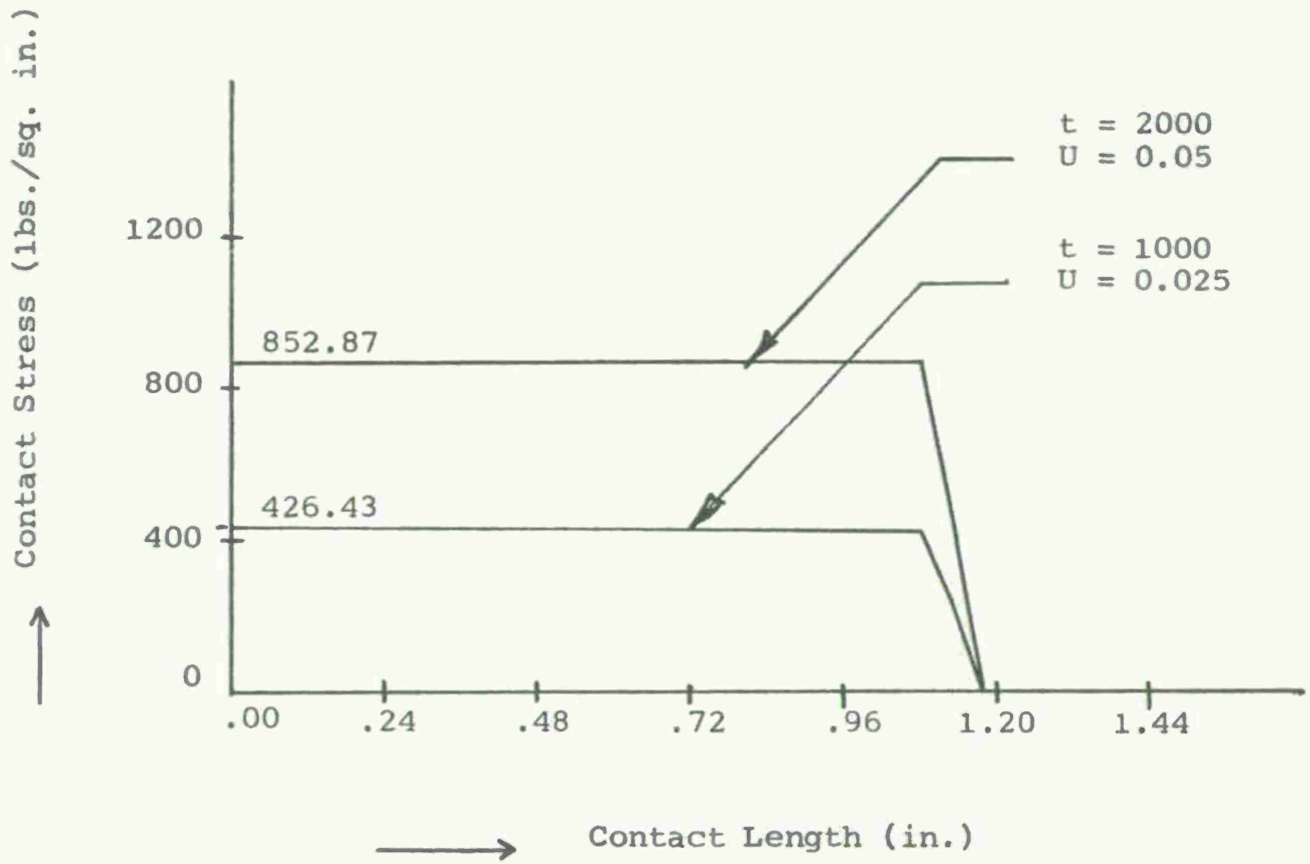


Figure 8. Contact Stress vs. Contact Length with Augmented Cost Function.

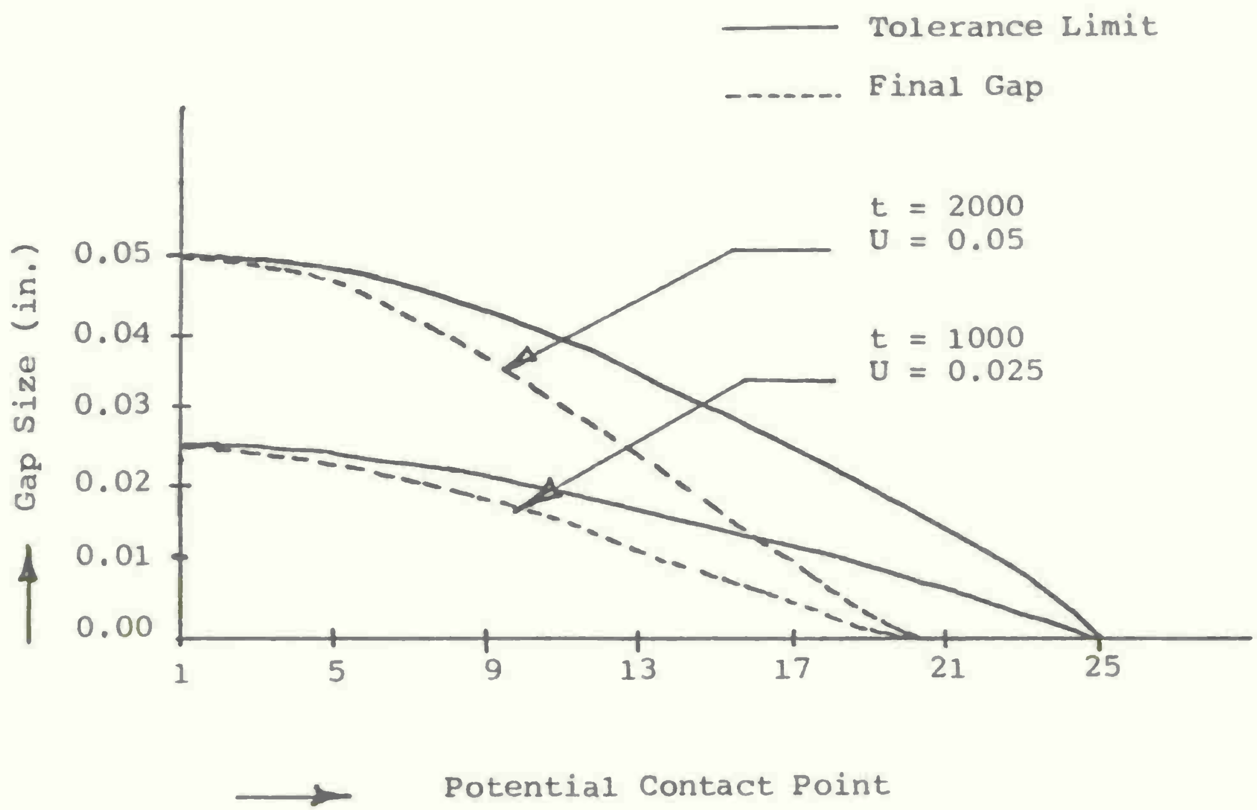


Figure 9. Gap Size vs. Potential Contact Point with Augmented Cost Function.

TABLE 1

COMPARISON OF RESULTS FOR LOADS
t = 1000 lbs. AND t = 2000 lbs.
FOR BEAM ON ELASTIC FOUNDATION

LOAD t	SR. NO.	U	PEAK STRESS lbs./sq.in.	COMPUTING TIME (SEC.)	NO. OF ITRNS	TIME/ ITRN (SEC.)	CONTACT LENGTH (IN.)
1000	1	.01	562.9	38.80	5	7.36	1.86
	2	.015	499.2	48.12	6	8.02	1.98
	3	.02	457.4	72.08	8	9.01	2.22
	4	.025	426.43	91.27	9	10.14	2.34
2000	1	.02	1125.6	36.87	5	7.374	1.86
	2	.03	998.25	48.94	6	8.157	1.98
	3	.04	914.8	73.00	8	9.125	2.22
	4	.05	852.87	93.17	9	10.35	2.34

iterations, with computing time per iteration varying only between 7 and 10 seconds. In all cases, peak stress is reduced considerably and the contact arc increases. For example, for t = 1000 lbs., the peak stress of the non-modified design was 1120 lbs./in.^{2} and contact arc length was 1.38 in. By adjusting the contours of two bodies, with a tolerance limit of U = 0.01 and t = 1000 lbs., Table 2 shows the peak stress reduces to 562.9 lbs./sq. in. and the contact arc length becomes 1.86 in.

3.2 Initially Bent Beam on an Elastic Foundation

The initial gap between a beam and elastic foundation is given by the formula $G\left[1 - \frac{x^2}{L^2}\right]$, where x is the distance from the center of the beam, and G is a constant greater than zero. Matrices B, A, b, and C are the same as for the problem in the preceding section, except that the vector a is obtained from the expression given above.

The quadratic programming problem was solved for 25 potential contact points on the half beam, with an interval size of 0.06 in. The potential contact length, as shown in Figure 10(a) is 2.88 in. With G = 0.0001 the resulting contact length was 1.38 in., for loads 1000 lbs. and 2000 lbs. These results are shown in Figure 10(b). The quadratic programming code ZORILLA [10] was again used to solve this problem.

TABLE 2

CONTACT STRESS DISTRIBUTION AND FINAL GAP
FOR BEAM ON ELASTIC FOUNDATION
t = 1000 lbs., U = 0.01

POINT NO.	CONTACT STRESS lb./sq.in.	TOLERANCE LIMIT (in.)	FINAL GAP SIZE FOR MODIFIED CONTOURS (in.)
1	562.886	0.1000000E-01	0.9999964E-02
2	562.888	0.9982634E-02	0.9910598E-02
3	562.889	0.9930551E-02	0.9614620E-02
4	562.891	0.9843744E-02	0.9165816E-02
5	562.891	0.9722218E-02	0.8595929E-02
6	462.892	0.9565968E-02	0.7923093E-02
7	562.892	0.9374995E-02	0.7147454E-02
8	562.894	0.9149302E-02	0.6315298E-02
9	562.893	0.8888885E-02	0.5400788E-02
10	562.893	0.8593746E-02	0.4458770E-02
11	562.895	0.82638863-02	0.3493937E-02
12	562.888	0.7899299E-02	0.2511400E-02
13	562.891	0.7499997E-02	0.1511019E-02
14	562.885	0.7065967E-02	0.5030558E-02
15	393.626	0.6597217E-02	0.0
16	59.206	0.6093744E-02	0.0
17	0.0	0.5555548E-02	0.0
18	0.0	0.4982639E-02	0.0
19	0.0	0.4374996E-02	0.0
20	0.0	0.3732643E-02	0.0
21	0.0	0.3055555E-02	0.0
22	0.0	0.2343752E-02	0.0
23	0.0	0.1597222E-02	0.0
24	0.0	0.8159771E-03	0.0
25	0.0	0.0	0.0

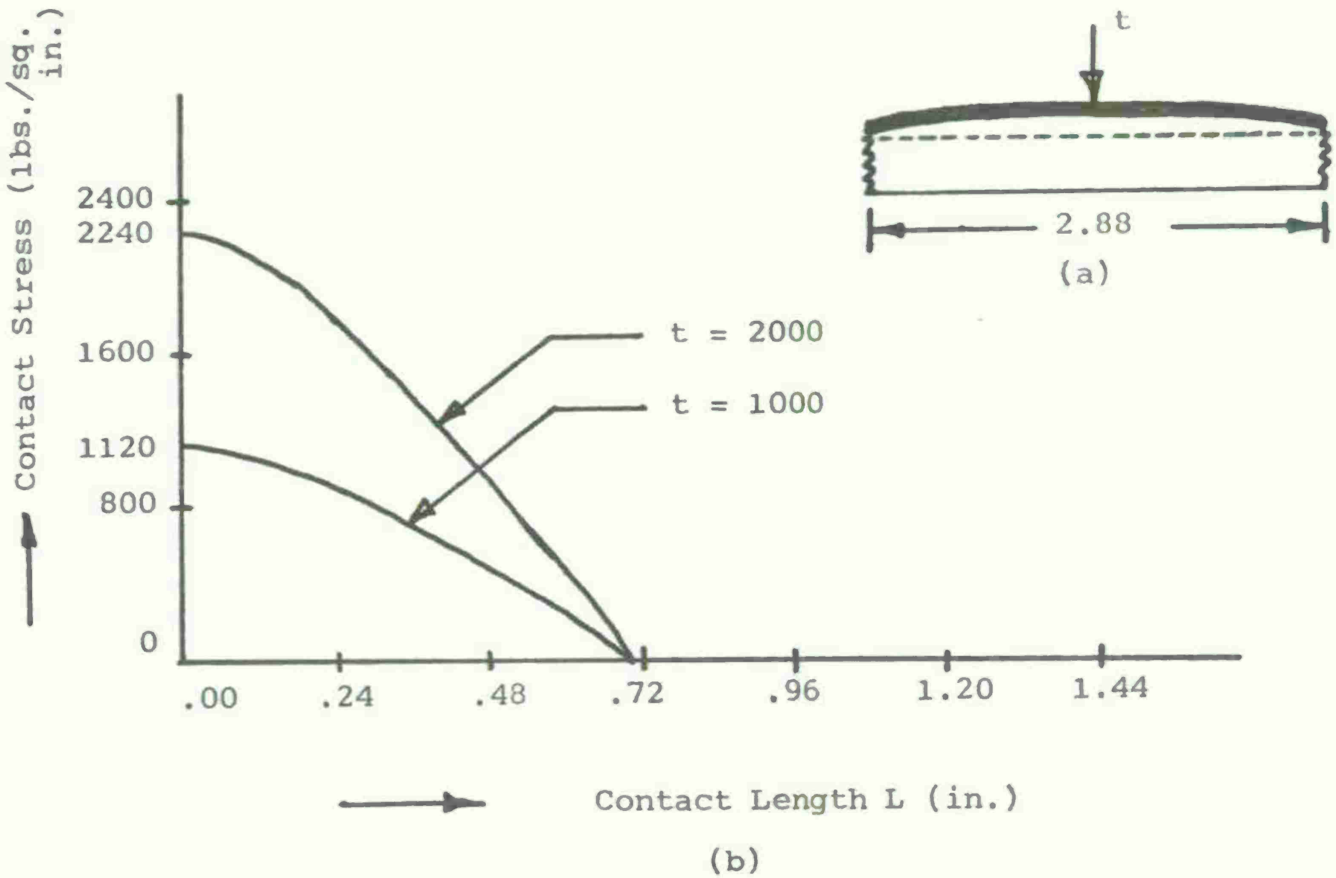


Figure 10. Contact Stress vs. Contact Length from Quadratic Programming Solution, with G = 0.0001.

One now has the contact arc to be employed for solving the linear programming problem:

Minimize $\bar{J} = -b_{n+1}$

subject to constraints

$$
\begin{aligned}
&\hat{S} - b_{n+1} \leq 0 \\
&-\hat{b} \leq -b^{o} \\
&\hat{b} \leq b^{1} \\
&-\hat{B}\hat{S} - \hat{A}q - \hat{b} = \hat{a} \\
&\hat{A}^{T}\hat{S} = C \\
&-\tilde{B}\hat{S} - \tilde{A}q \leq \tilde{a} + \tilde{b} \\
&\hat{S} \geq 0
\end{aligned}
\qquad (3.2.1)
$$

A linear programming tableau is formed from Equations 3.2.1, as explained in the previous section. The linear programming problem was solved and results are presented in Figures (11-16). A comparison of the results for two loads, t = 1000 lbs. and t = 2000 lbs., is given in Table 3. Values of U used are proportional to applied load for the cases t = 1000 lbs. and t = 2000 lbs. The peak stress obtained for t = 1000 lbs. is exactly half that for t = 2000 lbs. and the contact length is the same for t = 1000 lbs. and t = 2000 lbs. Thus, contact arc length and peak stress are dependent on both value of applied load and U. Table 4 gives the optimum stress

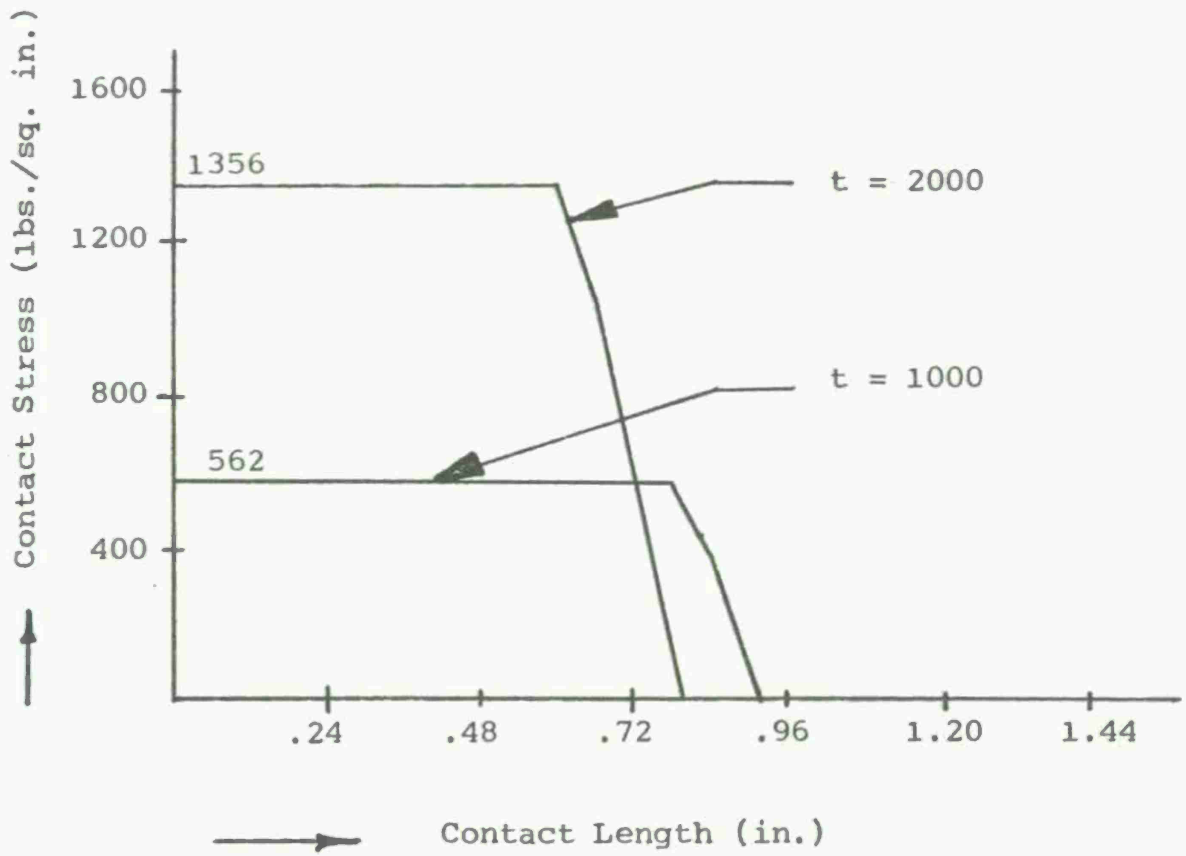


Figure 11. Contact Stress vs. Contact Length with U = .01 and G = 0.0001

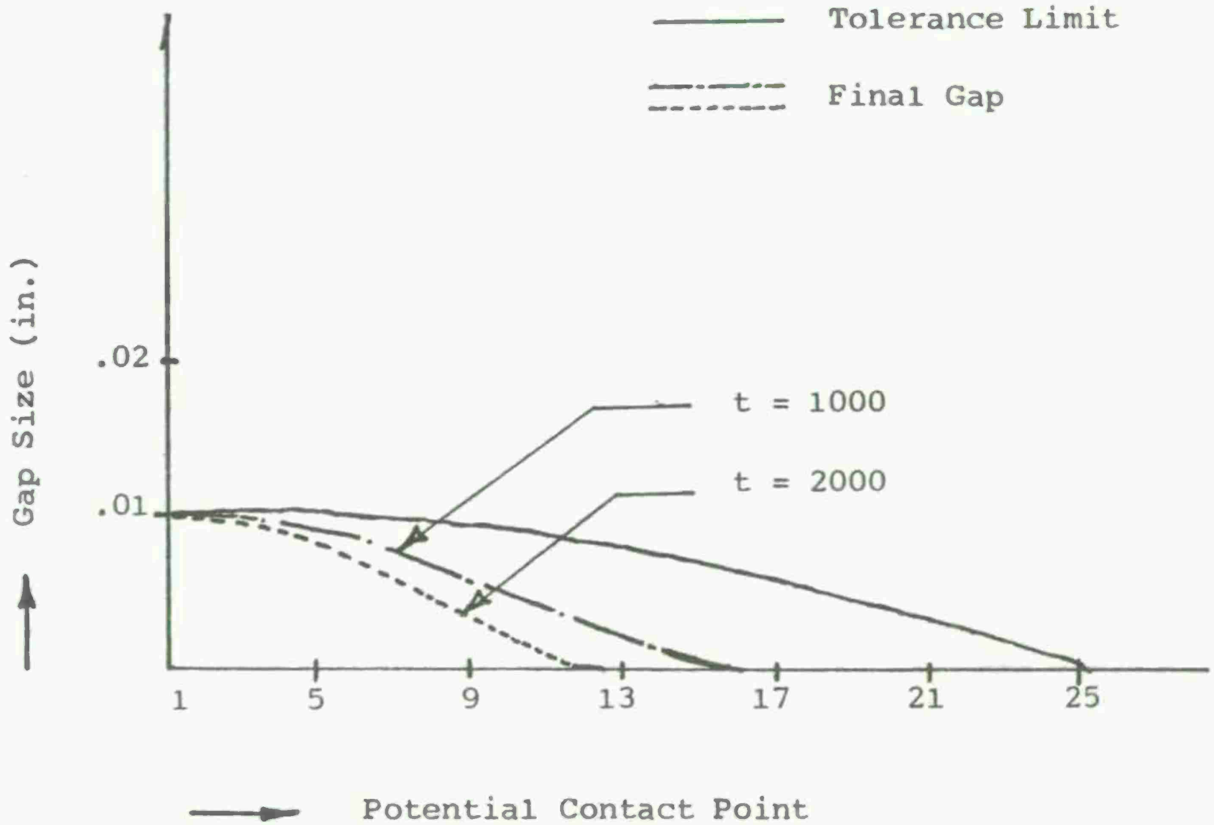


Figure 12. Gap Size vs. Potential Contact Point with U = 0.01 and G = 0.0001.

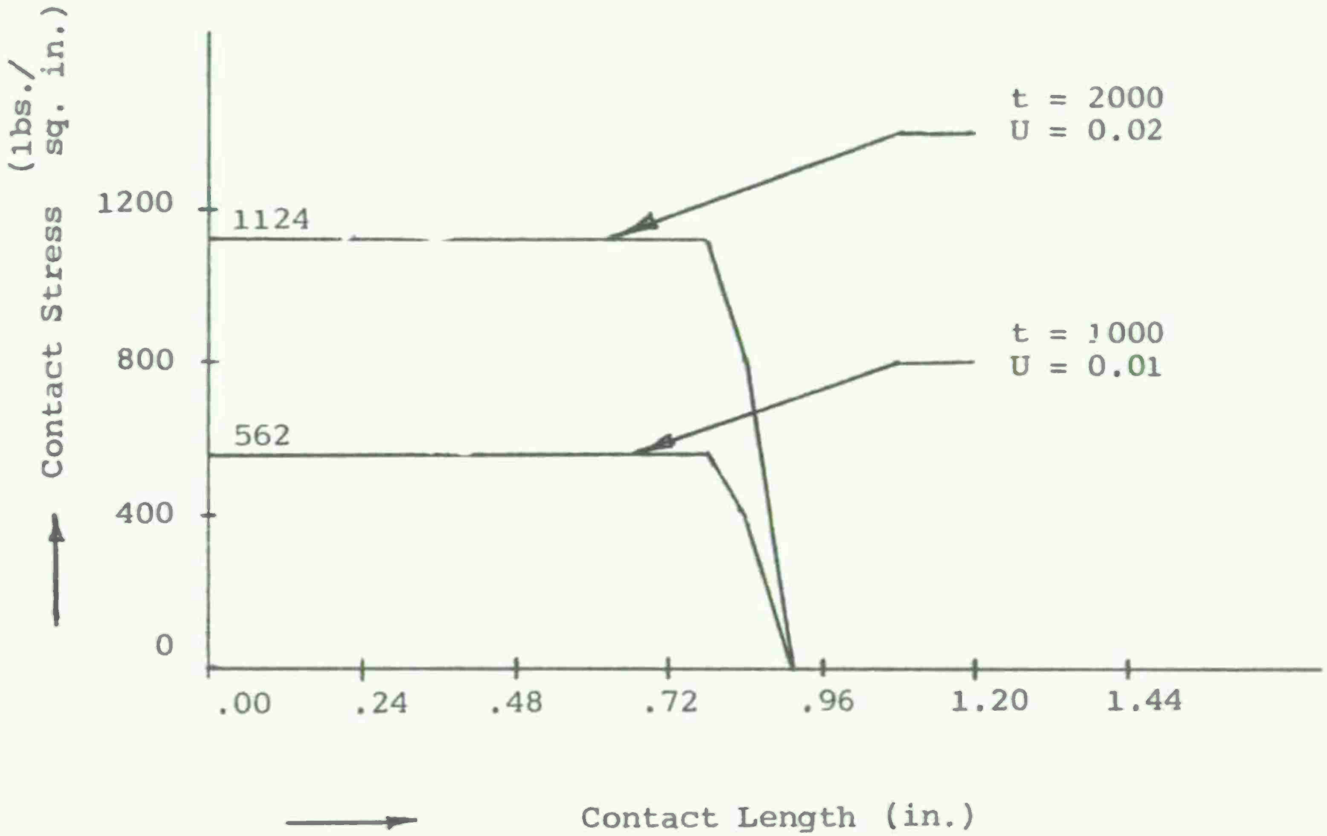


Figure 13. Contact Stress vs. Contact Length with G = 0.0001.

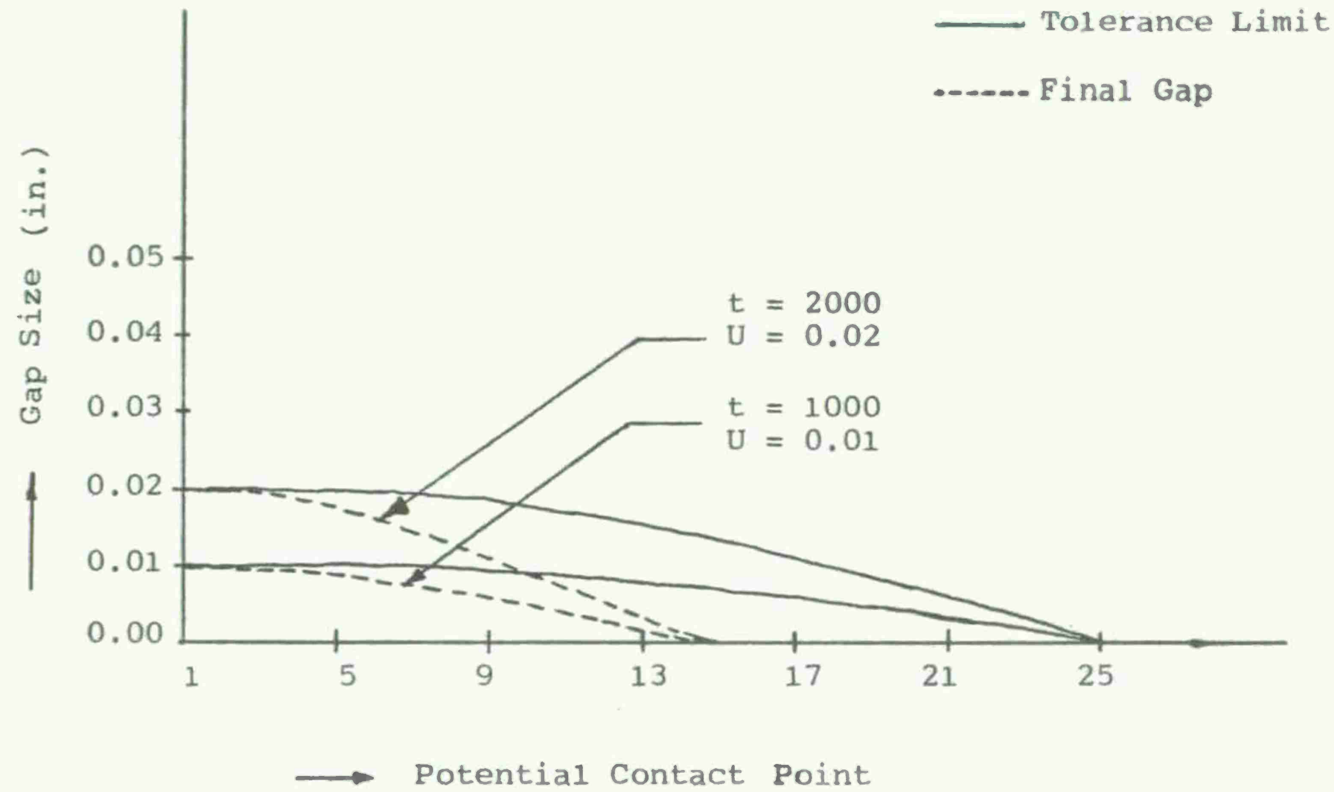


Figure 14. Gap Size vs. Potential Contact Point with G = 0.0001.

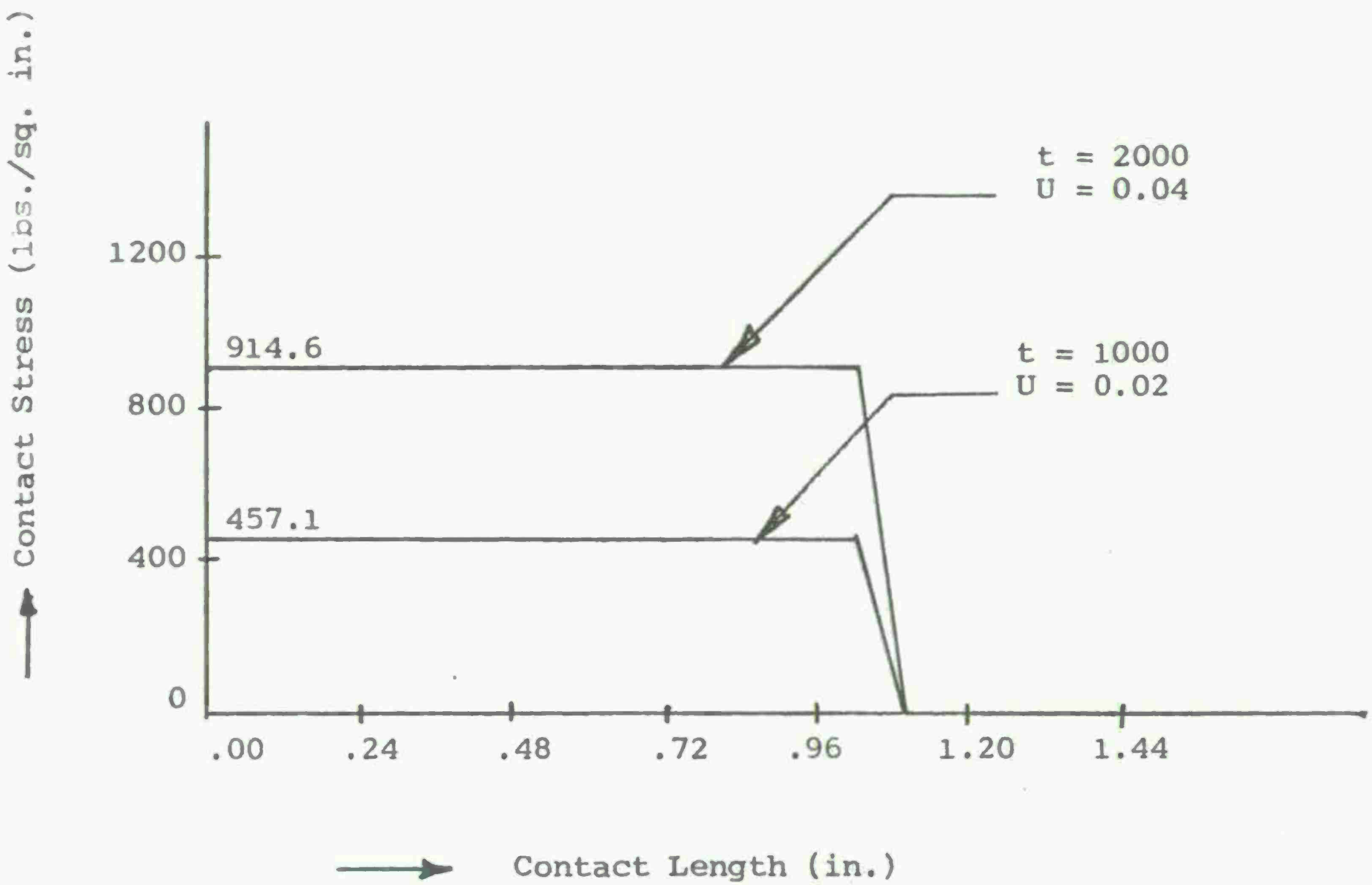


Figure 15. Contact Stress vs. Contact Length with Augmented Cost Function and with G = 0.0001.

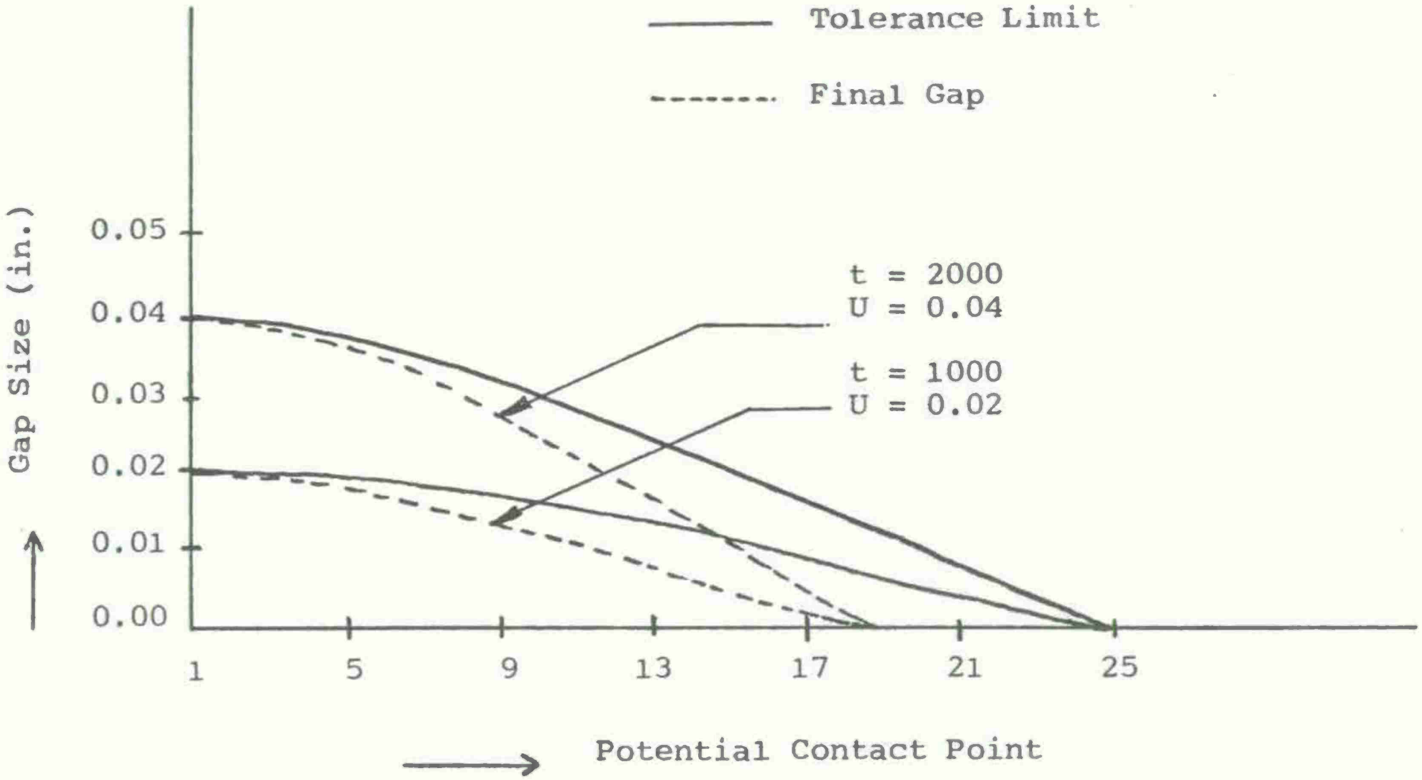


Figure 16. Gap Size vs. Potential Contact Point with Augmented Cost Function and with G = 0.0001.

TABLE 3

COMPARISON OF RESULTS FOR LOADS t = 1000 lbs. AND t = 2000 lbs. FOR AN INITIALLY BENT BEAM ON ELASTIC FOUNDATION

LOAD t	SR. NO.	U	PEAK STRESS lbs./sq.in.	COMPUTING TIME (SEC.)	NO.OF ITRNS	TIME/ ITRN (SEC.)	CONTACT LENGTH (in.)
1000	1	0.005	677.6	21.90	3	7.3	1.62
	2	0.01	561.7	39.05	5	7.81	1.86
	3	0.015	498.7	49.32	6	8.22	1.98
	4	0.02	457.1	71.82	8	8.98	2.22
2000	1	0.01	1356.0	21.65	3	7.217	1.62
	2	0.02	1124.05	41.37	5	8.274	1.86
	3	0.03	997.2	49.17	6	8.195	1.98
	4	0.04	914.6	73.74	8	9.22	2.22

TABLE 4

CONTACT STRESS DISTRIBUTION AND FINAL GAP FOR INITIALLY BENT BEAM ON ELASTIC FOUNDATION
t = 1000 lbs., U = 0.01

POINT NO.	CONTACT STRESS lb./sq. in.	TOLERANCE LIMIT (in.)	INITIAL GAP (in.)	FINAL GAP (in.)
1	561.712	0.1000000E-01	.1000000E-03	0.9999931E-02
2	561.714	0.9982634E-02	0.9982637E-04	0.9929962E-02
3	561.717	0.9930551E-02	0.9930554E-04	0.9638943E-02
4	561.719	0.9843744E-02	0.9843748E-04	0.9196758E-02
5	561.721	0.9722218E-02	0.9722220E-04	0.8627117E-02
6	561.721	0.9565968E-02	0.9565971E-04	0.7991239E-02
7	561.720	0.9374995E-02	0.9374999E-04	0.7174265E-02
8	561-720	0.9149302E-02	0.9149304E-04	0.6328776E-02
9	561.722	0.8888885E-02	0.8888886E-04	0.5433228E-02
10	561.722	0.8593746E-02	0.8593748E-04	0.4489828E-02
11	561.722	0.8263886E-02	0.8263886E-04	0.3522005E-02
12	561.722	0.7899299E-02	0.7899302E-04	0.2532016E-02
13	561.717	0.7499997E-02	0.7499997E-04	0.1539212E-02
14	561.719	0.7065967E-02	0.7065968E-04	0.5400113E-02
15	406.454	0.6597217E-02	0.6597218E-04	0.0
16	62.750	0.6093744E-02	0.6093745E-04	0.0
17	0.0	0.5555548E-02	0.5555548E-04	0.0
18	0.0	0.4982639E-02	0.4982639E-04	0.0
19	0.0	0.4374996E-02	0.4982639E-04	0.0
20	0.0	0.3732644E-02	0.3732644E-04	0.0
21	0.0	0.3055555E-02	0.3055555E-04	0.0
22	0.0	0.2343752E-02	0.2343752E-04	0.0
23	0.0	0.1597222E-02	0.1597222E-04	0.0
24	0.0	0.8159771E-03	0.8159771E-05	0.0
25	0.0	0.0	0.0	0.0

distribution, tolerance limit, initial gap, and final gap at the potential contact points. Peak contact stress with U = 0.01 and t = 1000 lbs. was found to be 561.712 lbs./sq. in. This peak stress is not much different from the one obtained in the case of beam on elastic foundation with no initial gap.

The quadratic programming problem was again solved for the same data, but with an initially bent beam on an elastic foundation with G = 0.001 and G = 0.005. The results are presented in Figures (17-18). It is observed that, as the value of G is increased, peak contact stress decreases. In the case of G = 0.005 contact length for 1000 lbs. and 2000 lbs. loads differs. It is also observed that peak contact stress depends on the applied load.

Once the contact arc is known from the contact analysis problem, the linear programming problem was solved for G = 0.001 and G = 0.005, with two different loads in each case. Results are presented in Figures (19-22). Table 5 shows the comparison of results with U = 0.01, for a beam on an elastic foundation with no initial gap and with an initial gap. Table 6 shows the comparison of results for t = 2000 lbs. with values of U ranging from 0.01 to 0.05. It is noted that the peak contact stress decreases if the value of U and G are increased.

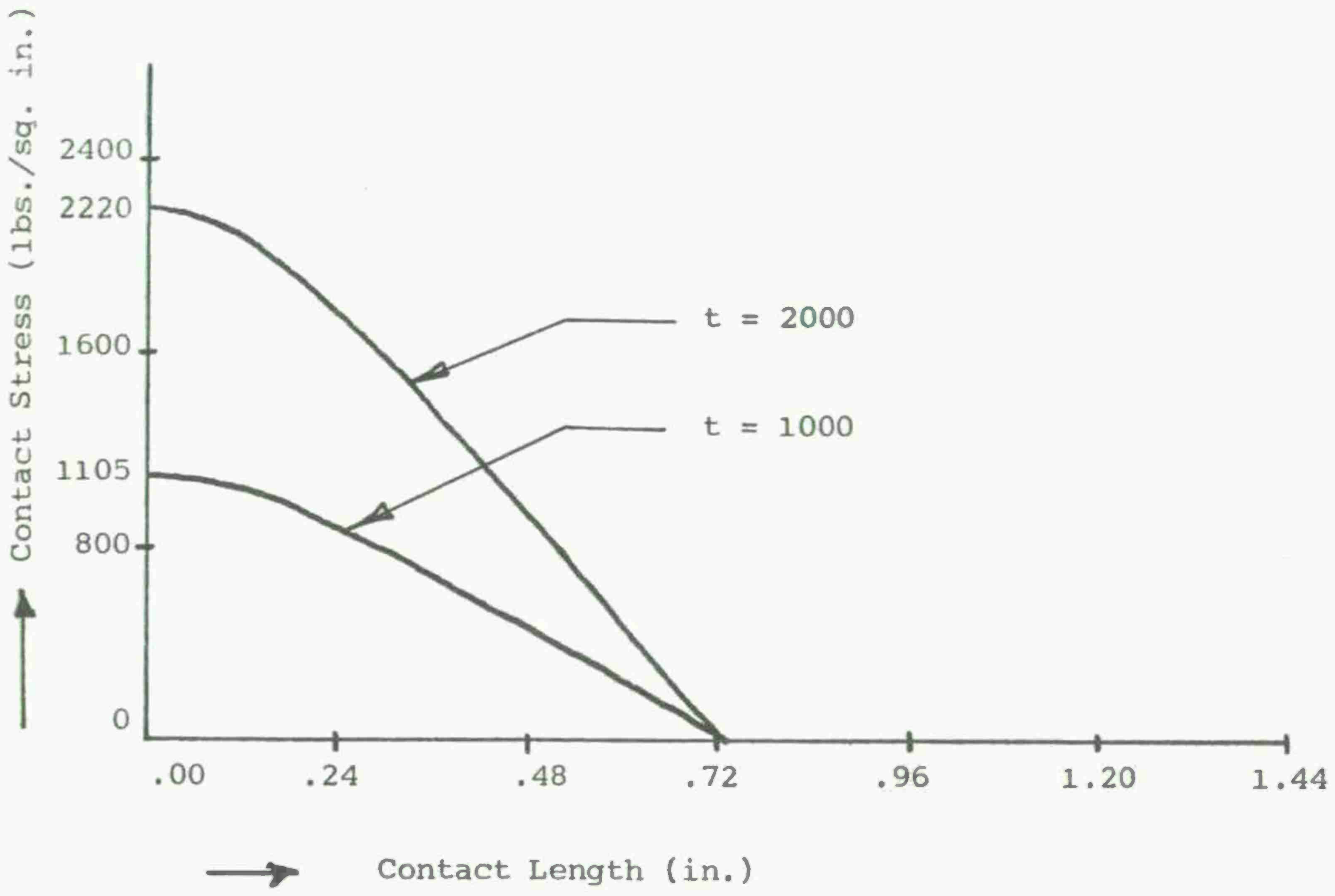


Figure 17. Contact Stress vs. Contact-Length from Quadratic Programming Solution with G = 0.001.

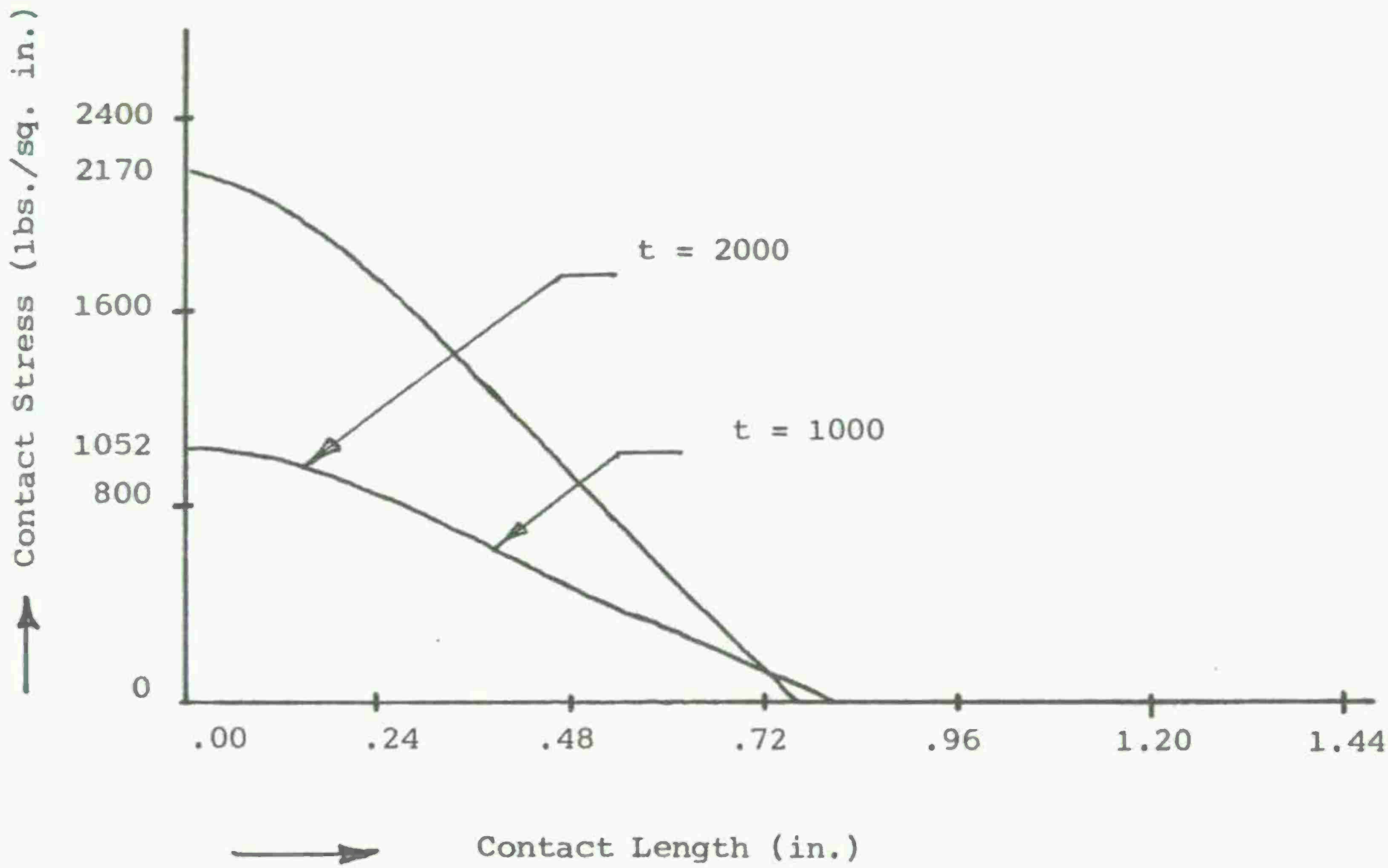


Figure 18. Contact Stress vs. Contact Length from Quadratic Programming Solution with G = 0.005.

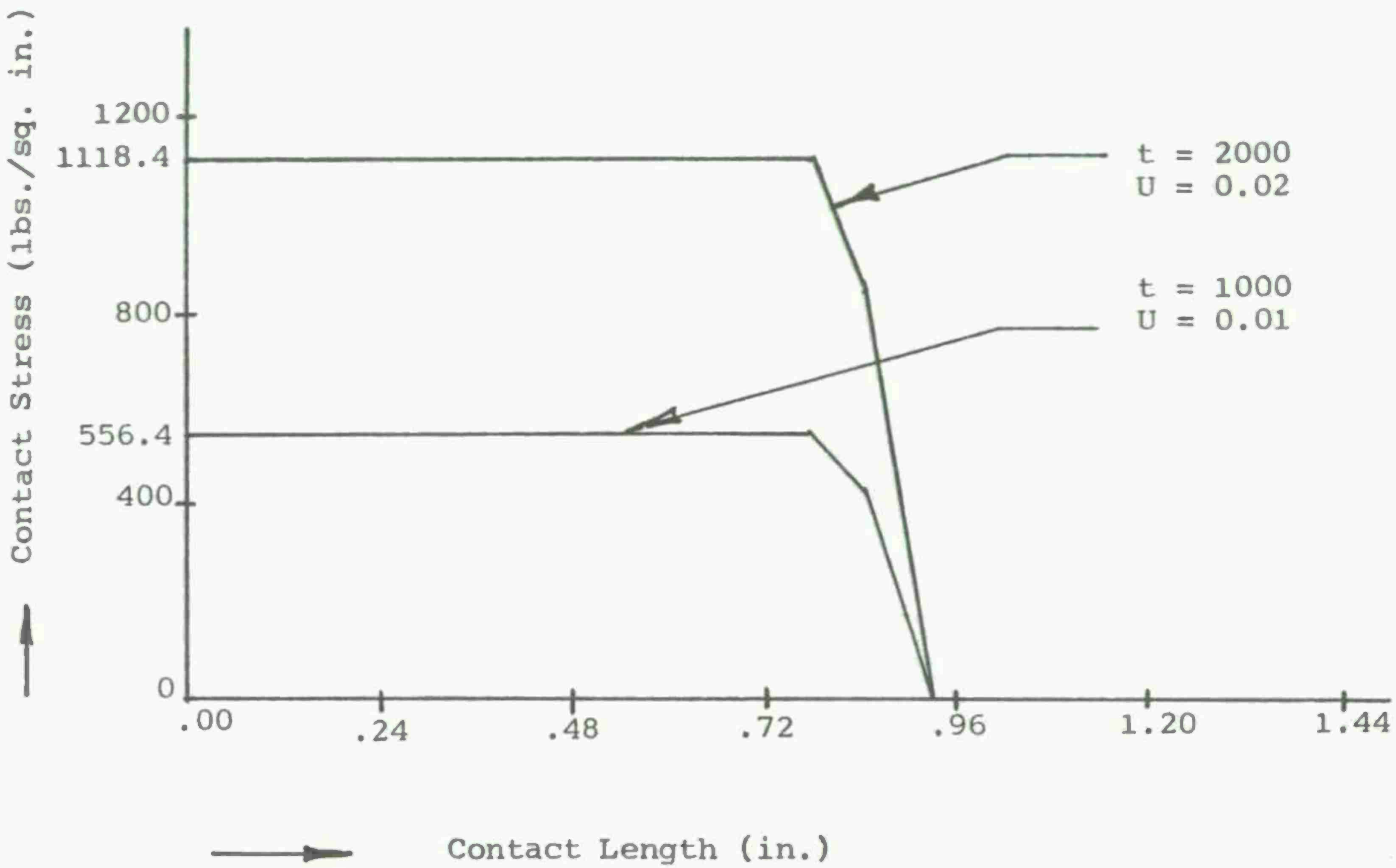


Figure 19. Contact Stress vs. Contact Length with G = 0.001.

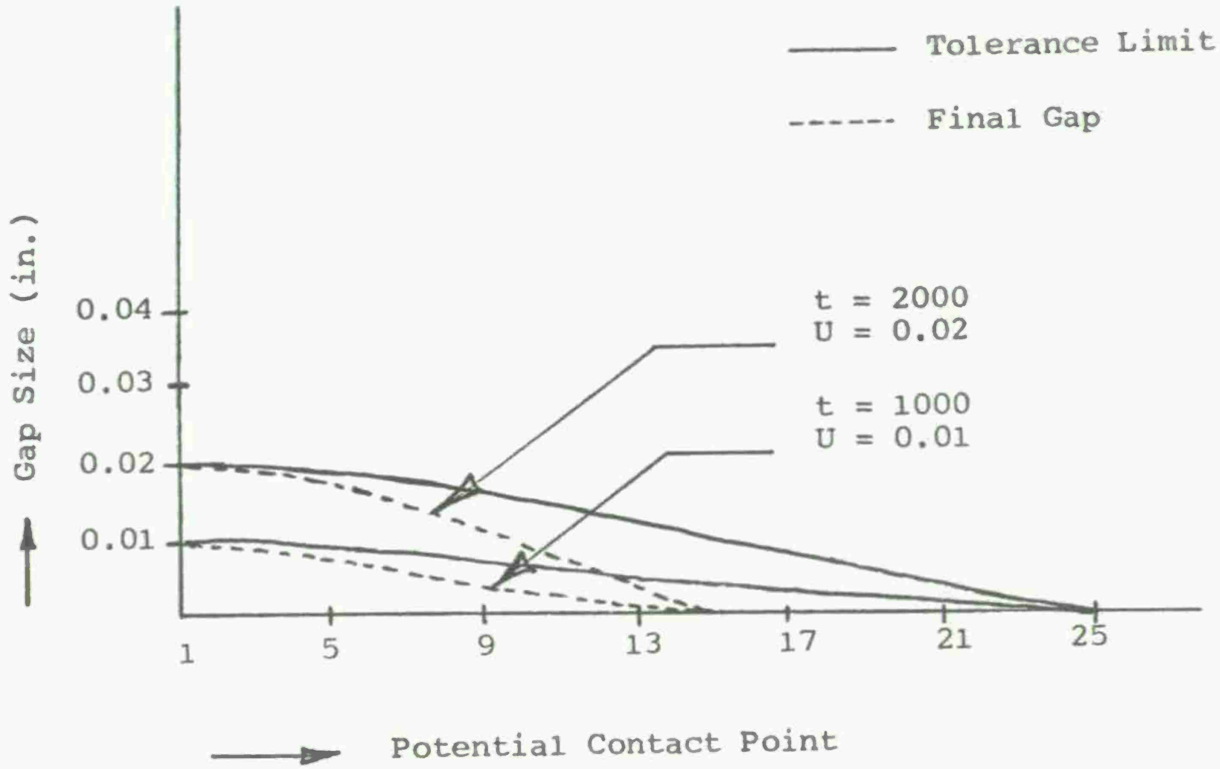


Figure 20. Gap Size vs. Potential Contact Point with G = 0.001.

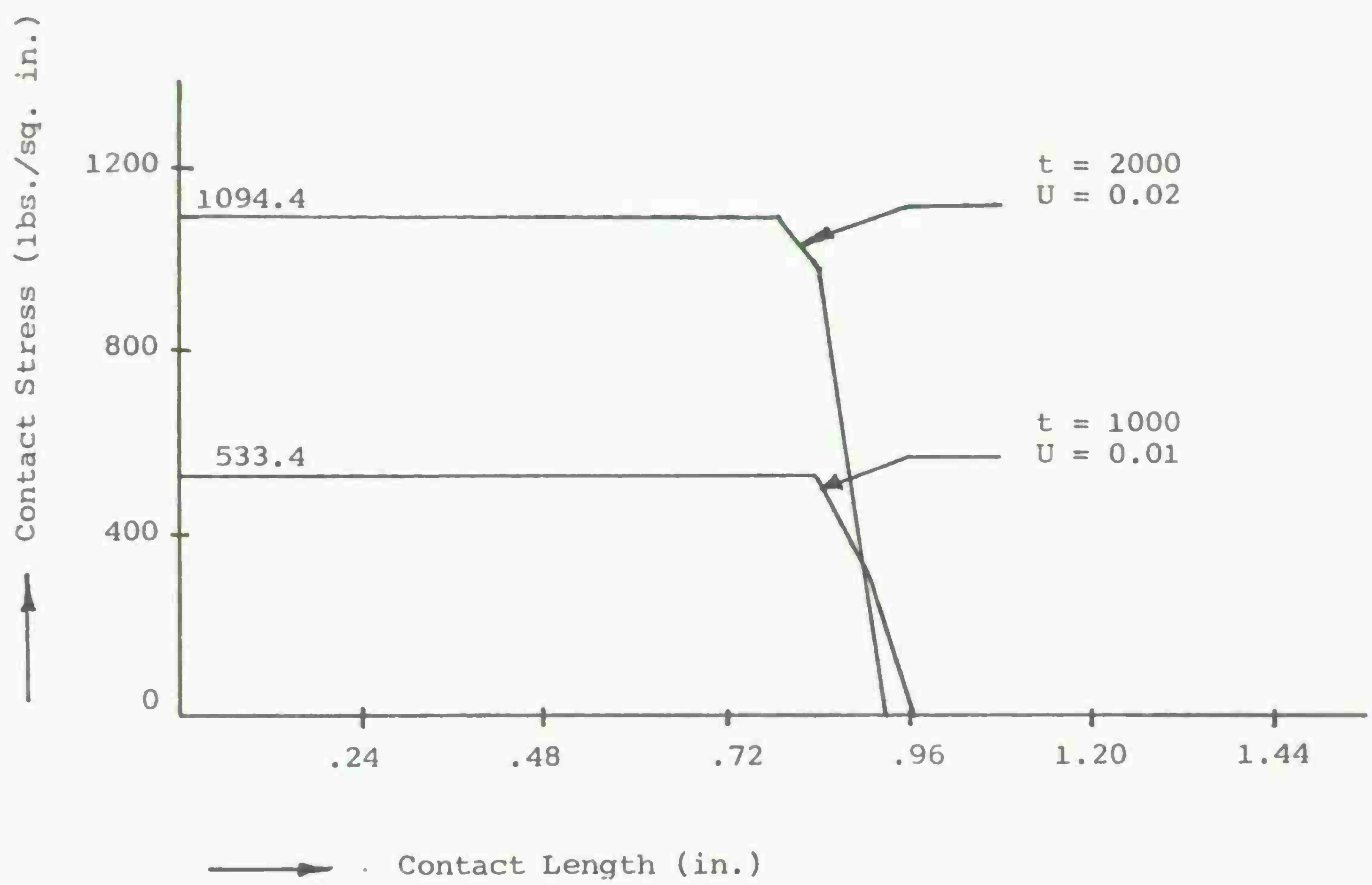


Figure 21. Contact Stress vs. Contact Length with G = 0.005.

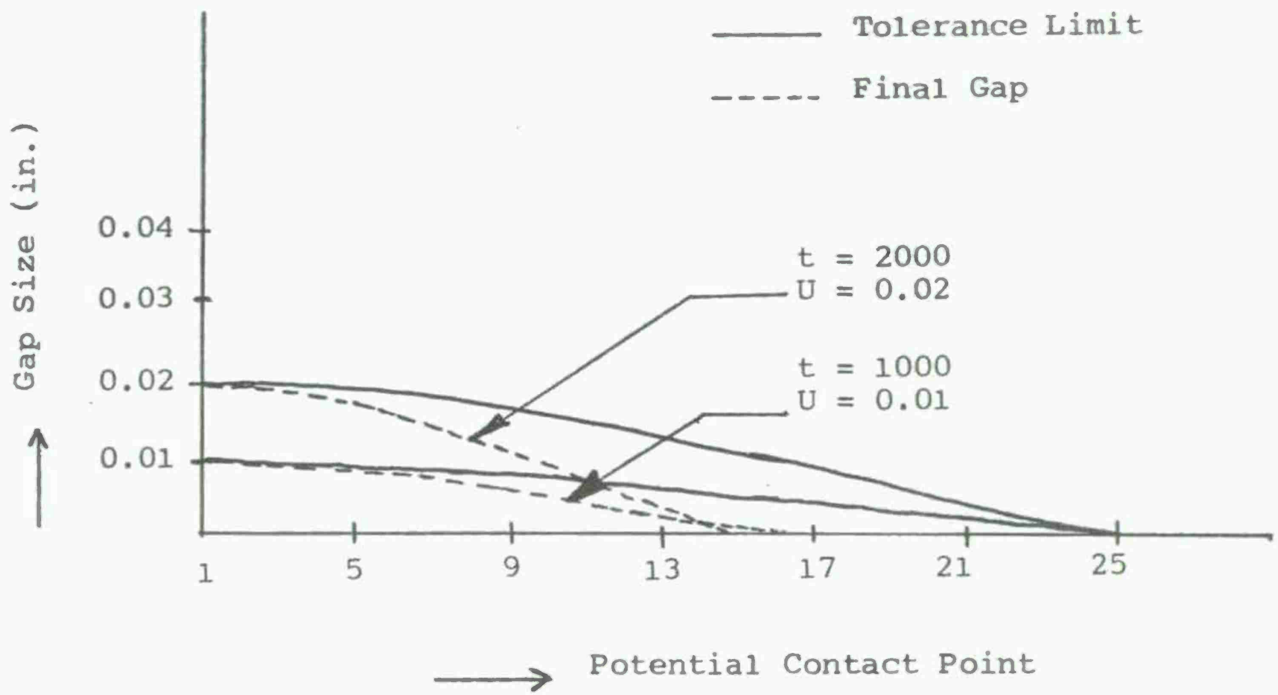


Figure 22. Gap Size vs. Potential Contact Point with G = 0.005.

TABLE 5

COMPARISON OF RESULTS FOR BEAM ON ELASTIC FOUNDATION
WITH NO INITIAL GAP AND WITH DIFFERENT
INITIAL GAPS FOR U = 0.01

LOAD (lbs.)	G	CONTACT STRESS (lbs./ sq. in.)	CONTACT LENGTH (in.)	NUMBER OF ITRATIONS	TIME/ITRN (sec.)
1000	0	562.9	1.86	5	7.36
	.0001	561.7	1.86	5	7.81
	.001	556.4	1.86	4	8.6
	.005	533.4	1.98	4	9.375
2000	0	1358.5	1.62	3	7.25
	.0001	1356.0	1.62	3	7.22
	.001	1349.5	1.62	2	7.92
	.005	1317.8	1.62	2	8.5

TABLE 6

COMPARISON OF RESULTS FOR BEAM ON ELASTIC FOUNDATION WITH NO INITIAL GAP AND WITH DIFFERENT INITIAL GAPS FOR t = 2000 lbs.

G	ITEM	QP SOLUTION	LINEAR PROGRAMMING SOLUTION				
			U				
			0.01	0.02	0.03	0.04	0.05
0	CONTACT STRESS lbs./in.^{2}	2242	1358.5	1125.6	993.25	914.8	852.87
.0001		2240	1356	1124	997.2	914.6	852.6
.001		2220	1349.5	1118.4	994.5	911.1	849.2
.005		2170	1317.8	1094	973.3	895.2	833.9
0	CONTACT LENGTH in.	1.38	1.62	1.86	1.98	2.22	2.34
.0001		1.38	1.62	1.86	1.98	2.22	2.34
.001		1.50	1.62	1.86	1.98	2.22	2.34
.005		1.50	1.62	1.86	2.10	2.22	2.34

Chapter IV

CONCLUSIONS

The technique presented here to solve surface contour design problem is quite simple, and has given consistently good results. By adjusting the contours of the contacting bodies, the stresses developed are reduced considerably. The problem of a beam on an elastic foundation demonstrates some of these facts. The peak stress of the unmodified structure is highest at the center of the beam and decreases at the end points of the contact region. The contact design problem gives a relatively constant stress distribution on the contact region, with much reduced peak contact stresses.

Tolerance limits affect the solution as is shown in Table 6. In this problem, the load and design tolerance limits have a measurable effect on the solution of the design problem. However, as shown in Table 1, the solution changes proportionally if the load and tolerance limit are changed in the same proportion.

Suggestions

This design method can be extended to other problems as follows:

1. Asymmetrical problems for beam on an elastic foundation.
2. Circular inclusion problems for elastic bodies in contact.
3. Multibody contact problems.

BIBLIOGRAPHY

1. Singh, D. P. Investigation of An Unbonded Contact Problem. Ph.D. Dissertation, Dept. of Mechanics and Hydraulics, University of Iowa, Iowa City, Iowa, April, 1970.

2. Chand, R., Solution of General Unbonded Contact Problems by Quadratic Programming and Finite Element Techniques, Ph.D. Dissertation, University of Iowa, Iowa City, Iowa, April, 1974.

3. Conry, T. F., The Use of Mathetmatic Programming in Design for Uniform Load in Nonlinear Elastic Systems, Ph.D. Thesis, Mechanical Engineering Dept., Univ. of Wisconsin, Wisc., January, 1970.

4. Yoon, K.E. Analysis of Contact Problems by Means of an Optimization Technique, Ph.D. Dissertation, Department of Mechanics and Hydraulics, University of Iowa, Iowa City, Iowa, August, 1970.

5. Hadley, G. Nonlinear and Dynamic Programming. Addison-Wesley Publishing Co., Inc., London, 1964.

6. Chand, R., Haug, E. J., Rim K., Analysis of Unbounded Contact Problems by Means of Quadratic Programming. J. Opt. Theory and Appl. To Appear, 1974.

7. Haug, E. J. Engineering Design Handbook. Computer Aided Design of Mechanical Systems. HQ U.S. Army Material Command, 5001 Eisenhower Avenue, Alexandria, Va-22304, July 1973.

8. Hadley, G. Linear Programming. Addison-Wesley Publishing Co., Inc. Reading, Massachusetts, U.S.A. London, England, 1962.

9. Goldburg, Richard R. Methods of Real Analysis, Xerox College Publishing, Waltham, Massachusetts, Toronto, November, 1963.

10. Soults, D. J., Zrubek, Janet J. and Sposito, V.A., 'ZORILLA' Reference Manual, Numerical Analysis-Programming Series, No. 9. Statistical Laboratory. Iowa State University, Ames, Iowa, May, 1969.

11. Kuester, James L., Mize, Joe H. _Optimization Techniques with Fortran_. McGraw Hill Book Co., New York, 1973.

APPENDICES

APPENDIX A:

FORMULATION OF MATRICES B, A, a, and C

The compatibility condition for deformation is given by

$$\varepsilon = u^1 + u^2 + d \geqslant 0 \qquad \text{(A-1)}$$

where ε is the gap vector after deformation of two bodies, u^1 and u^2 denote the normal displacement vectors of potential contact points on Bodies 1 and 2, and d is the vector of initial gap between the two bodies.

Body 2 of Figure 23 is fixed, so only rigid body displacement for Body 1 is considered. The displacement of points on Body 1 is due to rigid body displacements and elastic deformation. Elastic deformation of points on Body 1 is determined relative to zero values of the rigid body coordinates, and total displacement is found by superposition. Generalized displacement coordinates are denoted by q.

The total displacement of potential contact points on Body 1 can be written in vector form as

$$u^1 = P^T u^1_e + Aq \qquad \text{(A-2)}$$

where u^1_e is a reduced vector of elastic displacements of potential contact points on Body 1. The vector u^1_e

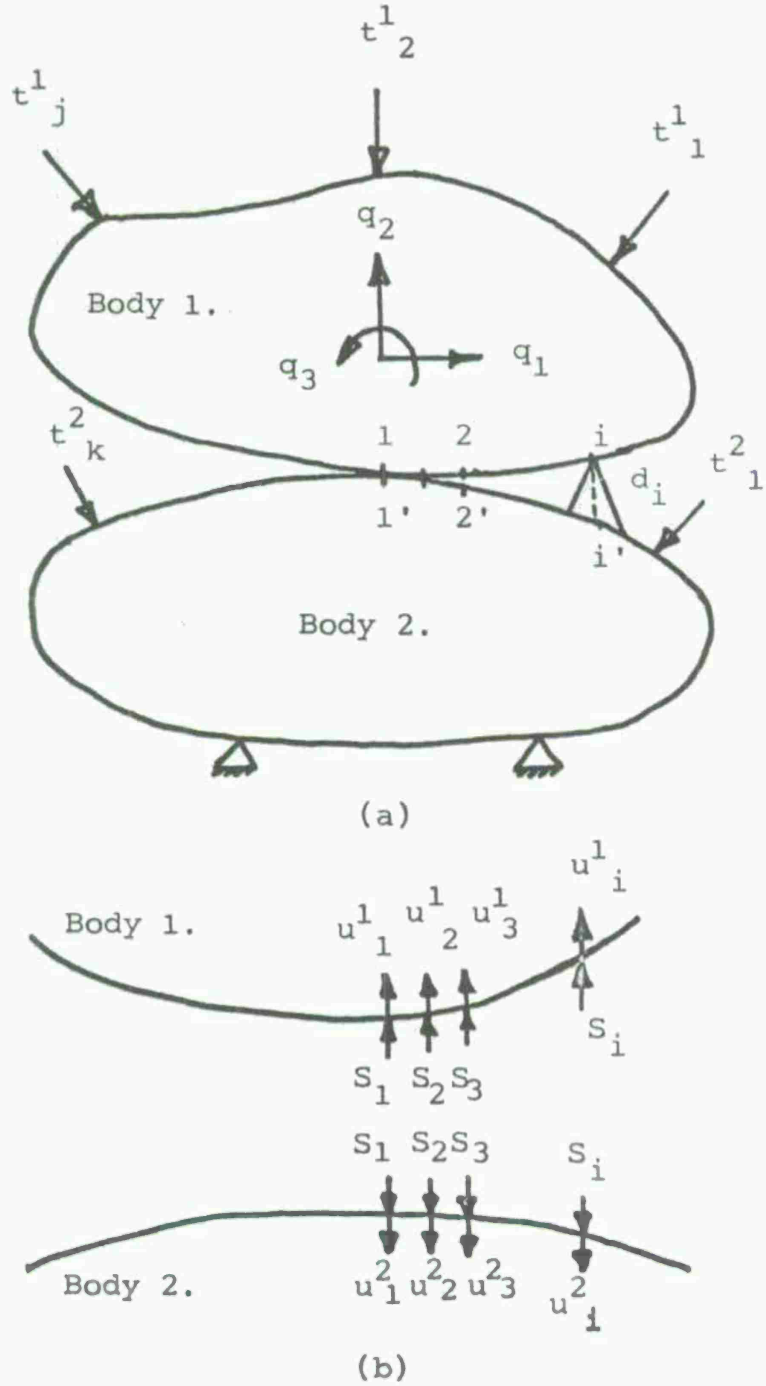


Figure 23. Two Bodies in Contact.

is determined with q = 0, so it cannot contain displacement components that are used as rigid body degrees of freedom. The projection matrix P is given by $P = [0 \mid I_{n-m}]$, where n is the number of potential contact points and m is the number of degrees of freedom. The matrix A is an nxm affine transformation that contributes to rigid body displacements.

The deformation vector u^1_e is further decomposed as

$$u^1_e = F^1PS + V^1_e \qquad \text{(A-3)}$$

where F^1 is the flexibility matrix for Body 1, with q = 0, and V^1_e is the vector of elastic displacements of potential contact points due to externally applied forces t^1 on Body 1, with q = 0. From equations (A-2) and (A-3),

$$u^1 = P^T[F^1PS + V^1_e] + Aq \qquad \text{(A-4)}$$

For Body 2,

$$u^2 = F^2S + V^2_e, \qquad \text{(A-5)}$$

where F^2 is an nxn flexibility matrix for potential contact points on Body 2, and V^2_e is displacement of potential contact points on Body 2, due to externally applied forces.

Equation (A-1) can now be written as

$$\mathcal{E} = (P^T F^1 P + F^2)S + Aq + P^T V^1{}_e + V^2{}_e + d \geqslant 0, \quad \text{(A-6)}$$

so

$$B = P^T F^1 P + F^2$$

$$A \equiv A$$

and

$$a = P^T V^1{}_e + V^2{}_e + d$$

<u>Equilibrium</u> <u>Equations</u>

The work done by stresses and the applied forces on Body 1, due to rigid body displacements only, is

$$W = S^T Aq + t^{1^T} Hq \qquad \text{(A-7)}$$

where H is a matrix that gives rigid body displacement at the points of application of t^1.

Varying the rigid body displacements, the principle of virtual work gives

$$\delta W = S^T A\delta q + t^{1^T} H\delta q = 0 \qquad \text{(A-8)}$$

Thus,

$$(S^TA + t^{1^T}H)\delta q = 0 \qquad \text{(A-9)}$$

Since all components of q are independent, it is necessary that

$$A^TS + H^Tt^1 = 0 \qquad \text{(A-10)}$$

or

$$A^TS = -H^Tt^1$$

Defining

$$C = -H^Tt^1 \qquad \text{(A-11)}$$

Equation (A-10) can be written as

$$A^TS = C \qquad \text{(A-12)}$$

APPENDIX B:

PROGRAM LISTING

```
C       ******************************************************
C       *                                                    *
C       *                    MAIN PROGRAM                    *
C       * BEAM ON AN ELASTIC FOUNDATION WITH NO INITIAL GAP. *
C       *                                                    *
C       ******************************************************
C
C
C               DESCRIPTION OF PARAMETERS.
C
C       DEL     IS THE INTERVAL SIZE BETWEEN THE POTENTIAL CONTACT
C               POINTS(P.C.P.)
C       AL      IS HALF THE P.C.LENGTH.
C       MNM     IS THE NO. OF DEGREES OF FREEDOM.
C       NN      IS THE NO. OF P.C.PS.
C       NC=NCI  IS THE NO. OF POINTS IN CONTACT.
C       NNC     IS THE NO. OF POINTS NOT IN CONTACT.
C       P       IS THE EXTERNALLY APPLIED LOAD.
C       ITRN    IS THE LINEAR PROGRAMMING ITERATION NUMBER.
C       M1      IS THE NO. OF CONSTRAINTS PLUS ONE.
C       N1      NO. OF DESIGN VARIABLES PLUS NO.OF GE. TYPE OF CONSTRAINTS.
C
C       M1=1+4*NC+MNM+NNC
C       N1=2*MNM+1+2*NC+NGE
C
C                    DESCRIPTION OF MATRICES.
C
C       A1(NN,MNM)   IS THE AFFINE TRANSFORMATION MATRIX FOR RIGID BODY
C                    DISPLACEMENT CONTRIBUTION FOR BODY 1.
C       T(NN,NN)     IS THE COMBINED INFLUENCE COEFFICIENT MATRIX.
C       F(NN,NN)     IS EQUIVALENT TO T(NN,NN).
C
C       FOLLOWING MATRICES ARE GENRATED FROM THE ABOVE TWO MATRICES.
C
```

```
C       BHAT(NN,NN)  IS THE REDUCED MATRIX 'T' WITH ROWS AND COLUMNS
C                    CORROSPONDING TO POINTS NOT IN CONTACT REMOVED.
C       AHAT(NN,MNM) IS THE MATRIX 'A1' WITH CORROSPONDING ROWS FOR
C                    POINTS NOT IN CONTACT REMOVED.
C       BTEL(NN,NN)  IS THE MATRIX 'T' WITH ROWS CORROSPONDING TO POINTS
C                    IN CONTACT AND COLUMNS CORROSPONDING TO POINTS NOT
C                    IN CONTACT REMOVED.
C       ATEL(NN,MNM) IS THE MATRIX 'A1' WITH ROWS CORROSPONDING TO
C                    POINTS IN CONTACT REMOVED.
C       B(M1,N1)     IS THE MATRIX OF CONSTRAINT COEFFICIENTS.
C
C                  DESCRIPTION OF VECTORS.
C
C       SA(NN)     IS THE VECTOR OF TOLRANCE LIMIT FOR CONTOUR
C                  DESIGN VARIABLES.
C       SSA(NN)    IS THE VECTOR OF INITIAL GAP BETWEEN THE TWO BODIES
C                  AT THEIR P.C.PS.
C       SB(NN)     IS THE VECTOR OF CONTOUR MODIFICATIONS.
C       SAHAT(NN)  IS THE VECTOR 'SA' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT REMOVED.
C       ASHAT(NN)  IS THE VECTOR'SSA' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT REMOVED.
C       ASHAT(NN)  IS THE VECTOR 'SB' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT REMOVED.
C       SATEL(NN)  IS THE VECTOR 'SA' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT.
C       SBTEL(NN)  IS THE VECTOR 'SA' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT.
C       ASTEL(NN)  IS THE VECTOR'SSA' WITH COMPONENTS CORROSPONDING TO
C                  POINTS NOT IN CONTACT.
C       RQ(M1)     IS THE VECTOR OF R.H.S. VALUES OF CONSTRAINTS.
C       NBP(N1)    COST FUNCTION ROW VECTOR.
C       K11(NN)    IS THE VECTOR OF INDICED FOR POINTS NOT IN CONTACT.
C       K(NN)      IS THE VECTOR OF INDICES OF POINTS IN CONTACT.
```

```
C     KI1(NN)     IS EQUIVALENT TO K(NN).
C     II(NN)      IS THE VECTOR HAVING COMPONENTS AS 1 OR 0.
C                 1 FOR POINTS IN CONTACT AND 0 FOR PTS. NOT IN CONTACT.
C     SHAT(NN)    IS THE VECTOR OF CONTACT FORCES.
C     RBD(MNM)    IS THE RIGID BODY DISPLACEMENT VECTOR.
C     RBD1(2*MNM)
C     Z(MNM)      IS THE VECTOR HAVING COMPONENTS AS TOTAL EXTERNAL
C                 APPLIED LOAD.
C     X(I)        IS THE DISTANCE OF I TH P.C.P. FROM CENTER.
C     CST(NN)     IS THE CONTACT STRESS VECTOR.
C     SIGN(M1)    INDICATES THE TYPE OF CONSTRAINT USED.
C                 -1 FOR GE.TYPE OF CONSTRAINT.
C                 0 FOR EQ. TYPE OF CONSTRAINT.
C                 1 FOR LE.TYPE OF CONSTRAINT.
C
C     NOTE THAT
C
C     THE DIMENSIONS OF THE MATRICES AND VECTORS SHOULD BE GIVEN AS
C     SHOWN IN THE BRACKETS.ACTUAL SIZE OF 'HAT' AND 'TELDA' MATRICES
C     AND VECTORS IS GENERALLY LESS.
C     HAT AND TEL ATTACHED TO THE MATRICES AND VECTORS
C     DENOTES ^ AND ~ RESPECTIVELY.
C
C
C     INPUT DATA FOR MAIN PROGRAM OF LP SUBROUTINE.
C
C
C     IST SET OF CARDS. READ IN M1,N1
C     2ND SET OF CARDS. READ IN OR GENERATE MATRIX 'B'
C     3RD SET OF CARDS. READ IN OR GENERATE VECTOR 'RQ'.
C     4TH SET OF CARDS. READ IN SIGN ACCORDING TO TYPE OF CONSTRAINT.
C
C
C     IMPORTENT NOTE.
```

```
C
C
C     IN CASE OF BEAM ON ELASTIC FOUNDATION WITH INITIAL GAP
C     A FEW CHANGES IN THE MAIN PROGRAM ARE NECESSARY.
C
C     VECTOR SSA IS CHANGED.
C
C     SSA IN THE EXAMPLE SOLVED IN THIS WORK IS TAKEN AS GIVEN BELOW.
C
C     SSA(I)=0.0001*(1.-((X(I))**2)/(AL**2))
C
C     FEW CHANGES IN MATRIX 'B' ARE AS FOLLOWS.
C
C     B(I+3*NC,K1)=-AHAT(I,K1)
C 320 B(I+3*NC,K1+MNM)=AHAT(I,K1)
C     B(I+3*NC,MM+I)=-1.
C 330 B(I+3*NC,MM+NC+K1)=-BHAT(I,K1)
C
C
      INTEGER SIGN,GP,PG
      REAL NBP
C
      COMMON B(105,105),RQ(105),NBP(105),SIGN(105),KK(105),ITRN
      COMMON RRQ(105),KN,GP,PG
C
      DIMENSION SHAT(25),SA(25),ASHAT(25),SB(25),K(25),II(25)
      DIMENSION KI1(25),BHAT(25,25),SAHAT(25),AHAT(25,2),SSA(25)
      DIMENSION BTEL(25,25),ATEL(25,2),SATEL(25),SBTEL(25),ASTEL(25)
      DIMENSION K11(25),P(105),RBD(2),RBD1(4)
      DIMENSION T(25,25),A1(25,2),Z(2),SBHAT(25),F(25,25),X(25),CST(25)
C
      READ(5,3) PP,U,AL,DEL,MNM,NN,NC,NNC,NGE
C
    3 FORMAT(4F10.4,5I5)
```

```
      Z(1)=PP
      Z(2)=0.0
      DO 5 J=1,MNM
C
    5 READ(5,2)(A1(I,J),I=1,NN)
C
      DO 20 I=1,NN
   20 SSA(I)=0.
      DO 1 I=1,NN
      X(I)=(I-1)*DEL
    1 SA(I)=U*(1.-(((X(I))**2)/(AL**2)))
      DO 4 I=1,NN
      SB(I)=0.
C
    4 READ(5,2)(F(I,J),J=1,NN)
C
    2 FORMAT(5E15.7)
      DO 22 I=1,NN
      DO 22 J=1,NN
   22 T(I,J)=F(I,J)
      ITRN=1
C
C     K(NC) EQUIVALENT TO KI1(NC) IS THE INDEX OF POINTS IN CONTACT.
C
C
      READ(5,40)(K(I),I=1,NC)
C
   40 FORMAT(40I2)
      DO 25 I=1,NC
   25 KI1(I)=K(I)
  725 CONTINUE
      JJ=0
      DO 35 I=1,NC
   35 K(I)=KI1(I)
```

```
      KO=0
      NM=0
      K2=0
      N12=0
      DO 42 I=1,NN
   42 II(I)=0
      DO 60 I=1,NN
      DO 70 NN1=1,NC
      L=K(NN1)
      IF(I-L)69,100,69
  100 KO=KO+1
      II(L)=1
C
C     TO FORM MATRIX B-'HAT',A-'HAT',SMALL-A-'HAT',SMALL-B-'HAT',
C     AND VECTOR OF INITIAL GAP FOR POINTS IN CONTACT.
C
      DO 220 KUI=1,MNM
  220 AHAT(KO,KUI)=A1(L,KUI)
      SAHAT(KO)=SA(L)
      SBHAT(KO)=SB(L)
      ASHAT(KO)=SSA(L)
      DO 71 J=1,NN
      DO 71 KM=1,NC
      M3=K(KM)
      IF(J-M3)71,200,71
  200 NM=NM+1
      BHAT(KO,NM)=T(I,J)
   71 CONTINUE
      NM=0
      GO TO 70
   69 K2=K2+1
      IF(NC-K2)70,80,70
   80 JJ=JJ+1
C
```

```
C     TO FORM MATRIX B-TELDA(BTEL),A-TELDA(ATEL),SMALL-A-TELDA
C     (ATEL),SMALL-B-TELDA(BTEL),AND VECTOR OF INITIAL GAP FOR
C     POINTS NOT IN CONTACT.
C
      DO 222 MZ=1,MNM
  222 ATEL(JJ,MZ)=A1(I,MZ)
      SATEL(JJ)=SA(I)
    . SBTEL(JJ)=SB(I)
      ASTEL(JJ)=SSA(I)
      K11(JJ)=I
      DC 72 J=1,NN
      DC 72 KM3=1,NC
      M2=K(KM3)
      IF(J-M2)72,250,72
  250 N12=N12+1
      BTEL(JJ,N12)=T(I,J)
   72 CCNTINUE
      N12=0
   70 CCNTINUE
      K2=0
   60 CCNTINUE
  101 FCRMAT(/5E15.7/)
      N1=2*MNM+1+2*NC+NGE
      N=2*MNM+1+2*NC
      M1=1+4*NC+MNM+NNC
C
C     FCRMULATION OF 'B' MATRIX.
C     FCRMULATION OF VECTOR 'RQ'.
C
      DO 300 I=1,M1
      RC(I)=0.
      DC 300 J=1,N1
  300 B(I,J)=0.
C
```

```
C     COST FUNCTION ROW FOR MAXIMIZATION PROBLEM(NBP).
C
      DO 310 I=1,N1
  310 NBP(NGE+I)=0.
      MM=2*MNM+1
      NBP(NGE+MM)=-1.0
      DO 350 I=1,NC
      B(I,MM)=-1.0
      B(I,MM+NC+I)=1.0
      B(I+NC,MM+I)=1.
      B(I+2*NC,MM+I)=-1.
      DO 320 K1=1,MNM
      B(I+3*NC,K1)=AHAT(I,K1)
  320 B(I+3*NC,K1+MNM)=-AHAT(I,K1)
      B(I+3*NC,MM+I)=1.
      DO 330 K1=1,NC
  330 B(I+3*NC,MM+NC+K1)=BHAT(I,K1)
      DO 340 K1=1,MNM
  340 B(4*NC+K1,MM+NC+I)=AHAT(I,K1)
  350 CONTINUE
      DO 360 KJ=1,NNC
      DO 355 K1=1,MNM
      B(KJ+4*NC+MNM,K1)=-ATEL(KJ,K1)
  355 B(KJ+4*NC+MNM,K1+MNM)=ATEL(KJ,K1)
      DO 358 KU=1,NC
  358 B(KJ+4*NC+MNM,MM+NC+KU)=-BTEL(KJ,KU)
  360 CONTINUE
      DO 370 J=1,NC
      RC(J)=0.0
      RC(J+NC)=SAHAT(J)
      RC(J+2*NC)=SAHAT(J)
  370 RC(J+3*NC)=ASHAT(J)
      DO 375 I1=1,MNM
  375 RC(4*NC+I1)=Z(I1)
```

```
      DO 380 IJ =1,NNC
  380 RC(4*NC+MNM+IJ)=SBTEL(IJ)+ASTEL(IJ)
C
C     SIGN INDICATES THE TYPE OF CONSTRAINTS.
C     ONE FOR LESS THAN EQUAL TO TYPE OF CONSTRAINT.
C     ZERO FOR EQUALITY CONSTRAINT.
C     MINUS ONE FOR GREATER THAN EQUAL TYPE OF CONSTRAINT.
C
      SIGN(M1)=0
      MK=3*NC
      DO 399 I=1,MK
  399 SIGN(I)=1
      KM1=MK+1
      KM2=MK+NC+MNM
      DO 420 I=KM1,KM2
  420 SIGN(I)=0
      KM3=KM2+1
      KM4=KM2+NNC
      DO 430 I=KM3,KM4
  430 SIGN(I)=1
C
C     ALL THE COEFFICIENT IN THE LP TABLEAU START LEAVING NGE-
C     COLUMNS BLANK.
C
      DO 600 I=1,M1
      DO 600 J=1,N
  600 B(I,NGE+J)=B(I,J)
      GP=0
      PG=0
C
      CALL LINP (M1,N,NGE)
C
C     CHECK IF SOLUTION IS UNBOUNDED OR INFEASIBLE THEN TERMINATE.
C
```

```
      IF(GP.EQ.1) GO TO 1000
      IF(PG.EQ.1) GO TO 1000
      DO 308 J=1,KN
C
C     REARRANGING THE INDICIES OF THE VARIABPES IN A SERIAL ORDER.
C
      DO 308 I=2,KN
      P(J)=KK(J)
      IF(J-I)307,308,308
  307 IF(P(J)-KK(I))308,308,299
  299 KK(J)=KK(I)
      KK(I)=P(J)
  308 CONTINUE
      DO 298 I=1,KN
  298 RQ(KK(I))=RRQ(KK(I))
C
C     RBD IS RIGID BODY DISPLACEMENT.
C
      IM=2*MNM
      DO 290 I=1,IM
  290 RBD1(I)=0.
C
C     TO FIND THE RIGID BODY DISPLACEMENTS (SMALL-'Q')
C
      DO 205 I=1,IM
      IF(KK(I)-IM)199,199,205
  199 KX=I
      IF(KK(I).LE.2)GO TO 202
      GO TO 275
  202 IF(KK(I).EQ.2) GO TO 280
      GO TO 285
  280 RBD1(2)=-RQ(KK(I))
      GO TO 205
  285 RBD1(1)=RQ(KK(I))
```

```
      GO TO 205
  275 IF(IM.EQ.2) GO TO 205
  282 IF(KK(I).EQ.3) GO TO 283
      GO TO 284
  283 RBD1(3)=RQ(KK(I))
      GO TO 205
  284 IF(KK(I).EQ.4)GO TO 279
      GO TO 205
  279 RBD1(4)=-RQ(KK(I))
  205 CONTINUE
      N21=0
      DO 281 J=1,MNM
      RBD(J)=RBD1(J+N21)+RBD1(J+1+N21)
      N21=N21+1
  281 WRITE(6,214) N21,RBD(J)
  214 FORMAT(/2X,'RBD(',I1,')=',E15.7/)
      KX=KX+1
      FX2=RQ(KK(KX))
C
C     TO GET THE CONTOUR DESIGN VARIABLES.
C     CONTOUR DESIGN VARIABLES START WITH INDEX OF RQ AS 2*MNM+2.
C
      J1=NC+MM
      DO 210 I=1,NC
      J2=KX+1
      JC=I+MM
      IF(KK(J2)-J1)206,206,221
  206 CONTINUE
      IF(KK(J2).EQ.JO) GO TO 203
   .  GO TO 221
  203 KX=KX+1
      SB(K(I))=RQ(KK(J2))
      GO TO 210
  221 SB(K(I))=0.0
```

```
  210 CONTINUE
      WRITE(6,226) FX2
  226 FORMAT(/2X,'COST FN=',E15.7/)
      NCI=NC
      WRITE(6,211)
  211 FORMAT(/1X,'SR.NO.',3X,'POINT',9X,'SHAT',15X,'EPHAT',15X,'SBHAT')
C
C     CONTACT STRESS VECTOR FORMATION.
C
      DO 224 I=1,NC
      JK=KX+1
      K12=MM+NC+I
      IF(KK(JK).EQ.K12) GO TO 277
      GO TO 278
  277 KX=KX+1
      SHAT(I)=RQ(K12)
      GO TO 224
  278 SHAT(I)=0.0
  224 CONTINUE
C
C     COMPUTE EPSLON 'HAT'.
C
      DO 225 I=1,NC
      EPHAT=SB(K(I))+ASHAT(I)
      DO 208 K8=1,MNM
  208 EPHAT=EPHAT+AHAT(I,K8)*RBD(K8)
      DO 209 J=1,NC
  209 EPHAT=EPHAT+BHAT(I,J)*SHAT(J)
  225 WRITE(6,223) I,K(I),SHAT(I),EPHAT,SB(K(I))
  223 FORMAT(/5X,I2,3X,I2,3X,E15.7,5X,E15.7,5X,E15.7)
C
C     COMPUTE EPSLON 'TELDA'.
C     CHECK WHICH OF THE EPSLON TELDA IS ZERO.
C
```

```
      WRITE(6,235)
  235 FORMAT(/5X,'POINT',8X,'EPSLON TELDA'/)
      DO 230 I=1,NNC
      EPTL=SBTEL(I)+ASTEL(I)
      DO 236 K7=1,MNM
  236 EPTL=EPTL+ATEL(I,K7)*RBD(K7)
      DO 240J=1,NC
  240 EPTL=EPTL+BTEL(I,J)*SHAT(J)
      WRITE(6,231) K11(I),EPTL
      IF(EPTL.LE.1E-04) GO TO 232
      GO TO 230
  232 II(K11(I))=1
  230 CONTINUE
  231 FORMAT(/5X,I5,5X,E15.7)
C
C     S-HAT IS THE VECTOR OF CONTACT FORCES.
C     CHECK WHICH OF THE S-HAT ARE ZERO.
C     POINTS WHERE S-HAT IS ZERO ,IS LIFTED.
C     KNC ARE THE NO. OF PTS. LIFTED.
C
      KNC=0
  251 FORMAT(/2X,'KNC-NO.OF PTS. WHICH CAN BE LIFTED=',I3)
      DO 259 I=1,NC
      IF(SHAT(I))259,249,259
  249 KNC=KNC+1
      II(K(I))=0
  259 CONTINUE
      WRITE(6,251)KNC
      NC=0
      DO 270 I=1,NN
      IF(II(I).EQ.0)GO TO 270
      NC=NC+1
      KI1(NC)=I
  270 CONTINUE
```

```
      NNC=NN-NC
      WRITE(6,255) NC,NNC
  255 FORMAT(/2X,'NC=',I3,2X,'NNC=',I3/)
C
C     COMPARE THE CONTOUR BETWEEN THE TWO CONSECTIVE ITRNS.
C
      IF(NCI-NC) 725,800,725
  800 CONTINUE
      JIK=0
      DO 825 I=1,NC
      IF(K(I)-KI1(I)) 728,825,728
  728 JIK=JIK+1
  825 K(I)=KI1(I)
      IF(JIK.EQ.0) GO TO 700
      GO TO 725
  700 WRITE(6,860)
  860 FORMAT(/5X,'FINAL RESULTS'/)
      WRITE(6,875) FX2
  875 FORMAT(/1X,'COST FN=',F12.6/)
      WRITE(6,405)
  405 FORMAT(/20X,'INITIAL GAP'/)
      WRITE(6,2)(SSA(I),I=1,NN)
      WRITE(6,403)
  403 FORMAT(/20X,'TOLERANCE LIMIT'/)
      WRITE(6,2)(SA(I),I=1,NN)
      WRITE(6,880)
  880 FORMAT(/5X,'POINT',7X,'FINAL  GAP   ',7X'CONTACT FORCE',6X        ,
     1'CONTACT STRESS')
      DO 900 I=1,NC
      CST(I)=SHAT(I)/DEL
  900 WRITE(6,890) K(I),SB(K(I)),SHAT(I),CST(I)
  890 FORMAT(/5X,I5,5X,E15.7,5X,E15.7,5X,E15.7)
 1000 CALL EXIT
      END
```

```
C     ***********************************
C     *                                 *
C     * SUBROUTINE LINEAR PROGRAMMING.  *
C     *                                 *
C     ***********************************
C
      SUBROUTINE LINP (M1,N,NGE)
C
      INTEGER  RNM1,RNM2,CLNM1,CLNM2,BLNK                           ,LPFTN  1
     1       IBN1(105),IBN2(105),NBN1(105),NBN2(105)
C
      INTEGER SIGN,GP,PG
      REAL   PIVOT,LST,XNBP,FN,CJBAR,X,VALUE,BP(105),PI(105),XPI(105)
      REAL NBP
C
      COMMON B(105,105),RQ(105),NBP(105),SIGN(105),KK(105),ITRN
      COMMON RRQ(105),KN,GP,PG
C                                                                    LPFTN  5
      DATA BLNK/4H    /
      DATA NM1,NM2/'CCCC','AAAA'/
C                                                                    LPFTN  8
C                                                                    LPFTN 16
C     INPUT PROGRAM                                                  LPFTN 17
C                                                                    LPFTN 18
      NI = 5
      NC = 6
      IN=1
      M=M1-1
      DO 101 I=1,M
      IF(SIGN(I))108,107,106
  106 BP(I)=0.
      GO TO 101
  108 BP(I)=-1.0
      B(I,IN)=-1.0
```

```
      NBN1(IN)=NM2
      NBN2(IN)=IN
      NBP(IN)=0.
      IN=IN+1
      GC TO 101
  107 BP(I)=-2.0
  101 CCNTINUE
      DC 102 J=1,N
      NBN1(J+NGE)=NM1
 102  NBN2(J+NGE)=J+NGE
      N=N+NGE
      DC 10 I=1,M
      IF(BP(I)+1.0) 19,11,12
   11 IBN1(I)=BLNK
      IBN2(I)=BLNK
      GC TO 10
   19 BP(I)=-1.0
      GC TO 11
   12 CCNTINUE
      IBN1(I)=NM2
      IBN2(I)=I
   10 CCNTINUE
C                                                                       LPFTN164
C     ACCUMULATE COUNT OF INFEASIBILITIES                               LPFTN165
C                                                                       LPFTN166
      NINF =0                                                           LPFTN167
      DO 6000 I=1,M                                                     LPFTN168
      IF(BP(I))6001,6000,6000                                           LPFTN169
 6001 NINF = NINF+1                                                     LPFTN170
 6000 CCNTINUE
C                                                                       LPFTN172
C      GENERATE INDICATORS FOR MINIMIZATION OF INFEASIBILITY            LPFTN173
C                                                                       LPFTN174
      DO 6101 J=1,N                                                     LPFTN175
```

```
      XPI(J) =0.                                                        LPFTN176
      DO 6101 I=1,M                                                     LPFTN177
      IF(BP(I))6102,6101,6101                                           LPFTN178
 6102 XPI(J) = XPI(J)-B(I,J)                                            LPFTN179
 6101 CCNTINUE                                                          LPFTN180
      DC 6002 I=1,M                                                     LPFTN181
 6002 BP(I) = 0.                                                        LPFTN182
      WRITE(6,401)
  401 FORMAT(/2X,'************************'/)
      WRITE(6,400) ITRN
  400 FORMAT(/5X,'MAIN ITRN NO.=',I3/)
      WRITE(6,402)
  402 FCRMAT(/2X,'************************'/)
      ITRN=ITRN+1
      IPHASE = 1                                                        LPFTN183
C                                                                       LPFTN184
C     MAIN ROUTINE                                                      LPFTN185

 9201 WRITE(NO,9202)
 9202 FORMAT ('0 ITERATION    VAR IN      VAR OUT            OBJ FN',/)  LPFTN188
      IT = 0                                                            LPFTN189
54325 CCNTINUE                                                          LPFTN190
C                                                                       LPFTN191
C      CALCULATE SHADOW PRICES                                          LPFTN192
C                                                                       LPFTN193
      DC 194 J=1,N                                                      LPFTN194
      PI(J) = -NBP(J)                                                   LPFTN195
      DC 194 I=1,M                                                      LPFTN196
  194 PI(J) = PI(J) + BP(I)*B(I,J)                                      LPFTN197
C                                                                       LPFTN198
C      SELECT BEST NONBASIS VECTOR                                      LPFTN199
C                                                                       LPFTN200
 9101 LST = -.0000001
      KCOL = 0                                                          LPFTN202
```

```
      GO TO (751,552),IPHASE                                            LPFTN203
  751 IF(NINF)54321,54321,552                                           LPFTN204
  552 CONTINUE                                                          LPFTN205
      DO 9102 J=1,N                                                     LPFTN206
C                                                                       LPFTN207
C      IGNORE ARTIFICIAL VARIABLES                                      LPFTN208
C                                                                       LPFTN209
      IF(NBN1(J)-BLNK+NBN2(J)-BLNK)651,9102,651                         LPFTN210
  651 CCNTINUE                                                          LPFTN211
      GC TO (6003,6004),IPHASE                                          LPFTN212
 6003 IF(XPI(J)-LST)6005,6006,6006                                      LPFTN213
 6005 KCOL=J                                                            LPFTN214
      LST = XPI(J)                                                      LPFTN215
      GC TO 9102
 6004 CCNTINUE                                                          LPFTN217
      IF(PI(J)-LST)9103,9102,9102                                       LPFTN218
 9103 KCOL = J                                                          LPFTN219
      LST = PI(J)                                                       LPFTN220
 6006 CCNTINUE                                                          LPFTN221
 9102 CCNTINUE                                                          LPFTN222
      IF (KCOL)54321,54321,9104                                         LPFTN223
C                                                                       LPFTN224
C      DETERMINE KEYROW                                                 LPFTN225
C                                                                       LPFTN226
 9104 KROW = 0                                                          LPFTN227
      CJBAR = LST                                                       LPFTN228
      LST = 1.0E20                                                      LPFTN229
      DC 9105 I=1,M                                                     LPFTN230
      IF(B(I,KCOL))9105,9105,9106
 9106 RATIO = RQ(I)/B(I,KCOL)                                           LPFTN232
      IF (RATIO-LST)9107,9105,9105                                      LPFTN233
 9107 LST = RATIO                                                       LPFTN234
      KROW=I                                                            LPFTN235
 9105 CONTINUE                                                          LPFTN236
```

```
      IF(KROW)9112,9112,9114                                            LPFTN237
 9112 WRITE(NO,9113) NBN1(KCOL),NBN2(KCOL)
9113  FORMAT(' VARIABLE ',A2,I3,'  UNBOUNDED ')
      GP=GP+1
      GO TO 54323                                                       LPFTN240
 9114 CONTINUE                                                          LPFTN241
C                                                                       LPFTN242
C      TRANSFORM                                                        LPFTN243
C                                                                       LPFTN244
C        DIVIDE BY PIVOT                                                LPFTN245
      PIVOT = B(KROW,KCOL)
      DO 9108 J=1,N                                                     LPFTN247
 9108 B(KROW,J) = B(KROW,J)/PIVOT                                       LPFTN248
      RQ(KROW) = RQ(KROW)/PIVOT                                         LPFTN249
      DO 9109 I=1,M                                                     LPFTN250
      IF(I-KROW)9110,9109,9110                                          LPFTN251
 9110 RQ(I) = RQ(I) - RQ(KROW)*B(I,KCOL)                                LPFTN252
      DO 4444 J = 1, N
      IF(J-KCOL)9111,4444,9111                                          LPFTN254
 9111 B(I,J) = B(I,J) - B(KROW,J)*B(I,KCOL)                             LPFTN255
 4444 CONTINUE
 9109 CONTINUE                                                          LPFTN256
      DO 9300 I=1,M                                                     LPFTN257
 9300 B(I,KCOL)  = -B(I,KCOL)/PIVOT                                     LPFTN258
      B(KROW,KCOL) = 1.0/PIVOT                                          LPFTN259
C                                                                       LPFTN260
C      INTERCHANGE BASIS AND NONBASIS VARIABLES                         LPFTN2
C                                                                       LPFTN262
      RNM1 = NBN1(KCOL)                                                 LPFTN263
      RNM2 = NBN2(KCOL)                                                 LPFTN264
      NBN1(KCOL) = IBN1(KROW)                                           LPFTN265
      NBN2(KCOL) = IBN2(KROW)                                           LPFTN266
      IBN1(KROW) = RNM1                                                 LPFTN267
      IBN2(KROW) = RNM2                                                 LPFTN268
```

```
      LST = NBP(KCOL)                                                   LPFTN269
      NBP(KCOL) = BP(KROW)                                              LPFTN270
      BP(KROW) = LST                                                    LPFTN271
      IT = IT + 1                                                       LPFTN272
      IF(NBN1(KCOL)-BLNK+NBN2(KCOL)-BLNK)6201,6200,6201                 LPFTN273
 6200 NINF = NINF-1                                                     LPFTN274
 6201 CCNTINUE                                                          LPFTN275
C
C      COMPUTE OBJECTIVE FUNCTION                                       LPFTN277
C                                                                       LPFTN278
      FN = 0.                                                           LPFTN279
      DC 9301 I=1,M                                                     LPFTN280
 9301 FN = FN + BP(I)*RQ(I)                                             LPFTN281
      GC TO (7000,7001),IPHASE                                          LPFTN282
 7000 SAVE = PI(KCOL)                                                   LPFTN283
      DO 7003 J=1,N                                                     LPFTN284
      PI(J) = PI(J) - SAVE*B(KROW,J)                                    LPFTN285
      XPI(J) = XPI(J) - CJBAR*B(KROW,J)                                 LPFTN286
 7003 CCNTINUE                                                          LPFTN287
      PI(KCOL) = -SAVE/PIVOT                                            LPFTN288
      XPI(KCOL) = -CJBAR/PIVOT                                          LPFTN289
      GC TO 7004                                                        LPFTN290
 7001 CCNTINUE
      DC 9302 J=1,N                                                     LPFTN292
 9302 PI(J) = PI(J) - CJBAR*B(KROW,J)                                   LPFTN293
      PI(KCOL) = -CJBAR/PIVOT                                           LPFTN294
 7004 CCNTINUE                                                          LPFTN295
C           CHECK FOR ESSENTIAL ZERO                                    LPFTN296
      DO 6111 I=1,M                                                     LPFTN297
      DO 6111 J=1,N                                                     LPFTN298
      X=B(I,J)                                                          LPFTN299
      IF(ABS(X)-.0000001)6112,6112,6111                                 LPFTN300
 6112 B(I,J) = 0.                                                       LPFTN301
 6111 CONTINUE                                                          LPFTN302
```

```
C                                                                    LPFTN303
C                                                                    LPFTN304
C      LOG ITERATION                                                 LPFTN305
C
      WRITE(NO,9120)IT,IBN1(KROW),IBN2(KROW),NBN1(KCOL),NBN2(KCOL),FN
 9120 FORMAT(I9,7X,A2,I3,8X,A2,I3,3X,F13.3)
      GO TO 9101                                                     LPFTN309
C                                                                    LPFTN310
C                                                                    LPFTN311
54321 CONTINUE                                                       LPFTN312
      IF(IPHASE-1)8000,8000,54322                                    LPFTN313
 8000 IPHASE = 2                                                     LPFTN314
      IF(NINF)8003,8003,8004                                         LPFTN315
 8004 WRITE(NO,8005)
 8005 FORMAT('0 SOLUTION INFEASIBLE',/)                              LPFTN317
      PG=PG+1
      GO TO 54323
 8003 CONTINUE                                                       LPFTN319
      WRITE(NO,8002)
 8002 FORMAT('0  SOLUTION FEASIBLE ',/)                              LPFTN321
      GO TO 54325
54322 CONTINUE                                                       LPFTN323
C                                                                    LPFTN324
C      OUTPUT ROUTINE                                                LPFTN325
C                                                                    LPFTN326
      WRITE(NO,301) IT,FN
  301 FORMAT('1','  ITERATION',I5,'  OBJ FN ',F15.3/)               LPFTN328
      WRITE(NO,302)
  302 FORMAT(3X,'BASIS VAR',17X,'AMOUNT')
      KN=0
      DO 3033 I=1,M                                                  LPFTN332
C                                                                    LPFTN333
C     COST RANGING                                                   LPFTN334
C                                                                    LPFTN335
```

```
      VALUE = 1.0E20                                                    LPFTN336
      LST = 1.0E20                                                      LPFTN337
      DC 12300 J=1,N                                                    LPFTN338
      IF(NBN1(J)-BLNK+NBN2(J)-BLNK)12305,12300,12305                    LPFTN339
12305 CCNTINUE                                                          LPFTN340
      IF(B(I,J))12301,12300,12302                                       LPFTN341
12302 X=PI(J)/B(I,J)                                                    LPFTN342
      IF(X-LST)12303,12300,12300                                        LPFTN343
12303 LST=X                                                             LPFTN344
      GC TO 12300                                                       LPFTN345
12301 X=-PI(J)/B(I,J)                                                   LPFTN346
      IF(X-VALUE)12304,12300,12300                                      LPFTN347
12304 VALUE = X                                                         LPFTN348
12300 CCNTINUE                                                          LPFTN349
      LST = BP(I) - LST                                                 LPFTN350
      VALUE = BP(I) + VALUE                                             LPFTN351
C
C     VARIABLES WITH NAMES AS 'CC' ARE SEPERATED .
C     THESE VARIABLES ARE THE DESIGN VARIABLES OF LP PROBLEM.
C
C
C     VARIABLES WITH NAMES 'AA' ARE ELIMINATED IN THE NEXT STEP.
C     THESE ARE THE SLACK VARIABLES.
C
      IF(IBN1(I)-NM1)3033,306,3033
  306 KN=KN+1
      KK(KN)=IBN2(I)
      RRQ(KK(KN))=RQ(I)
      WRITE(6,304)IBN1(I),IBN2(I),RQ(I)
 3033 CCNTINUE
  304 FCRMAT(7X,A2,I3,7X,F16.6)
54323 CCNTINUE                                                          LPFTN 19
      RETURN
      END                                                               LPFTN365
```

```
C     ***************
C     *             *
C     * INPUT DATA. *
C     *             *
C     ***************

    P          U          AL         DEL      MNM  NN   NC   NNC  NGE

  1000.     0.01       1.44       0.06         1   25   12   13    0

          A1-MATRIX.

  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01
  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01
  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01
  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01
  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01  0.1000000E 01

          T-MATRIX.

          ROW  1

  0.5000000E-04  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00
  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00
  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00
  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00
  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00  0.0000000E 00

          ROW  2

  0.0000000E 00  0.5002033E-04  0.5082351E-07  0.8131758E-07  0.1118117E-06
  0.1423058E-06  0.1728000E-06  0.2032941E-06  0.2337882E-06  0.2642823E-06
  0.2947765E-06  0.3252706E-06  0.3557647E-06  0.3862589E-06  0.4167530E-06
  0.4472471E-06  0.4777413E-06  0.5082351E-06  0.5387294E-06  0.5692230E-06
```

0.5997176E-06	0.6302112E-06	0.6607058E-06	0.6911994E-06	0.7216940E-06

ROW 3

0.0000000E 00	0.5082351E-07	0.5016263E-04	0.2846118E-06	0.4065882E-06
0.5285647E-06	0.6505411E-06	0.7725175E-06	0.8944938E-06	0.1016469E-05
0.1138446E-05	0.1260422E-05	0.1382399E-05	0.1504375E-05	0.1626352E-05
0.1748329E-05	0.1870305E-05	0.1992281E-05	0.2114258E-05	0.2236233E-05
0.2358211E-05	0.2480186E-05	0.2602163E-05	0.2724139E-05	0.2846116E-05

ROW 4

0.0000000E 00	0.8131758E-07	0.2846118E-06	0.5054889E-04	0.8233413E-06
0.1097787E-05	0.1372235E-05	0.1646682E-05	0.1921129E-05	0.2195577E-05
0.2470023E-05	0.2744470E-05	0.3018918E-05	0.3293365E-05	0.3567811E-05
0.3842259E-05	0.4116706E-05	0.4391150E-05	0.4665600E-05	0.4940044E-05
0.5214493E-05	0.5488939E-05	0.5763387E-05	0.6037832E-05	0.6312281E-05

ROW 5

0.0000000E 00	0.1118117E-06	0.4065882E-06	0.8233413E-06	0.5130107E-04
0.1788988E-05	0.2276894E-05	0.2764800E-05	0.3252706E-05	0.3740612E-05
0.4228518E-05	0.4716424E-05	0.5204330E-05	0.5692236E-05	0.6180142E-05
0.6668048E-05	0.7155954E-05	0.7643855E-05	0.8131765E-05	0.8619662E-05
0.9107576E-05	0.9595473E-05	0.1008339E-04	0.1057128E-04	0.1105920E-04

ROW 6

0.0000000E 00	0.1423058E-06	0.5285647E-06	0.1097787E-05	0.1788988E-05
0.5254117E-04	0.3303528E-05	0.4065882E-05	0.4828235E-05	0.5590588E-05
0.6352941E-05	0.7115294E-05	0.7877644E-05	0.8639997E-05	0.9402351E-05
0.1016470E-04	0.1092706E-04	0.1168940E-04	0.1245176E-04	0.1321410E-04
0.1397646E-04	0.1473880E-04	0.1550115E-04	0.1626350E-04	0.1702584E-04

ROW 7

0.0000000E 00	0.1728000E-06	0.6505411E-06	0.1372235E-05	0.2276894E-05
0.3303528E-05	0.5439114E-04	0.5488941E-05	0.6586730E-05	0.7684514E-05
0.8782305E-05	0.9880087E-05	0.1097788E-04	0.1207567E-04	0.1317344E-04
0.1427124E-04	0.1536903E-04	0.1646680E-04	0.1756460E-04	0.1866238E-04
0.1976016E-04	0.2085796E-04	0.2195574E-04	0.2305352E-04	0.2415132E-04

ROW 8

0.0000000E 00	0.2032941E-06	0.7725175E-06	0.1646682E-05	0.2764800E-05
0.4065882E-05	0.5488941E-05	0.5697299E-04	0.8467199E-05	0.9961411E-05
0.1145562E-04	0.1294982E-04	0.1444404E-04	0.1593825E-04	0.1743247E-04
0.1892669E-04	0.2042089E-04	0.2191508E-04	0.2340930E-04	0.2490351E-04
0.2639773E-04	0.2789193E-04	0.2938615E-04	0.3088034E-04	0.3237458E-04

ROW 9

0.0000000E 00	0.2337882E-06	0.8944938E-06	0.1921129E-05	0.3252706E-05
0.4828235E-05	0.6586730E-05	0.8467199E-05	0.6040865E-04	0.1236027E-04
0.1431190E-04	0.1626351E-04	0.1821514E-04	0.2016676E-04	0.2211839E-04
0.2407002E-04	0.2602163E-04	0.2797325E-04	0.2992488E-04	0.3187648E-04
0.3382812E-04	0.3577973E-04	0.3773137E-04	0.3968297E-04	0.4163462E-04

ROW 10

0.0000000E 00	0.2642823E-06	0.1016469E-05	0.2195577E-05	0.3740612E-05
0.5590588E-05	0.7684514E-05	0.9961411E-05	0.1236027E-04	0.6482012E-04
0.1729015E-04	0.1976016E-04	0.2223020E-04	0.2470022E-04	0.2717024E-04
0.2964027E-04	0.3211029E-04	0.3458030E-04	0.3705033E-04	0.3952034E-04
0.4199038E-04	0.4446038E-04	0.4693042E-04	0.4940042E-04	0.5187046E-04

ROW 11

0.0000000E 0C	0.2947765E-06	0.1138446E-05	0.2470023E-05	0.4228518E-05
0.6352941E-05	0.8782305E-05	0.1145562E-04	0.1431190E-04	0.1729015E-04
0.7032939E-04	0.2337880E-04	0.2642823E-04	0.2947764E-04	0.3252705E-04
0.3557646E-04	0.3862588E-04	0.4167526E-04	0.4472470E-04	0.4777408E-04
0.5082351E-04	0.5387289E-04	0.5692233E-04	0.5997172E-04	0.6302114E-04

ROW 12

0.0000000E 00	0.3252706E-06	0.1260422E-05	0.2744470E-05	0.4716424E-05
0.7115294E-05	0.9880087E-05	0.1294982E-04	0.1626351E-04	0.1976016E-04
0.2337880E-04	0.7705843E-04	0.3074821E-04	0.3443801E-04	0.3812779E-04
0.4181759E-04	0.4550737E-04	0.4919713E-04	0.5288694E-04	0.5657670E-04
0.6026651E-04	0.6395628E-04	0.6764609E-04	0.7133585E-04	0.7502566E-04

ROW 13

0.0000000E 00	0.3557647E-06	0.1382399E-05	0.3018918E-05	0.5204330E-05
0.7877644E-05	0.1097788E-04	0.1444404E-04	0.1821514E-04	0.2223020E-04
0.2642823E-04	0.3074821E-04	0.8512923E-04	0.3952038E-04	0.4391152E-04
0.4830268E-04	0.5269384E-04	0.5708495E-04	0.6147614E-04	0.6586725E-04
0.7025844E-04	0.7464956E-04	0.7904074E-04	0.8343186E-04	0.8782303E-04

ROW 14

0.0000000E 00	0.3862589E-06	0.1504375E-05	0.3293365E-05	0.5692236E-05
0.8639997E-05	0.1207567E-04	0.1593825E-04	0.2016676E-04	0.2470022E-04
0.2947764E-04	0.3443801E-04	0.3952038E-04	0.9466373E-04	0.4981723E-04
0.5497073E-04	0.6012425E-04	0.6527771E-04	0.7043124E-04	0.7558471E-04
0.8073825E-04	0.8589170E-04	0.9104525E-04	0.9619871E-04	0.1013523E-03

ROW 15

0.0000000E 00	0.4167530E-06	0.1626352E-05	0.3567811E-05	0.6180142E-05
0.9402351E-05	0.1317344E-04	0.1743247E-04	0.2211839E-04	0.2717024E-04

0.3252705E-04	0.3812779E-04	0.4391152E-04	0.4981723E-04	0.1057839E-03
0.6176077E-04	0.6773762E-04	0.7371441E-04	0.7969131E-04	0.8566810E-04
0.9164499E-04	0.9762179E-04	0.1035987E-03	0.1095755E-03	0.1155524E-03

ROW 16

0.0000000E 00	0.4472471E-06	0.1748329E-05	0.3842259E-05	0.6668048E-05
0.1016470E-04	0.1427124E-04	0.1892669E-04	0.2407002E-04	0.2964027E-04
0.3557646E-04	0.4181759E-04	0.4830268E-04	0.5497073E-04	0.6176077E-04
0.1186118E-03	0.7547297E-04	0.8233408E-04	0.8919531E-04	0.9605644E-04
0.1029176E-03	0.1097788E-03	0.1166400E-03	0.1235010E-03	0.1303622E-03

ROW 17

0.0000000E 00	0.4777413E-06	0.1870305E-05	0.4116706E-05	0.7155954E-05
0.1092706E-04	0.1536903E-04	0.2042089E-04	0.2602163E-04	0.3211029E-04
0.3862588E-04	0.4550737E-04	0.5269384E-04	0.6012425E-04	0.6773762E-04
0.7547297E-04	0.1332693E-03	0.9107571E-04	0.9888226E-04	0.1066886E-03
0.1144952E-03	0.1223016E-03	0.1301082E-03	0.1379146E-03	0.1457212E-03

ROW 18

0.0000000E 00	0.5082351E-06	0.1992281E-05	0.4391150E-05	0.7643855E-05
0.1168940E-04	0.1646680E-04	0.2191508E-04	0.2797325E-04	0.3458030E-04
0.4167526E-04	0.4919713E-04	0.5708495E-04	0.6527771E-04	0.7371441E-04
0.8233408E-04	0.9107571E-04	0.1498782E-03	0.1086910E-03	0.1175037E-03
0.1263166E-03	0.1351293E-03	0.1439421E-03	0.1527549E-03	0.1615677E-03

ROW 19

0.0000000E 00	0.5387294E-06	0.2114258E-05	0.4665600E-05	0.8131765E-05
0.1245176E-04	0.1756460E-04	0.2340930E-04	0.2992488E-04	0.3705033E-04
0.4472470E-04	0.5288694E-04	0.6147614E-04	0.7043124E-04	0.7969131E-04
0.8919531E-04	0.9888226E-04	0.1086910E-03	0.1685610E-03	0.1284410E-03

0.1383211E-03	0.1482012E-03	0.1580813E-03	0.1679613E-03	0.1778415E-03

ROW 20

0.0000000E 00	0.5692230E-06	0.2236233E-05	0.4940044E-05	0.8619662E-05
0.1321410E-04	0.1866238E-04	0.2490351E-04	0.3187648E-04	0.3952034E-04
0.4777408E-04	0.5657670E-04	0.6586725E-04	0.7558471E-04	0.8566810E-04
0.9605644E-04	0.1066886E-03	0.1175037E-03	0.1284410E-03	0.1894390E-03
0.1504476E-03	0.1614558E-03	0.1724642E-03	0.1834724E-03	0.1944808E-03

ROW 21

0.0000000E 00	0.5997176E-06	0.2358211E-05	0.5214493E-05	0.9107576E-05
0.1397646E-04	0.1976016E-04	0.2639773E-04	0.3382812E-04	0.4199038E-04
0.5082351E-04	0.6026651E-04	0.7025844E-04	0.8073825E-04	0.9164499E-04
0.1029176E-03	0.1144952E-03	0.1263166E-03	0.1383211E-03	0.1504476E-03
0.2126351E-03	0.1748324E-03	0.1870302E-03	0.1992277E-03	0.2114255E-03

ROW 22

0.0000000E 00	0.6302112E-06	0.2480186E-05	0.5488939E-05	0.9595473E-05
0.1473880E-04	0.2085796E-04	0.2789193E-04	0.3577973E-04	0.4446038E-04
0.5387289E-04	0.6395628E-04	0.7464956E-04	0.8589170E-04	0.9762179E-04
0.1097788E-03	0.1223016E-03	0.1351293E-03	0.1482012E-03	0.1614558E-03
0.1748324E-03	0.2382702E-03	0.2017181E-03	0.2151659E-03	0.2286140E-03

ROW 23

0.0000000E 00	0.6607058E-06	0.2602163E-05	0.5763387E-05	0.1008339E-04
0.1550115E-04	0.2195574E-04	0.2938615E-04	0.3773137E-04	0.4693042E-04
0.5692233E-04	0.6764609E-04	0.7904074E-04	0.9104525E-04	0.1035987E-03
0.1166400E-03	0.1301082E-03	0.1439421E-03	0.1580813E-03	0.1724642E-03
0.1870302E-03	0.2017181E-03	0.2664672E-03	0.2312262E-03	0.2459851E-03

ROW 24

0.0000000E 00	0.6911994E-06	0.2724139E-05	0.6037832E-05	0.1057128E-04
0.1626350E-04	0.2305352E-04	0.3088034E-04	0.3968297E-04	0.4940042E-04
0.5997172E-04	0.7133585E-04	0.8343186E-04	0.9619871E-04	0.1095755E-03
0.1235010E-03	0.1379146E-03	0.1527549E-03	0.1679613E-03	0.1834724E-03
0.1992277E-03	0.2151659E-03	0.2312262E-03	0.2973471E-03	0.2634786E-03

ROW 25

0.0000000E 00	0.7216940E-06	0.2846116E-05	0.6312281E-05	0.1105920E-04
0.1702584E-04	0.2415132E-04	0.3237458E-04	0.4163462E-04	0.5187046E-04
0.6302114E-04	0.7502566E-04	0.8782303E-04	0.1013523E-03	0.1155524E-03
0.1303622E-03	0.1457212E-03	0.1615677E-03	0.1778415E-03	0.1944808E-03
0.2114255E-03	0.2286140E-03	0.2459851E-03	0.2634786E-03	0.3310333E-03

POINTS IN CONTACT OBTAINED FROM QUADRATIC PROGRAMMING.

1 2 3 4 5 6 7 8 9101112