Numerical calculation of the deformation of end-supported pressure rollers : theory and program description

Technische
Universiteit
Eindhoven

Instituut
Wiskundige Dienstverlening
Eindhoven

REPORT IWDE 90-04

Numerical calculation of the deformation of end-supported pressure rollers; theory (*) and program description (**)

S.W. Rienstra (*)
H. Willemsen (**)

May 1990

Den Dolech 2
Postbus 513
5600 MB Eindhoven

Numerical calculation of the deformation of end-supported pressure rollers; theory (*) and program description (**)

by
S.W.Rienstra (*)
and
H.Willemsen (**)

IWDE report 90-04
May 1990.

The present report and computer program are written under contract with OCE BV, Venlo.

Abstract

A mathematical model and corresponding computer program is described for the problem of the deformation two contacting rollers, supported at the ends, and covered by a thin elastic layer.

A simple configuration is solved exactly, while for the general problem scaling laws are derived for some regions of the parameters.

Eventually, the equations are solved by the package "COLSYS"; therefore no numerical analysis is given, apart from an indication of the necessary settings of the various COLSYS-parameters.

1 Introduction

Consider the configuration of a pair of contacting rollers, supported at the ends and covered with a thin compliant layer, as used for (e.g.) paper sheet transport. The finite bending stiffness and the weight of the rollers , the forces and moments applied at the supports, and the elastic properties of the layer result into elastic deformation of the rollers, and hence, in general, a non-uniform contact line.

However, since paper is relatively vulnerable, it is very important to operate with a contact line with a width as constant as possible. To achieve this goal one has to compensate in some way for the deformation. One way to do this is to crown or camber (vary the radius) of one or both rollers. Other possibilities are adjusting the applied bending moments, varying the flexural rigidity, or the elastic layer.

Since the number of parameters is rather large, it has been found unsatisfactory and inadequate to design a suitable configuration by trial and error. Therefore, a numerical solution of an appropriate mathematical model is described here, to allow the designer to experiment numerically, covering a larger range of variations of a larger number of parameters.

The model adopted is similar to the one presented in [1]. This model is described by a system of two non-linear coupled fourth order ordinary differential equations with prescribed boundary values. This system will be solved here by the public domain package "COLSYS" ([2,...,6]), which is based on collocation and splining in subintervals, with automatic grid adaptation.

2 The model

Although the relatively slender and rigid roller is well described by a linear one-dimensional beam model ([1,7]), the behaviour of the contact area is inherently more complicated (nonlinear, three dimensional, etc.). If, however, the two contacting surfaces are assumed to be undeformable but 'transparent', and the resulting 'interpenetration' δ to generate a reaction force (given by some semi-empirical formula), then the reaction due to the elastic contact can be included in the beam formulation [1]. This results into the following differential equations.

2.1 Differential equations

See figure 1. Two rollers of length L are horizontally positioned in the (x,z)-plane, with the gravity directed in the negative z-direction. The position of the centerline of roller 1 is $z = z_1(x)$, and of roller 2, $z = D + z_2(x)$. The upper surface of the lower roller (1) is described by $z = R_1(x) + d_1(x) + z_1(x)$, and the lower surface of the upper roller (2) is $z = -R_2(x) - d_2(x) + z_2(x) + D$. D is defined such, that the two rollers just touch each other in rest position ($z_1 = z_2 \equiv 0$), so

$$D = \max_{0 \le x \le L} \left(R_1(x) + d_1(x) + R_2(x) + d_2(x) \right).$$

The approach

$$\eta(x) = z_1(x) - z_2(x)$$

minus the crown (or camber) function

$$s(x) = D - R_1(x) - d_1(x) - R_2(x) - d_2(x)$$

yields the interpenetration

$$\delta(x) = \eta(x) - s(x)$$

which produces a reaction $P = P(x, \delta)$ from roller 2 on roller 1, and $-P$ from roller 1 on roller 2.
The present model assumes P to be given by some (semi-) empirical formula, for example

$$P = A(x)\delta^{\alpha(x)} \text{ if } \delta \ge 0, \qquad P = 0 \text{ if } \delta < 0$$

Typically, α is of the order of 2.

The weight per length of each roll is given by

$$W_i = W_i(x) \qquad (i = 1, 2)$$

(For a cylinder with outer and inner radii r_o, r_i, and specific gravity ϱg is $W = \pi(r_o^2 - r_i^2)\varrho g$).
Introduce [7] the (linearized) bending moments

$$M_i(x) = EI_i(x)\frac{d^2}{dx^2}z_i(x) \qquad (i = 1, 2),$$

and the shearing forces

$$F_i(x) = -\frac{d}{dx}M_i(x) \qquad (i = 1, 2),$$

where E is Young's modulus, and I_i is the moment of inertia (for a cylinder with outer and inner radii r_o, r_i is $I = \frac{1}{4}\pi(r_o^4 - r_i^4)$).

Equilibrium of forces and moments now requires

$$\frac{d^2}{dx^2}\left(EI_1(x)\frac{d^2}{dx^2}z_1(x)\right) = -W_1(x) - P(x, \delta)$$
$$\frac{d^2}{dx^2}\left(EI_2(x)\frac{d^2}{dx^2}z_2(x)\right) = -W_2(x) + P(x, \delta)$$

Bending moments are applied at the supported ends of roll 1:

$$z_1(0) = 0 \qquad EI_1(0)z_1''(0) = M_1(0)$$
$$z_1(L) = 0 \qquad EI_1(L)z_1''(L) = M_1(L)$$

Bending moments and external forces (perpendicular to the roll) are applied at the ends of roll 2:

$$EI_2(0)z_2''(0) = M_2(0) \qquad (EI_2z_2'')'(0) = -F_2(0)$$
$$EI_2(L)z_2''(L) = M_2(L) \qquad (EI_2z_2'')'(L) = -F_2(L)$$

Note, that since external point forces induce a discontinuity, equal to that force, in the shearing force, $F_2(0)$ and $F_2(L)$ will have opposite sign if both forces act in the same direction: $F_2 = 0$ for $x < 0$ and for $x > L$ ([7]). Specifically, if both forces are applied in downward direction (the usual situation) then

$$F_2(0) > 0 \qquad\qquad F_2(L) < 0$$

Similarly, the reaction forces applied by the supports of roll 1 are equal to $-F_1(0)$ and $F_1(L)$.

It may be instructive to include here, for reference, the equations relevant to the configuration with internal supports (i.e. at $x = x_{1,i},\ x_{2,i} \neq 0,\ \neq L$) A most convenient way to express these point forces and moments is by means of δ-functions (note that this notation δ is only used here for the the δ-function; otherwise it denotes the interpenetration $\eta - s$).

$$(EI_i z_i'')'' = -W_i \mp P + \sum_{j=1}^{2} F_i(x_{ij})\delta(x - x_{ij}) + M_i(x_{ij})\delta'(x - x_{ij}) \qquad (i = 1, 2)$$

$F_1(x_{1j})$ are of course to be determined. At the free ends the boundary conditions $F_i(0) = F_i(L) = M_i(0) = M_i(L) = 0$ hold, whereas $z_1(x_{1j}) = 0$.

3 Non-dimensionalisation and order of magnitude estimates

Since it is usually more convenient for the designer to work with the dimensional problem, the equations will be left in the form as presented above, for the main part of this report. It is, however, to be emphasised here that both for a numerical evaluation and for a proper understanding of the various physical effects **scaling** is crucial. A first step to this is a non-dimensionalisation and order of magnitude estimates of the equations. This is the only way to identify characteristic non dimensional parameters, to isolate the dominating effects, and to recognize possible singular behaviour.

Consider therefor a typical case with, for convenience, all the problem parameters constant and symmetric boundary conditions

$$EI_i z_i^{(4)} = -W_i \mp A(z_1 - z_2)^\alpha$$

$$z_1(.) = 0, \quad EI_i z_i''(.) = M_i, \quad EI_2 z_2'''(.) = \mp F.$$

Introduce

$$\zeta_i = z_i/L, \quad t = x/L, \quad \beta_i = W_i L^3/EI_i$$

$$\gamma_i = AL^{\alpha+3}/EI_i, \quad \mu_i = LM_i/EI_i, \quad f = FL^2/EI_2$$

to obtain

$$\zeta_i^{(4)} = -\beta_i \mp \gamma_i(\zeta_1 - \zeta_2)^\alpha$$

$$\zeta_i(.) = 0, \quad \zeta_i''(.) = \mu_i, \quad \zeta_2'''(.) = \mp f$$

Since the number of parameters is too large to give here an exhaustive summary of all possibilities, we will indicate only some of the most relevant cases.

First, we observe that in any case the typical amplitude of ζ_i' (which of course depends on the problem parameters) should be small enough for the linear beam theory to be valid. It is therefore appropriate to introduce the small parameters ε_1 and ε_2 and typical penetration depth b (a priori unknown) such that

$$\zeta_1 = \varepsilon_1 \bar{\zeta}_1, \quad \zeta_2 = \varepsilon_2 \bar{\zeta}_2 - b$$

Of course, there is no need to introduce b if it is smaller than or of the same magnitude as ε_2. So $b = 0$ if $b \leq \mathcal{O}(\varepsilon_2)$.

Hence we have

$$\varepsilon_1 \overline{\zeta}_1^{(4)} = -\beta_1 - \gamma_1(b + \varepsilon_1\overline{\zeta}_1 - \varepsilon_2\overline{\zeta}_2)^\alpha$$

$$\varepsilon_2 \overline{\zeta}_2^{(4)} = -\beta_2 + \gamma_2(b + \varepsilon_1\overline{\zeta}_1 - \varepsilon_2\overline{\zeta}_2)^\alpha$$

$$\overline{\zeta}_1(.) = 0, \quad \varepsilon_i\overline{\zeta}_i^{(2)}(.) = \mu_i, \quad \varepsilon_2\overline{\zeta}_2^{(3)}(.) = \mp f$$

If we were to apply relatively large moments μ_i, the problem would be dominated by the external moments and $\varepsilon_i = \mu_i$. However, the practical situation here is that the moments are only used to compensate, so $\mu_i \leq \mathcal{O}(\varepsilon_i)$, and ε_i is always dictated by the other factors.

If roller 1 is stiff and the force f is small, then roller 2 rests with only little deformation on the elastic layer:

$$0 = -\beta_2 + \gamma_2 b^\alpha \quad \text{so} \quad b = (\beta_2/\gamma_2)^{1/\alpha}$$

$$\varepsilon_1\overline{\zeta}_1^{(4)} \simeq -\beta_1 - \gamma_1 b^\alpha \quad \text{so} \quad \varepsilon_1 = \beta_1 + \gamma_1\beta_2/\gamma_2$$

under the condition $\varepsilon_1 \ll b$. Furthermore

$$\varepsilon_2\overline{\zeta}_2^{(4)} \simeq \alpha\gamma_2 b^{\alpha-1}(\varepsilon_1\overline{\zeta}_1 - \varepsilon_2\overline{\zeta}_2)$$

so

$$\varepsilon_2 = \varepsilon_1 \qquad \text{if} \qquad \alpha\gamma_2 b^{\alpha-1} = \mathcal{O}(1)$$

or

$$\varepsilon_2 = \alpha\gamma_2 b^{\alpha-1}\varepsilon_1 \qquad \text{if} \qquad \alpha\gamma_2 b^{\alpha-1} \ll 1$$

under the condition that $f \leq \mathcal{O}(\varepsilon_2)$.

An important case practically is with f relatively large and roller 2 sufficiently stiff, so that it dictates

$$\varepsilon_2 = f \quad \text{with} \quad \overline{\zeta}_2^{(3)} \simeq 2t - 1.$$

This implies uniformly loaded rollers with

$$b = \left(\frac{2f + \beta_2}{\gamma_2}\right)^{1/\alpha}$$

$$\varepsilon_1 = \beta_1 + (2f + \beta_2)\gamma_1/\gamma_2 \qquad \text{with} \qquad \overline{\zeta}_1^{(4)} \simeq -1$$

under the conditions $\varepsilon_1 \ll b$ and $\varepsilon_2 \ll b$.
(A constant $\varepsilon_1\overline{\zeta}_1$ rather than b is not possible because of the boundary conditions). If roller 1 is much more bendable than roller 2 it may occur that ε_1 is larger than b. In that case the above estimate for ε_1 is still valid (as it corresponds to the fact that the supports of roll 1 bear both the weight of roll 1 and roll 2 and the pressure force F),but the other estimates have to be reconsidered according to the order of magnitudes of the various parameters.

It is clear that a lot of cases may be analysed in the above way, with analogous results. In many configurations a priori estimates are possible, and sometimes even with approximated full solutions, if the equations simplify sufficiently.

4 The ideal solution $z_1 - z_2 =$ constant

The technically ideal solution would be $\delta = z_1 - z_2$ is constant. This is however, rather difficult to achieve.

Theorem. A solution $\delta =$ constant does not exist unless at least

$$M_2 \frac{d}{dx}\left(\frac{I_1}{I_2}\right) \neq 0$$

at one or both ends.

Proof: Suppose there is a configuration with solution δ is constant. By adding both equations for z_1 and z_2 we obtain

$$(EI_1 z_1'')'' + (EI_2 z_2'')'' = -W_1 - W_2$$

and

$$F_1 + F_2 = F_1(0) + F_2(0) + \int_0^x W_1 + W_2 \, dx'$$

By assumption is

$$F_1 = \frac{I_1}{I_2} F_2 - \left(\frac{I_1}{I_2}\right)' M_2.$$

Therefore

$$\left(1 + \frac{I_1(L)}{I_2(L)}\right) F_2(L) - \left(1 + \frac{I_1(0)}{I_2(0)}\right) F_2(0) =$$
$$(I_1/I_2)'_{x=L} \; M_2(L) - (I_1/I_2)'_{x=0} \; M_2(0) + G$$
$$\text{where } \; G = \int_0^L W_1 + W_2 \, dx' \; .$$

Now since $G > 0$, $F_2(L) < 0$ and $F_2(0) > 0$, it is necessary that
$M_2 \frac{d}{dx}(I_1/I_2)_{x=0,L} \neq 0$.

5 An exact solution

In order to check a numerical solution it is always of vital importance to have a check against an exact analytical solution. Of course an analytical solution involves some simplifications, otherwise the numerical solution was not necessary in the first place. The exact solution we will present here is for the fully linear problem ($\alpha = 1$, P not necessarily positive) with constant coefficients. Although not absolutely necessary, we will here also utilize symmetric boundary conditions.

$$EI_1 z_1^{(4)} = -W_1 - A(z_1 - z_2)$$

$$EI_2 z_2^{(4)} = -W_2 + A(z_1 - z_2)$$

$$z_1(0) = z_1(L) = 0$$

$$EI_1 z_1''(0) = EI_1 z_1''(L) = M_1$$

$$EI_2 z_2''(0) = EI_2 z_2''(L) = M_2$$

$$EI_2 z_2'''(0) = -EI_2 z_2'''(L) = -F$$

We introduce a similar notation as in chapter 3.

$$\zeta_i = z_i/L, \quad t = x/L, \quad \beta_i = W_i L^3/EI_i$$

$$\gamma_i = AL^4/EI_i, \quad \mu = M_i L/EI_i, \quad f = FL^2/EI_2$$

(note that we assume for simplicity $M_1/I_1 = M_2/I_2$) with furthermore (note the slightly different meaning of δ)

$$\delta = \zeta_1 - \zeta_2, \quad \eta = \zeta_1 + \zeta_2$$

$$\sigma = \beta_1 + \beta_2, \quad \lambda = \tfrac{1}{2}\sqrt{2}(\gamma_1 + \gamma_2)^{\frac{1}{4}}, \quad K = \frac{\beta_2 - \beta_1}{\gamma_1 + \gamma_2}$$

to arrive at

$$\eta^{(4)} = -\sigma$$

$$\delta^{(4)} = 4\lambda^4(K - \delta)$$

with

$$\eta(0) + \delta(0) = \eta(1) + \delta(1) = 0$$

$$\eta''(0) \pm \delta''(0) = \eta''(1) \pm \delta''(1) = 2\mu$$

$$\eta'''(0) - \delta'''(0) = -\eta'''(1) + \delta'''(1) = -2f.$$

The solution is readily found to be

$$\eta = -\tfrac{1}{24}\sigma t(t-1)(t^2 - t - 1) + \mu t(t-1) - \delta(0)$$

$$\delta = K + \frac{f + \sigma/4}{\lambda^3} \frac{\cosh(\lambda t)\cos(\lambda t - \lambda) + \cosh(\lambda t - \lambda)\cos(\lambda t)}{\sinh(\lambda) + \sin(\lambda)}$$

6 the program OCE1.FOR

The program OCE1.FOR calculates the deformations (z_1, z_2), the moments (M_1, M_2), and forces (F_1, F_2) of the problem described in chapter 2:

$$\frac{d^2}{dx^2}\left(EI_1(x)\frac{d^2}{dx^2}z_1\right) = -W_1(x) - P(x,\delta)$$
$$\frac{d^2}{dx^2}\left(EI_2(x)\frac{d^2}{dx^2}z_2\right) = -W_2(x) + P(x,\delta)$$

with

$$P = \begin{cases} A(x)\delta^{\alpha(x)} & \text{if } \delta > 0 \\ 0 & \text{if } \delta \leq 0 \end{cases}$$

and boundary conditions at the ends

$$z_1(0) = z_1(L) = 0$$
$$EI_1(0)\frac{d^2}{dx^2}z_1(0) = M_1(0)$$
$$EI_1(L)\frac{d^2}{dx^2}z_1(L) = M_1(L)$$
$$EI_2(0)\frac{d^2}{dx^2}z_2(0) = M_2(0)$$
$$EI_2(L)\frac{d^2}{dx^2}z_2(L) = M_2(L)$$
$$\frac{d}{dx}(EI_2(0)\frac{d^2}{dx^2}z_2(0)) = -F_2(0)$$
$$\frac{d}{dx}(EI_2(L)\frac{d^2}{dx^2}z_2(L)) = -F_2(L)$$

Before solving the numerical problem, the system of 4-th order differential equations is rewritten to a system of four non-linear coupled second order ordinary differential equations

$$\begin{aligned} z_1''(x) &= \frac{u_1(x)}{EI_1(x)} \\ z_2''(x) &= \frac{u_2(x)}{EI_2(x)} \\ u_1''(x) &= -W_1(x) - P(x,\delta) \\ u_2''(x) &= -W_2(x) + P(x,\delta) \end{aligned} \tag{1}$$

where

$$\left.\begin{aligned} u_i(x) &= EI_i(x)\frac{d^2}{dx^2}z_i(x) = M_i(x) \\ u_i'(x) &= \frac{d}{dx}(EI_i(x)\frac{d^2}{dx^2}z_i(x)) = -F_i(x) \end{aligned}\right\} i = 1,2.$$

and with boundary values

$$\begin{array}{ll} z_1(0) = 0 & z_1(L) = 0 \\ u_1(0) = M_1(0) & u_1(L) = M_1(L) \\ u_2(0) = M_2(0) & u_2(L) = M_2(L) \\ u_2'(0) = -F_2(0) & u_2'(L) = -F_2(L) \end{array} \qquad (2)$$

Let us introduce the vectors

$$\underline{\zeta} = [z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2']^T,$$

the solutions of (1), and the vectorfield

$$\underline{\Phi}(x,\underline{\zeta}),$$

the righthand side of the differential equation $(z_1'',\ z_2'',\ u_1'',\ u_2'')^T = \underline{\Phi}$ to be solved, and the functions

$$\Gamma_i(x,\underline{\zeta}) = 0 \qquad \text{for } i = 1,8.$$

At the points χ_i the boundary conditions are given via Γ such that $\underline{\zeta}$ is the solution of (1) with boundary conditions (2) as follows

$$\begin{array}{lll} \Gamma_1(x,\underline{\zeta}) = \zeta_1 & \text{at} & x = \chi_1 = 0 \\ \Gamma_2(x,\underline{\zeta}) = \zeta_3 - M_1(x) & \text{at} & x = \chi_2 = 0 \\ \Gamma_3(x,\underline{\zeta}) = \zeta_7 - M_2(x) & \text{at} & x = \chi_3 = 0 \\ \Gamma_4(x,\underline{\zeta}) = \zeta_8 + F_2(x) & \text{at} & x = \chi_4 = 0 \\ \Gamma_5(x,\underline{\zeta}) = \zeta_1 & \text{at} & x = \chi_5 = L \\ \Gamma_6(x,\underline{\zeta}) = \zeta_3 - M_1(x) & \text{at} & x = \chi_6 = L \\ \Gamma_7(x,\underline{\zeta}) = \zeta_7 - M_2(x) & \text{at} & x = \chi_7 = L \\ \Gamma_8(x,\underline{\zeta}) = \zeta_8 + F_2(x) & \text{at} & x = \chi_8 = L \end{array} \qquad (3)$$

The solution is solved by the new version of COLSYS ([5,6]).

The method used to approximate the solution $\underline{\zeta}$ is collocation at gaussian points, requiring one continuous derivative in the i-th component, i = 1, ..., 4 of $\underline{\Phi}$.
NCOL denotes the number of collocation points (stages) per subinterval and is chosen such that NCOL $\geq$ 2. In this particular case NCOL=3.
A Runge-Kutta-monomial solution representation is utilized.

6.1 input to COLSYS

NCOMP number of differential equations (=4)

M(j) order of the j-th differential equation (=2 for j=1,...,4)

ALEFT left end of interval (=0)

ARIGHT right end of interval (=L, given in the input file)

CHI(j) j-th side condition point χ_j (boundary point). Must have CHI(j) $\leq$ CHI(j+1). All side condition points are set in the program as described in (3).

IPAR an integer array dimensioned at least 11.

A list of the parameters in IPAR and their meaning is:

IPAR(1) = 0 if the problem is linear

= 1 if the problem is nonlinear

IPAR(2) = number of collocation points per subinterval (=NCOL) where max M(I) $\leq$ NCOL $\leq$ 7.
Set in the program: NCOL=3.

IPAR(3) > 0 number of subintervals in the initial mesh (=N).

= 0 then COLSYS sets N = 5.

Set in the program: IPAR(3)=0.

IPAR(4) > 0 number of solution and derivative tolerances (=NTOL).
Set in the program: NTOL=4.

IPAR(5) > 0 dimension of FSPACE. (=NDIMF)
Set in the program: NDIMF=22960.

IPAR(6) > 0 dimension of ISPACE. (=NDIMI)
Set in the program: NDIMI=943.

IPAR(7) output control (=IPRINT)

= -1 for full diagnostic printout

= 0 for selected printout

= 1 for no printout

IPRINT is given in the inputfile.

IPAR(8) = 0 causes COLSYS to generate a uniform initial mesh.

= 1 if the initial mesh is provided by the user. It is defined in FSPACE as follows: the mesh 0=X(1)<X(2)< ... <X(N)<X(N+1)=L will occupy FSPACE(1), ..., FSPACE(N+1). The user needs to supply only the interior mesh points FSPACE(J) = X(J), J = 2, ..., N.

= 2 if the initial mesh is supplied by the user as with IPAR(8)=1, and in addition no adaptive mesh selection is to be done.

Set in the program: IPAR(8)=0.

IPAR(9) (=IGUESS)

= 0 if no initial GUESS for the solution is provided.

= 1 if an initial GUESS is provided by the user in subroutine GUESS.

= 2 if an initial mesh and approximate solution coefficients are provided by the user in FSPACE. (the former and new mesh are the same).

= 3 if a former mesh and approximate solution coefficients are provided by the user in FSPACE, and the new mesh is to be taken twice as coarse; i.e., every second point from the former mesh.

= 4 if in addition to a former initial mesh and approximate solution coefficients, a new mesh is provided in FSPACE as well. (see description of output for further details on IGUESS = 2, 3, and 4.)

Set in the program: IPAR(9)=1.

IPAR(10) = 0 if the problem is regular

= 1 if the first relax factor is RSTART, and the nonlinear iteration does not rely on past covergence (use for an extra sensitive nonlinear problem only).

= 2 if we are to return immediately upon (a) two successive nonconvergences, or (b) after obtaining error estimate for the first time.

Set in the program: IPAR(10)=0.

IPAR(11) = number of fixed points in the mesh other than 0 and L. The code requires that all side condition points other than 0 and L (see description of CHI) be included as fixed points in FIXPNT.

Set in the program: IPAR(11)=0.

LTOL an array of dimension IPAR(4). LTOL(J) = L specifies that the J-th tolerance in TOL controls the error in the L-th component of $\underline{\zeta}$. Also require that $1 \leq$LTOL(1)<LTOL(2)< ... <LTOL(NTOL)≤ 8.

Set in the program:

LTOL(1)=1 (controls z_1), LTOL(2)=3 (controls $u_1 = M_1$)

LTOL(3)=5 (controls z_2), LTOL(4)=7 (controls $u_2 = M_2$).

TOL an array of dimension IPAR(4). TOL(J) is the error tolerance on the LTOL(J) -th component of $\underline{\zeta}$. thus, the code attempts to satisfy for J=1,...,NTOL on each subinterval $|\nu_L - \zeta_L| \leq TOL_J \times |\nu_L| + TOL_J$ if $\underline{\zeta}$ is the approximate solution vector for $\underline{\nu}$, the exact solution.

TOL(1)=...=TOL(4)=EPS, given in the inputfile.

FIXPNT an array of dimension IPAR(11). It contains the points, other than 0 and L, which are to be included in every mesh.
In our case: empty.

ISPACE an integer work array of dimension IPAR(6).

FSPACE a real work array of dimension IPAR(5).

IFLAG the mode of return from COLSYS.

= 1 for normal return

= 0 if the collocation matrix is singular.

=-1 if the expected number of subintervals exceeds storage specifications.

=-2 if the nonlinear iteration has not converged.

=-3 if there is an input data error.

On return from COLSYS, the arrays FSPACE and ISPACE contain information specifying the approximate solution. The user can produce the solution vector $\underline{\zeta}$ at any point x, $0 \leq x \leq L$, by the statement,

```
CALL APPSLN(X, Z, FSPACE, ISPACE)
```

When saving the coefficients for later reference, only ISPACE(1),...,ISPACE(7+4) and FSPACE(1),...,FSPACE(ISPACE(7)) need to be saved as these are the quantities used by APPSLN. A formerly obtained solution can easily be used as the first approximation for the nonlinear iteration for a new problem by setting
(IGUESS =) IPAR(9) = 2, 3 or 4. (in our case: IPAR(9)=2).
If the former solution has just been obtained then the values needed to define the first approximation are already in ISPACE and FSPACE. Alternatively, if the former solution was obtained in a previous run and its coefficients were saved then those coefficients must be put back into ISPACE(1), ..., ISPACE(11) and FSPACE(1), ..., FSPACE(ISPACE(7)).
For IPAR(9) = 2 or 3 set IPAR(3) = ISPACE(1), the size of the previous mesh.

The following subroutines must be declared external in the main program which calls COLSYS.

FSUB	the subroutine calculates the vector field $\underline{\Phi}(x, \underline{\zeta})$ for given values of x and $\underline{\zeta}$
DFSUB	the subroutine calculates the Jacobian matrix of $\underline{\Phi}$, here by numerical differentiation.
GSUB	the subroutine gives the function Γ_i for given values of i and $\underline{\zeta}$ such that $\Phi_i = 0$ in the point $x = \chi_i$ (See (3)).
DGSUB	the subroutine calculates the vector DG, the i-th row of the Jacobian matrix of Γ with $DG_j = \frac{\partial \Gamma_i}{\partial \chi_j}$, $(j = 1, ..., 8)$
GUESS	the subroutine calculates an initial guess for $\underline{\zeta}$ and $\underline{\Phi}$.

The following routines depend on the model and can be changed by the user.

P	the reaction of the elastic layer. Example: $P(x) = A(x)\delta^{\alpha(x)}$ with $A(x)$ and $\alpha(x)$ are chosen by the user.
DELTA	the interpenetration δ of the rollers, defined as $\delta(x) = z_1(x) - z_2(x) + D - (R_1(x) + R_2(x) + D_1(x) + D_2(x))$
I1	the moment of inertia of roller 1
I2	the moment of inertia of roller 2
D1	thickness of the elastic layer of roller 1
D2	thickness of the elastic layer of roller 2
R1	the radius of roller 1
R2	the radius of roller 2
W1	the weight density of roller 1
W2	the weight density of roller 2
DD	$D = \max(R_1 + R_2 + D_1 + D_2)$

6.2 the inputfile

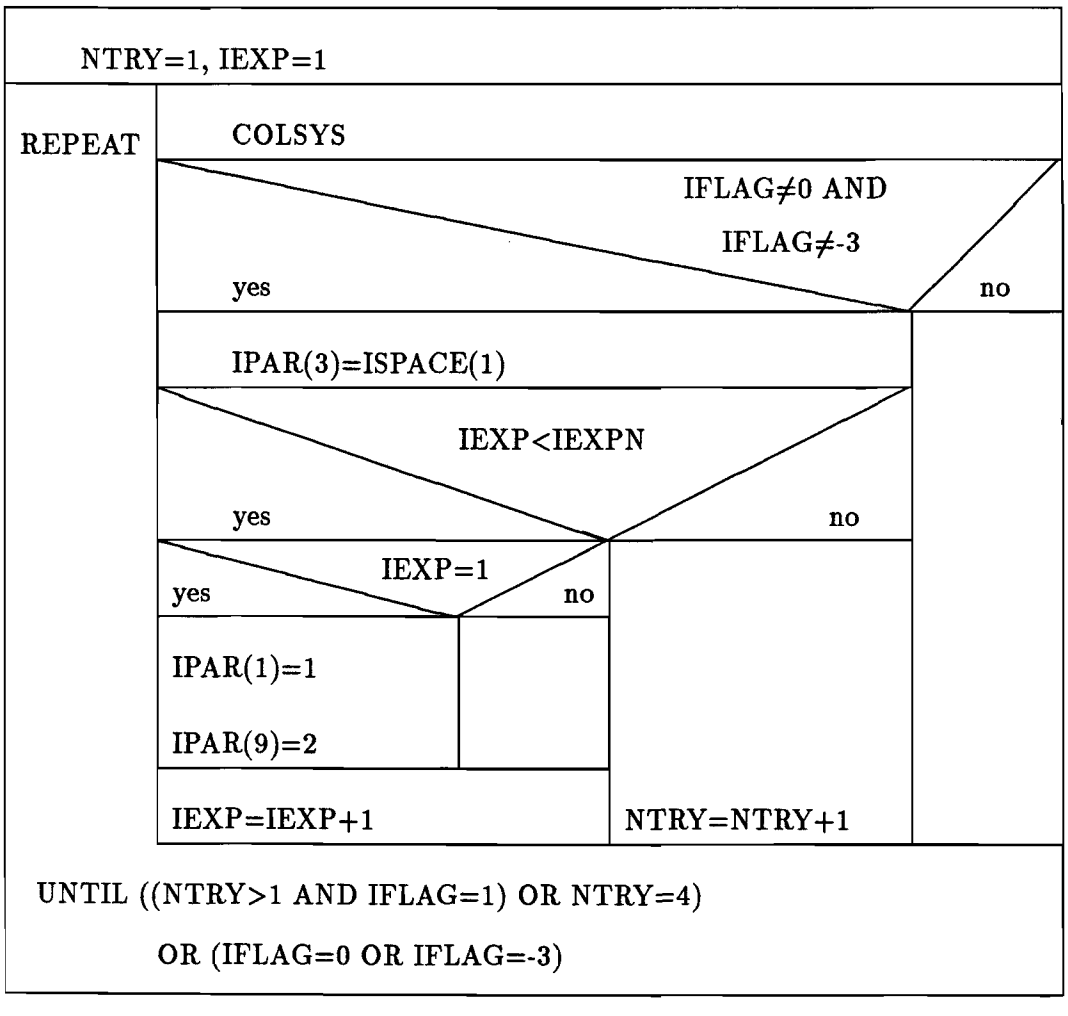
In the present form the program needs the following input data from the input file INPFILE.DAT:

EPS		the maximum tolerance ε for the calculated approximations $z_1 = \zeta_1$, $z_2 = \zeta_5$, $M_1 = \zeta_3$ and $M_2 = \zeta_7$.
IPRINT		Is used for output control.
L	mm	The length of the both rollers.
M10	N.mm	The moment $M_1(0)$ in x=0 for roller 1.

M1L	N.mm	The moment $M_1(L)$ in x=L for roller 1.
M20	N.mm	The moment $M_2(0)$ in x=0 for roller 2.
M2L	N.mm	The moment $M_2(L)$ in x=L for roller 2.
F20	N	The force $F_2(0)$ in x=0 for roller 2.
F2L	N	The force $F_2(L)$ in x=L for roller 2.
R1	mm	The radius R_1 of roller 1.
R2	mm	The radius R_2 of roller 2.
W1	N/mm	Weight/length W_1 for roller 1.
W2	N/mm	Weight/length W_2 for roller 2.
I1	mm^4	Moment of inertia I_1 for roller 1.
I2	mm^4	Moment of inertia I_2 for roller 2.
D1	mm	Thickness D_1 of the elastic layer on roller 1.
D2	mm	Thickness D_2 of the elastic layer on roller 2.
E		Young's modulus E
PCOEF		The coefficient $A(x)$ for the function P
	(in this program: the function $A(x)$ is constant)	
PEXP		The exponent $\alpha(x)$ for the function P
	(in this program: the function $\alpha(x)$ is constant)	
OUTFILE		The name of the outputfile

6.3 the iterationproces

We consider the case that $P = A\delta^{\alpha}$ with α is constant: α =PEXP. If P is linear, say $\alpha = 1$, then the solution of (1) is easily found. To find the solution for an arbitrary α we can therefore start at $\alpha = 1$, and then iterate in IEXPN steps via
α = `PEXP1 = 1 + PEXP * (IEXP - 1) / (IEXPN - 1).`
This iteration is depicted in the following program structure diagram.



7 user subroutines

7.1 the subroutine DD

1 Effect:

The subroutine determines D as maximum of $R_1 + R_2 + D_1 + D_2$ (here R_1, R_2, D_1 and D_2 are constant functions).
If R_1, R_2, D_1 and D_2 are explicit functions of x, it is strongly recommended to determine this maximum analytically and to read this maximum as input (a constant value).

2 Heading:

```
SUBROUTINE DD(D)
DOUBLE PRECISION D
```

3 Parameters:

D — On output: the distance between the axes of the two rollers when they just touch each other in rest position.

4 Remarks:

The subroutine uses the external functions R1, R2, D1 and D2.

7.2 the function P

1 Effect:

P is the reaction $P(x, \delta)$ of the rollers, given as a (positive) empirical function of x, $A(x)$, $\alpha(x)$ and $z_1 - z_2$ (as described in chapter 2). If the rollers don't touch each other ($\delta \leq 0$) then $P = 0$. In this program $A(x)$=PCOEF and $\alpha(x)$=PEXP.

2 Heading:

```
FUNCTION P(X, Z)
DOUBLE PRECISION P, X, Z(*)
```

3 Parameters:

X — On input: the x-coordinate with $0 \leq$ X $\leq$ L

Z — On input: the vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$

4 Remarks:

The subroutine uses the external function DELTA and the variables PCOEF and PEXP1 of the commonblock /PPARAM/, declared in the mainprogram.

7.3 the function DELTA

1 Effect:

DELTA is the interpenetration function δ of both rollers with
$\delta = z_1 - z_2 + D - (R_1 + R_2 + D_1 + D_2)$

2 Heading:

```
FUNCTION DELTA(X, Z)
DOUBLE PRECISION DELTA, X, Z(*)
```

3 Parameters:

X On input: the x-coordinate with $0 \leq$ X $\leq$ L

Z On input: the vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$

4 Remarks:

The subroutine uses the external functions R1, R2, D1 and D2 and the variable D of the commonblock /D/, declared in the mainprogram.

7.4 the function R1

1 Effect:

R1 is the radius of the roller 1; place dependent (in this program constant)

2 Heading:

```
FUNCTION R1(X)
DOUBLE PRECISION R1, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq$ X $\leq$ L

4 Remarks:

The subroutine uses the variable R1INP of the commonblock /GEOM/, declared in the mainprogram.

7.5 the function R2

1 **Effect:**

R2 is the radius of the roller 2; place dependent (in this program constant)

2 **Heading:**

```
FUNCTION R2(X)
DOUBLE PRECISION R2, X
```

3 **Parameters:**

X On input: the x-coordinate with $0 \leq X \leq L$

4 **Remarks:**

The subroutine uses the variable R2INP of the commonblock /GEOM/, declared in the mainprogram.

7.6 the function D1

1 **Effect:**

D1 is the thickness of the elastic layer on roller 1; place dependent (in this program constant)

2 **Heading:**

```
FUNCTION D1(X)
DOUBLE PRECISION D1, X
```

3 **Parameters:**

X On input: the x-coordinate with $0 \leq X \leq L$

4 **Remarks:**

The subroutine uses the variable D1INP of the commonblock /GEOM/, declared in the mainprogram.

7.7 the function D2

1 **Effect:**

D2 is the thickness of the elastic layer on roller 2; place dependent (in this program constant)

2 **Heading:**

```
FUNCTION D2(X)
DOUBLE PRECISION D2, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

4 Remarks:

The subroutine uses the variable D2INP of the commonblock /GEOM/, declared in the mainprogram.

7.8 the function W1

1 Effect:

W1 is the weight density of roller 1; place dependent (in this program constant)

2 Heading:

```
FUNCTION W1(X)
DOUBLE PRECISION W1, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

4 Remarks:

The subroutine uses the variable W1INP of the commonblock /GEOM/, declared in the mainprogram.

7.9 the function W2

1 Effect:

W2 is the weight density of roller 2; place dependent (in this program constant)

2 Heading:

```
FUNCTION W2(X)
DOUBLE PRECISION W2, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

4 Remarks:

The subroutine uses the variable W2INP of the commonblock /GEOM/, declared in the mainprogram.

7.10 the function I1

1 Effect:

I1 is the moment of inertia of roller 1; place dependent (in this program constant)

2 Heading:

```
FUNCTION I1(X)
DOUBLE PRECISION I1, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

4 Remarks:

The subroutine uses the variable I1INP of the commonblock /GEOM/, declared in the mainprogram.

7.11 the function I2

1 Effect:

I2 is the moment of inertia of roller 2; place dependent (in this program constant)

2 Heading:

```
FUNCTION I2(X)
DOUBLE PRECISION I2, X
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

4 Remarks:

The subroutine uses the variable I2INP of the commonblock /GEOM/, declared in the mainprogram.

8 routines voor COLSYS

8.1 the subroutine FSUB

1 Effect:

the subroutine evaluates the vector field $\underline{\Phi}(x, \underline{\zeta})$ at a given point x and given vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$.
u_1 and u_2 are defined by
$u_1(x) = EI_1(x)z_1''$ and $u_2(x) = EI_2(x)z_2''$

2 Heading:

```
SUBROUTINE FSUB(X, Z, F)
DOUBLE PRECISION X, Z(*), F(*)
```

3 Parameters:

X	On input: the x-coordinate with $0 \leq$ X $\leq$ L
Z	On input: the vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$
F	On output: the vector $\underline{\Phi}$

4 Remarks:

The subroutine uses the external functions W1, W2, P, I1 and I2 en the variable E of the commonblock /E/, declared in the mainprogram.

8.2 the subroutine DFSUB

1 Effect:

The subroutine calculates by numerical differentiation the Jacobian matrix (the matrix of partial derivatives) of $\underline{\Phi}(x, \underline{\zeta})$.
So: $DF_{ij} = \frac{\partial \Phi_i}{\partial \zeta_j}$, $i = 1, ..., 4$, $j = 1, ..., 8$.

2 Heading:

```
SUBROUTINE DFSUB(X, Z, DF)
DOUBLE PRECISION X, Z(*), DF(4,*)
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

Z On input: the vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$

DF On output: the matrix with elements $\frac{\partial \Phi_i}{\partial z_j}$, $i = 1,...,4$, $j = 1,...,8$.

4 Remarks:

The subroutine uses the variable PRECIS of the commonblock /PREC/, declared in the mainprogram.
PRECIS = $\sqrt{\eta}$ with η the machine precision, calculated in the mainprogram.

8.3 the subroutine GSUB

1 Effect:

the subroutine evaluates the i-th component of $\Gamma(x, \underline{\zeta})$ at the point $x = \chi_i$. This subroutine is used by COLSYS to satisfy the boundary condition with index i by searching the zero of the function $\Gamma(\chi_i, \underline{\zeta})$.

2 Heading:

```
SUBROUTINE GSUB(I, Z, G)
INTEGER I
DOUBLE PRECISION Z(*), G
```

3 Parameters:

I On input: the index of the i-de boundary condition with $1 \leq i \leq 4$

Z On input: the vector $\underline{\zeta} = (z_1,\ z_1',\ u_1,\ u_1',\ z_2,\ z_2',\ u_2,\ u_2')^T$.

G On output: $\Gamma(\chi_i, \underline{\zeta})$

4 Remarks:

The subroutine uses the variables F20, F2L, M10, M1L, M20 en M2L of the commonblock /BOUNDS/, declared in the mainprogram.

8.4 the subroutine DGSUB

1 Effect:

the subroutine evaluates the vector DG, the i-th row of the Jacobian matrix (the matrix of partial derivatives) of Γ with:

$DG_j = \frac{\partial \Gamma_i}{\partial \zeta_j}$ $j = 1,.....,8.$

2 Heading:

```
SUBROUTINE DGSUB(I, Z, DG)
INTEGER I
DOUBLE PRECISION Z(*), DG(*)
```

3 Parameters:

I On input: the index of the i-th boundary condition with $1 \leq i \leq 4$

Z On input: the vector $\underline{\zeta}$ (in the point x=χ_i)

DG On output: the vector with components $\frac{\partial \Gamma(\chi_i)}{\partial \zeta_j}$ $j = 1, ..., 8.$

4 Remarks:

none.

8.5 the subroutine GUESS

1 Effect:

The subroutine calculates a first estimate in the point x for:

1. $(z_1, z_1', M_1, -F_1, z_2, z_2', M_2, -F_2)$
   by linear interpolation of the given boundary values.
2. The four second derivatives as calculated by the subroutine FSUB, but with a perturbation added, since otherwise the COLSYS routine starts with effectively singular input.

2 Heading:

```
SUBROUTINE GUESS(X, Z, DMVAL)
DOUBLE PRECISION X, Z(*), DMVAL(*)
```

3 Parameters:

X On input: the x-coordinate with $0 \leq X \leq L$

Z On output: an estimate of the vector $\underline{\zeta}$

DMVAL On output: an estimate for the vector $(z_1'', z_2'', u_1'', u_2'')^T$

4 Remarks:

The subroutine uses the external functions W1, W2 and the variables F20, F2L, M10, M1L, M20, M2L and L of the commonblocks /BOUNDS/ and /L/, declared in the mainprogram.

9 an example

9.1 program data

```
 1.00000000000000E-0004
1
 3.15000000000000E+0002
-1.00000000000000E+0005
-1.00000000000000E+0005
 0.00000000000000E+0000
 0.00000000000000E+0000
 1.50000000000000E+0003
-1.50000000000000E+0003
 5.00000000000000E+0001
 5.00000000000000E+0001
 0.00000000000000E+0000
 0.00000000000000E+0000
 8.59200000000000E+0003
 8.59200000000000E+0007
 1.00000000000000E+0000
 1.00000000000000E+0000
 2.06000000000000E+0005
 5.60000000000000E+0001
 1.84000000000000E+0000
uit.dat
```

9.2 program results

uit.dat

```
 M10 = -.1000E+06 N.mm
 M1L = -.1000E+06 N.mm
 M20 = 0.0000E+00 N.mm
 M2L = 0.0000E+00 N.mm
 F20 = 0.1500E+04 N
 F2L = -.1500E+04 N
 lengte  van de walsen      =   315.00 mm
 straal  van wals 1         =    50.00 mm
 straal  van wals 2         =    50.00 mm
 dikte rubberlaag op wals 1 =     1.00 mm
 dikte rubberlaag op wals 2 =     1.00 mm
 gewicht/lengte (wals 1)    = 0.0000E+00 N/mm
 gewicht/lengte (wals 2)    = 0.0000E+00 N/mm
 traagheidsmoment I1        = 0.8592E+04 mm^4
 traagheidsmoment I2        = 0.8592E+08 mm^4
 Young's modulus E          = 0.2060E+06 N/mm^2
 D                          =   102.00 mm
 uitdrukking voor  P(z1-z2) = 56.0000*DELTA**1.8400

 Tolerantie voor  z1(x)                      =0.1000E-03
 Tolerantie voor  M1(x) =  E.I1(x).z1''(x)   =0.1000E-03
 Tolerantie voor  z2(x)                      =0.1000E-03
 Tolerantie voor  M2(x) =  E.I2(x).z2''(x)   =0.1000E-03

 Z(1)=  z1(x)
 Z(2)=  z1'(x)
 Z(3)=  M1(x) =  E.I1(x).z1''(x)
 Z(4)= -F1(x) = (E.I1(x).z1''(x))'
 Z(5)=  z2(x)
 Z(6)=  z2'(x)
 Z(7)=  M2(x) =  E.I2(x).z2''(x)
 Z(8)= -F2(x) = (E.I2(x).z2''(x))'
```

P(X) = 56.0000*DELTA(X,Z)**1.0000

X	Z1-Z2	Z1	Z2	M1	M2	F1	F2
0.	0.142E+00	0.000E+00	-0.142E+00	-0.10E+06	0.00E+00	-0.15E+04	0.15E+04
32.	0.179E+00	0.365E-01	-0.142E+00	-0.57E+05	-0.43E+05	-0.12E+04	0.12E+04
63.	0.183E+00	0.405E-01	-0.142E+00	-0.24E+05	-0.76E+05	-0.89E+03	0.89E+03
95.	0.173E+00	0.306E-01	-0.142E+00	-0.11E+04	-0.99E+05	-0.57E+03	0.57E+03
126.	0.162E+00	0.196E-01	-0.142E+00	0.12E+05	-0.11E+06	-0.28E+03	0.28E+03
158.	0.157E+00	0.151E-01	-0.142E+00	0.17E+05	-0.12E+06	-0.73E-02	0.73E-02
189.	0.162E+00	0.196E-01	-0.142E+00	0.12E+05	-0.11E+06	0.28E+03	-0.28E+03
221.	0.173E+00	0.306E-01	-0.142E+00	-0.11E+04	-0.99E+05	0.57E+03	-0.57E+03
252.	0.183E+00	0.405E-01	-0.142E+00	-0.24E+05	-0.76E+05	0.89E+03	-0.89E+03
284.	0.179E+00	0.365E-01	-0.142E+00	-0.57E+05	-0.43E+05	0.12E+04	-0.12E+04
315.	0.142E+00	0.171E-15	-0.142E+00	-0.10E+06	0.43E-11	0.15E+04	-0.15E+04

P(X) = 56.0000*DELTA(X,Z)**1.8400

X	Z1-Z2	Z1	Z2	M1	M2	F1	F2
0.	0.354E+00	0.000E+00	-0.354E+00	-0.10E+06	0.00E+00	-0.15E+04	0.15E+04
32.	0.390E+00	0.365E-01	-0.354E+00	-0.57E+05	-0.43E+05	-0.12E+04	0.12E+04
63.	0.394E+00	0.404E-01	-0.354E+00	-0.24E+05	-0.76E+05	-0.89E+03	0.89E+03
95.	0.384E+00	0.303E-01	-0.354E+00	-0.11E+04	-0.99E+05	-0.58E+03	0.58E+03
126.	0.373E+00	0.192E-01	-0.354E+00	0.12E+05	-0.11E+06	-0.28E+03	0.28E+03
158.	0.369E+00	0.147E-01	-0.354E+00	0.17E+05	-0.12E+06	-0.24E-04	0.24E-04
189.	0.373E+00	0.192E-01	-0.354E+00	0.12E+05	-0.11E+06	0.28E+03	-0.28E+03
221.	0.384E+00	0.303E-01	-0.354E+00	-0.11E+04	-0.99E+05	0.58E+03	-0.58E+03
252.	0.394E+00	0.404E-01	-0.354E+00	-0.24E+05	-0.76E+05	0.89E+03	-0.89E+03
284.	0.390E+00	0.365E-01	-0.354E+00	-0.57E+05	-0.43E+05	0.12E+04	-0.12E+04
315.	0.354E+00	-0.206E-15	-0.354E+00	-0.10E+06	-0.78E-11	0.15E+04	-0.15E+04

References

1. B.Singh, B.Paul,
   Elastic deformation of end-supported pressure rollers,
   Journal of Vibration, Acoustics, Stress and Reliability in Design (Transactions of the ASME), 1984, Vol.106, p.517-522

2. U.Ascher, J.Christiansen and R.D.Russell,
   Collocation software for boundary-value ODEs,
   ACM Trans. Math Software 7 (1981), 209-222.

3. G.Bader and U.Ascher,
   A new basis implementation for a mixed order boundary value ODE solver,
   SIAM J. Scient. Stat. Comput.(1987)

4. U.Ascher, J.Christiansen and R.D.Russell,
   A collocation solver for mixed order systems of boundary value problems,
   Math. Comp. 33 (1979), 659-679.

5. U.Ascher, J.Christiansen and R.D.Russell,
   COLSYS - a collocation code for boundary value problems,
   Lecture Notes Comp. Sc. 76, Springer Verlag,
   B.Childs et. al. (eds.) (1979), 164-195.

6. U.Ascher, R.M.M.Mattheij, R.D.Russell,
   Numerical Solution of Boundary Value Problems for Ordinary Differential Equations,
   Prentice Hall, New Jersey, 1988.

7. L.D.Landau, E.M.Lifshitz,
   Theory of Elasticity,
   Pergamon Press, Oxford, 1970.

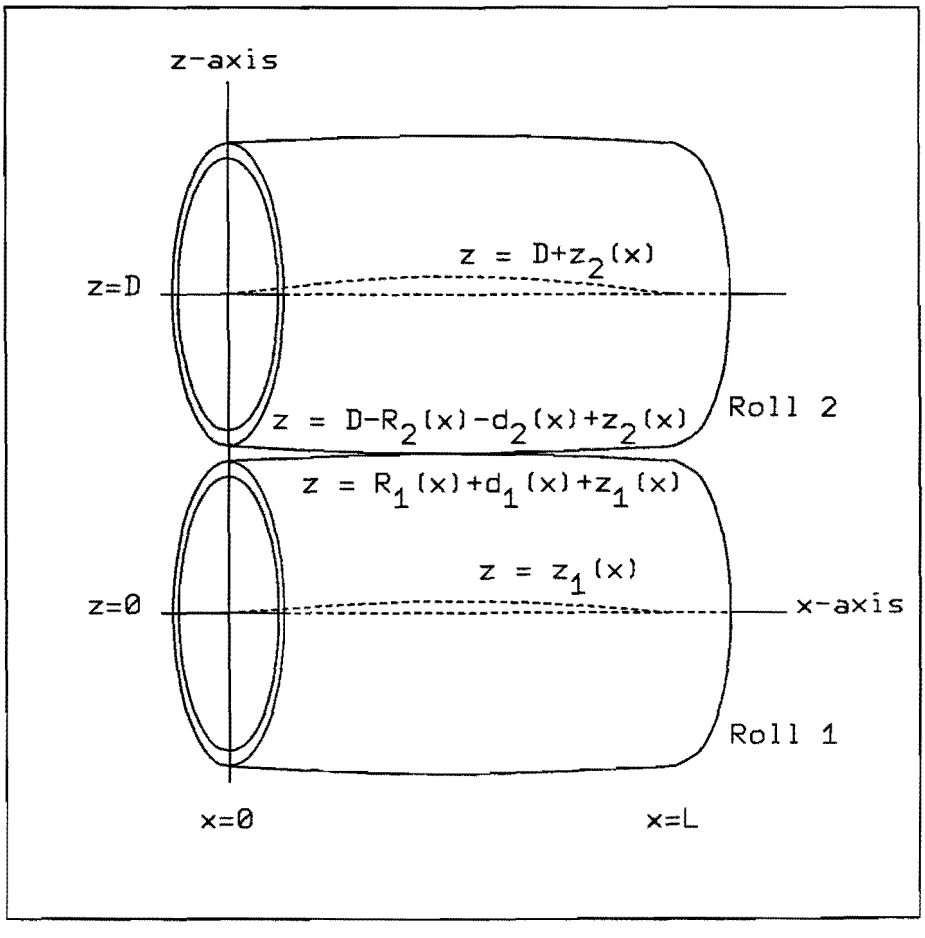


Figure 1. Sketch of geometry