Memorandum M-536

An efficient approximate solution method for predicting the buckling load of axially compressed imperfect isotropic cylindrical shells

L.Ph. J. Frenken

Delft - The Netherlands August 1985
Acknowledgement

The author wishes to take this opportunity to sincerely thank prof.dr. J. Arbocz for the patience and guidance he generously extended during the course of this investigation.

The author also thanks Jan Hol for his advice and comments.
ABSTRACT

A theoretical investigation of an efficient numerical solution scheme to solve approximately the nonlinear Donnell equations for imperfect isotropic cylindrical shells with edge restraints and under axial compression was carried out.

In order to solve approximately the Donnell equations, the equivalent set of nonlinear ordinary differential equations derived by Arbocz and Sechler, and in addition the concept of "end shortening" introduced by the same authors, were used. The resulting nonlinear 2-point boundary value problem was solved using Newton's discretized method of quasi-linearization in order to obtain a linearized system of central difference equations which was written in the form of a bordered block tridiagonal matrix equation. Arbocz' modification of Potters' method was used to solve efficiently this matrix equation. By successive iterations a solution to the set of nonlinear ordinary differential equations was obtained. Using increments in "end shortening" it was possible to get around the limit point of the nonlinear prebuckling equilibrium states without encountering the usual convergence difficulties at axial load levels close to the limit point.

The use of this method makes it possible to investigate how the axial load level at the limit point is affected by the following factors: the choice of in-plane boundary conditions, the prebuckling growth caused by radial edge constraint, the location of the load application point, the orientation and shape of the axisymmetric and asymmetric imperfection components, and the finite length of the shell.
TABLE OF CONTENTS

Part
1. Introduction

2. Theoretical analysis
   2.1 Formulation of the Donnell equations
   2.2 Reduction to an equivalent set of ordinary differential equations
   2.3 Boundary conditions
   2.4 The concept of "end shortening"

3. Numerical analysis
   3.1 Linearization of the equations
   3.2 Discretization of the linearized equations
   3.3 Enforcement of the boundary conditions
   3.4 Completion of the solution
   3.5 Initial values
   3.6 Further remarks

4. Potters' method

5. Arbocz' modification of Potter's method

6. Numerical results

7. Conclusions

References

Appendix A

Appendix B

Appendix C

Tables

Figures
NOMENCLATURE

A

matrix, Eq. (67)
A₀, A₁
axial dependence of radial imperfection, positive outward, Eq (8)
c
Poisson's effect \( = \sqrt{3(1-v^2)} \)
dₜ
matrix coefficient, Eq. (67)
Dᵢ
matrix, Eq. (67)
E
Young's modulus
F, f₀, f₁, f₂
Airy stress functions, Eq. (10)
h
dimensionless spacing of grid points
L
shell length
Lᵢ
matrix, Eq. (67)
n
number of full waves in the circumferential direction
N
number of grid points minus 1
Nₓ, Nᵧ, Nₓᵧ
stress resultants
o
matrix coefficient defined on pages 44 and 45
q
load eccentricity measured from the skin midsurface, positive outward
q
nondimensional load eccentricity \( = \frac{4cR}{t^2} q \)
Fᵢ, Fᵢ₁
vectors, Eq. (67)
R
radius of shell
t
thickness of shell
u, v
in-plane displacement components
Uᵢ
matrix, Eq. (67)
w, w₀, w₁
radial displacement, positive outward, Eq. (9)
\( \dot{w} \)
radial imperfection from perfect circular cylinder, Eq. (8)
x, y
axial and circumferential coordinates on middle surface of shell, respectively
\( \bar{x}, \bar{y} \)
nondimensional coordinates \( \bar{x} = \frac{x}{R}, \bar{y} = \frac{y}{R} \)
\( \bar{x}_t, \bar{y}_t \)
vectors, Eq. (67)
xₜ, yₜ
vector elements, Eq. (67)
Y
unified vector variable, Eq. (30)
Δ
end shortening
Δ₝
end shortening component \( = \delta - \lambda \)
εₓ
strain component
λ
nondimensional load parameter \( = \frac{cσ R}{Et} \)
v
Poisson's ratio
σₓ, σᵧ, σₓᵧ
normal and shear stresses respectively
Subscripts

1 reference to grid point

Difference notation

\[ \frac{d}{dx} \]

Correction term notation

\[ \delta \]

correction term

Vector and matrix notation

capital letter = matrix
underlined letter = vector
1. INTRODUCTION

The stability of circular cylindrical shells under axial compression has been studied extensively both theoretically and experimentally by many investigators. Buckling loads, predicted by linearized small deflection theories, proved much higher than those realized in experiments, and the experimental results showed a perplexingly large scatter band. Choice of in-plane boundary condition and prebuckling deformation caused by edge restraint and the location of the load application point have been showed to affect the buckling load (Refs. 1, 2, 3 and 4). However, initial geometric imperfections have been accepted as the main cause for poor correlation and the wide experimental scatter between the predictions of the linearized small deflection theory and the experimental results.

In 1974 Arbocz and Sechler presented a solution for the case of axially compressed isotropic shells with both axisymmetric and asymmetric imperfection, where a rigorous satisfaction of the boundary conditions was included (Ref. 5). The method of numerical solution of the equations they used here is known as "parallel shooting".

This method is slower than a coarse standard finite difference or finite element scheme; however, if the length of the intervals of integration is chosen properly so that numerical instabilities are avoided, then, as they claim, this method gives more accurate results (Ref. 6). As an alternative Bremmer developed in 1975 a less time-consuming numerical solution of the equations by using Newton's discretized method of quasi linearization.

In 1976 Arbocz and Sechler presented an extension of their method to include ring- and stringer-stiffened shells, where, in addition it was shown that by introducing the concept of "end-shortening", it is possible to integrate around the limit point of the nonlinear prebuckling equilibrium states without encountering the usual convergence difficulties at axial load levels close to the limit point (Ref. 6). It was the purpose of this study to include the concept of "end-shortening" in the method developed by Bremmer (Ref. 13), in order to get also around the limit point.
2. THEORETICAL ANALYSIS

2.1. Formulation of the Donnell equations

Assuming that the radial displacement $W$ is positive outward and that the two-di-
mensional membrane stress resultants can be obtained from a Airy stress function
$F$ as $F_{yy} = N_x = t\sigma_x$, $F_{xx} = N_y = t\sigma_y$ and $-F_{xy} = N_y = t\sigma_{xy}$, then the Donnell equa-
tions for an imperfect cylindrical shell can be written as (Ref. 12).

Displacement Compatibility:

$$\frac{1}{Et} \nabla^4 F - \frac{1}{R} W_{xx} + \frac{1}{2} L (w, w+2 \bar{w}) = 0 \quad (1)$$

Equilibrium:

$$\frac{Et^3}{12(1-v^2)} \nabla^4 w + \frac{1}{R} F_{xx} - L (F, w+\bar{w}) = 0 \quad (2)$$

where the nonlinear operator $L$ is defined by

$$L(S, T) = S_{xx}T_{yy} - 2S_{xy}T_{xy} + S_{yy}T_{xx} \quad (3)$$

and $\nabla^4$ is the two-dimensional biharmonic operator. Commas in the subscripts de-
note repeated partial differentiation with respect to the independent variables
following the comma.

2.2 Reduction to an equivalent set of ordinary differential equations

In order to obtain an approximate solution of the nonlinear Donnell-type equa-
tions which will include rigorous satisfaction of the experimental boundary
conditions, the form used by Arbocz and Sechler (Ref. 10) is assumed to
represent the initial imperfection surface, namely

$$\bar{w}(x, \bar{y}) = tA_0(\bar{x}) + tA_1(\bar{x}) \cos n\bar{y} \quad (8)$$

where $A_0(\bar{x})$ and $A_1(\bar{x})$ are known functions of $\bar{x}$. Now equations (1) and (2) admit
seperable solutions of the form
\[ w(\overline{x}, \overline{y}) = \frac{\xi v \lambda}{c} + t w_0(\overline{x}) + tw_1(\overline{x}) \cos n \overline{y} \]  
(9)

\[ F(\overline{x}, \overline{y}) = \frac{Er \xi^2}{c} \left[ - \frac{\lambda}{2} \overline{y}^2 + f_0(\overline{x}) + f_1(\overline{x}) \cos n \overline{y} + f_2(\overline{x}) \cos 2n \overline{y} \right] \]  
(10)

Assuming the axial dependence of the response to be an unknown function of \( \overline{x} \) will reduce the buckling problem to the solution of a set of nonlinear ordinary differential equations, which allows the satisfaction of the experimental boundary conditions. The value of the Poisson's expansion \( \frac{\xi v \lambda}{c} \) is obtained by enforcing the circumferential periodicity condition.

Substituting the expressions assumed for \( \overline{w}, w \) and \( F \) into the compatibility equation (1), using some trigonometric identities, and equating coefficients of like terms, (see Appendix A), results in the following system of 3 nonlinear ordinary differential equations

\[ f_0^{IV} - cw_0'' - \frac{c}{2 R} n^2 \left[ A_1 w_1'' + A_1' w_1 + (2A_1'' + w_1') w_1 + w_1 w_1'' \right] = 0 \]  
(11)

\[ f_1^{IV} - 2n^2 f_1'' + n^4 f_1' - cw_1'' - \frac{c}{R} n^2 \left[ A_0'' w_1 + (A_1 + w_1) w_1'' \right] = 0 \]  
(12)

\[ f_2^{IV} - 8n^2 f_2'' + 16n^4 f_2 - \frac{c}{2 R} n^2 \left[ A_1 w_1'' + A_1' w_1 - (2A_1'' + w_1') w_1 + w_1 w_1'' \right] = 0 \]  
(13)

Substituting in turn the expressions assumed for \( \overline{w}, w \) and \( F \) into the equilibrium equation (2) and apply Galerkin's procedure (see Appendix A) gives the following two nonlinear ordinary differential equation

\[ w_0^{IV} + 4c \left( \frac{R}{\xi} \right)^2 f_1'' + 4c \frac{R}{\xi} \lambda \left( A_0'' + w_0'' \right) + 2c \frac{R}{\xi} n^2 \left[ (A_1'' + w_1'') f_1 + (A_1 + w_1) f_1' \right] = 0 \]  
(14)

\[ w_1^{IV} - 2n^2 w_1'' + n^4 w_1' + 4c \left( \frac{R}{\xi} \right)^2 f_1'' + 4c \frac{R}{\xi} \lambda \left( A_1'' + w_1' \right) 
+ 2c \frac{R}{n^2} \left[ 2(A_0'' + w_0'') f_1 + 4 (A_1'' + w_1') f_2 + (A_1 + w_1) f_2'' \right] 
+ 4 (A_1' + w_1') f_2' + 2 (A_1 + w_1) f_2'' \right] = 0 \]  
(15)

where \( ' = \frac{d}{dx} \)
As pointed out by Narasimhan and Hoff (Ref. 7) equation (11) can be integrated twice to yield

\[ f_0'' = c \omega_0 + \frac{c}{4} \frac{t}{R} n^2 (2A_1 + \omega_1) \omega_1 \]  

(16)

where the constants of integration are set equal to zero in order to satisfy periodicity in the circumferential direction (Ref. 3). Substituting equation (16) into equations (12-15) gives the following system of four nonlinear ordinary differential equations.

\[ f_1'' = 2n^2 f_1'' + n^4 f_1 - c w_1'' - \frac{c t}{R} n^2 [A_0'' \omega_1 + (A_1 + \omega_1) w_0'''] = 0 \]  

(17)

\[ f_2'' = 8n^2 f_2'' + 16n^4 f_2 - \frac{c t}{R} n^2 [A_1 w_1'' + A_1'' \omega_1] \]  

\[ - (2A_1 + \omega_1') \omega_1'' + \omega_1 w_1'' = 0 \]  

(18)

\[ w_0'' = 4c^2 \left( \frac{R}{t} \right)^2 w_0 + c^2 \frac{R}{t} n^2 (2A_1 + \omega_1) \omega_1 + 4c \frac{R}{t} \lambda (A_0'' + \omega_0''') \]  

\[ + 2c \frac{R}{t} n^2 [(A_1'' + \omega_1''') f_1' + (A_1 + \omega_1) f_1'' + 2 (A_1 + \omega_1) f_1'] = 0 \]  

(19)

\[ w_1'' = 2n^2 w_1'' + n^4 [1 + c^2 (A_1 + \omega_1)(2A_1 + \omega_1)] \omega_1 \]  

\[ + 4c^2 \frac{R}{t} n^2 (A_1 + \omega_1) w_0 + 4c \left( \frac{R}{t} \right)^2 f_1'' + 4c \frac{R}{t} \lambda (A_1'' + \omega_1''') \]  

\[ + 2c \frac{R}{t} n^2 [2(A_0'' + \omega_0''') f_1 + 4 (A_1'' + \omega_1''') f_2 + (A_1 + \omega_1) f_2'' \]  

\[ + 4(A_1 + \omega_1) f_2'] = 0 \]  

(20)

2.3 Boundary conditions (Ref. 8)

\[ SS-1: \ w = N_{xy} = 0, M_x = N_0 q, N_x = -N_0 \text{ at } \bar{x} = 0, \ L/R \]

which reduces to
\( w_0 = - \frac{v}{c} \lambda \)

\( w_0'' = - q \lambda \) \hspace{1cm} (21)

\( w_1 = f_1' = f_2' = w_1' = f_1 = f_2 = 0 \)

**SS-2:** \( w = N_{xy}, u = 0, M_x = N_0q \text{ at } \bar{x} = 0, L/R \)  

which reduces to  

\( w_0' = - \frac{v}{c} \lambda \)

\( w_0'' = - q \lambda \) \hspace{1cm} (22)

\( f_1''' = cw_1' + c e \frac{E}{R} n^2 n_x \lambda w_0' \)

\( f_2''' = c \frac{e}{2} \left( \frac{n_x}{R} \right)^2 \lambda w_1' \)

\( w_1 = f_1' = f_2' = w_1' = 0 \)

**SS-3:** \( w = v = 0, M_x = N_0q, N_x = -N_0 \text{ at } \bar{x} = 0, L/R \)  

which reduces to  

\( w_0' = - \frac{v}{c} \lambda \)

\( w_0'' = - q \lambda \) \hspace{1cm} (23)

\( w_1 = f_1' = f_2' = w_1' = f_1 = f_2 = 0 \)

**SS-4:** \( w = v = u = 0, M_x = N_0q \text{ at } \bar{x} = 0, L/R \)  

which reduces to  

\( w_0' = - \frac{v}{c} \lambda \)

\( f_1''' = -v n^2 f_1 \) \hspace{1cm} (24)

\( f_2''' = -4v n^2 f_2 \)

\( w_0'' = - q \lambda \)
\[ f_1''' = (2+v)n^2 f_1' + cw_1' + \frac{c}{R} n^2 A_1 w_0' \]

\[ f_2''' = (2+v) 4n^2 f_2' + \frac{c^2}{R} n^2 A_1 w_1' \]

\[ w_1 = w_1' = 0 \]

\( C-1: \) \( w = w_x = N_y = 0, N_x = -N_0 \) at \( \bar{x} = 0, L/R \)
which reduces to
\[ w_0 = -\frac{v}{c} \lambda \]
\[ w_1 = w_1' = f_1' = f_2' = f_1'' = f_2''' = 0 \]  \hspace{1cm} (25)

\( C-2: \) \( w = w_x = N_y = u = 0 \) at \( \bar{x} = 0, L/R \)
which reduces to
\[ w_0 = -\frac{v}{c} \lambda \]
\[ w_1 = w_1' = f_1' = f_2' = f_1'' = f_2''' = 0 \]  \hspace{1cm} (26)

\( C-3: \) \( w = w_x = v = 0, N_x = -N_0 \) at \( \bar{x} = 0, L/R \)
which reduces to
\[ w_0 = -\frac{v}{c} \lambda \]
\[ w_1 = w_1' = f_1' = f_2' = f_1'' = f_2''' = 0 \]  \hspace{1cm} (27)

\( C-4: \) \( w = w_x = v = u = 0 \) at \( \bar{x} = 0, L/R \)
which reduces to
\[ w_0 = -\frac{v}{c} \lambda \]
\[ f_1'' = -v n^2 f_1' \]
\[ f_2'' = -v n^2 f_2' \]
\[ f_1''' = (2+v) n^2 f_1' \]  \hspace{1cm} (28)
\[ f'''' = (2+\nu) \frac{4n}{\pi^2} f'' \]

\[ w_1 = w'_0 = w'_1 = 0 \]

**Symm.:**  \( H = w'_x = u = N_{xy} = 0 \) at \( \bar{x} = 0, L/R \)

which reduces to

\[ w'_0 = w'_1 = f'_1 = f'_2 = f'' = f'''' = w'''' = w''' = 0 \] at \( \bar{x} = 0, L/R \) \hspace{1cm} (29)

### 2.4 The concept of "end shortening"

Introducing as an unified variable, the 16-dimensional vector \( \underline{Y} \) defined as follows

\[ Y_1 = w_0, \; Y_2 = w_1, \; Y_3 = f_1, \; Y_4 = f_2, \; Y_5 = w'_0, \; \ldots, \; Y_{16} = f'''' \]

then the system of equations (17-20) can be reduced to a nonlinear 2-point boundary value problem

\[ \frac{d}{dx} \underline{Y} = f(x, \underline{Y}; \lambda) \quad \text{for} \; 0 < \bar{x} < L/R \]

chosen boundary conditions expressed in terms of vector \( \underline{Y} \)

\[ \text{at} \; \bar{x} = 0, L/R \] \hspace{1cm} (30)

The solution of this nonlinear 2-point boundary value problem will then locate the limit point of the prebuckling states. By definition the value of the loading parameter \( \lambda \) corresponding to the limit point can be obtained by plotting the maximum amplitude of the asymmetric component of the prebuckling deformation versus axial load level (Fig. 1). In order to solve the nonlinear 2-point boundary value problem (30), Newton's method of quasilinearization is used. Using load increments, then, dependent on initial guesses of the solution this iterative method fails to converge near the limit point. A closer look at the solution curve presented in Fig. 1 reveals, however, that one should be able to extend the response curve beyond the limit point by using increments in deformation instead of increments in loading.
Following Ref. 9, let us define "unit end shortening" as

$$\epsilon = \frac{1}{2\pi RL} \int_{0}^{L} \int_{0}^{\pi} (u_{xx} - qw_{x}) \, dx \, dy$$

(31)

where

$$u_{xx} = \epsilon \bar{w}_{xx} + \frac{1}{2} w_{x} (w_{x} + 2\bar{w}_{x})$$

(32)

$$\epsilon = \frac{1}{6t} \left( F_{yy} - v F_{xx} \right)$$

(33)

Introducing these expressions into equation (31), substituting for \(w\), \(\bar{w}\) and \(F\) from the equations (8-10), carrying out the \(y\)-integration, substituting for \(f''\) from equation (16), introducing \(\epsilon_{c\lambda} = t/cR\) and the usual nondimensional parameters, yields (see Appendix A)

$$\delta = \frac{\epsilon}{\epsilon_{c\lambda}} = \frac{L}{R} \int_{0}^{L} \left[ w_{0} + \frac{1}{2} w_{1} (w_{1} + 2\bar{w}_{1}) \right] \, dx$$

$$+ \frac{ct}{2L} \int_{0}^{L} \left[ w_{0}' (w_{0}' + 2\bar{w}_{0}') + \frac{1}{2} w_{1}' (w_{1}' + 2\bar{w}_{1}') \right] \, dx + \frac{t^{2}}{4LR} q \int_{0}^{L} w_{0}'' \, dx$$

(34)

This equation, expressed in terms of the unified vector variable \(Y\), represents the one additional equation needed when solving the 2-point boundary value problem (30) using increments in "end shortening" instead of increments of axial loading.

As equation (34) reveals it may also be useful to use increments in \(\delta_{NL} = \delta - \lambda\) in order to extend the response curve beyond the limit point.
3. NUMERICAL ANALYSIS

Due to the highly nonlinear nature of the above 2-point boundary value problem, anything but a numerical solution is out of the question. All of the known numerical techniques for the solution of nonlinear equations involve iterative improvements of initial guesses of the solution.

Here in order to solve the above nonlinear 2-point boundary value problem (30, 34) while using increments in "end-shortening" Newton's discretized method of quasilinearization, a quadratically convergent method is applied.

3.1. Linearization of the equations

The assumed forms for out-of-plane displacements, the Airy stress function, and the loading parameter λ are written as the sum of initial values plus correction terms. The initial values are considered to be known values, while the correction terms are the unknowns. Writing in detail, the out-of-plane displacement is represented by

$$w = t \left\{ \frac{\sqrt{2}}{c} \left( \lambda + \delta \lambda \right) + w_0(x) + \delta w_0(x) + \left[ w_1(x) + \delta w_1(x) \right] \cos \bar{y} \right\}$$

(35)

The Airy stress function is written as

$$F = \frac{E\rho}{c^2} \left[ \frac{-w_0}{2} \left( \lambda + \delta \lambda \right) + f_0(x) + \delta f_0(x) + \left[ f_1(x) + \delta f_1(x) \right] \cos \bar{y} \right]$$

$$+ \left[ f_2(x) + \delta f_2(x) \right] \cos 2\bar{y}$$

(36)

Elaborating in the same way the equations (17-20) and (34) and dropping terms with products of correction terms yields the following set of linearized equations for the correction terms

$$n^4 \delta f_1 - 2n^2 \delta f_1'' + \delta f_1^{IV} = \frac{cE}{R} n^2 (w_1 + A_\lambda) \delta w_0'' - \frac{cE}{R} n^2 \left( w_0'' + A_0'' \right) \delta w_1$$

$$-c \delta w_1'' = -n^4 f_1 + 2n^2 f_1'' - f_1^{IV} + \frac{cE}{R} n^2 \left( (w_1 + A_\lambda) w_0'' + A_0'w_1 \right) + cw_1''$$

(37)
\[ 16 n^4 \delta f_2 - 8 n^2 \delta f_1' + \delta f_1^\IV - \frac{c}{2} \frac{t}{R} n^2 (w_1' + A_1') \delta w_1 + c \frac{t}{R} n^2 (w_1' + A_1') \delta w_1' = -16 n^4 f_2 + 8 n^2 f_1' - f_1^\IV + \frac{c}{2} \frac{t}{R} n^2 [(w_1' + A_1') w_1 + A_1' w_1 - (w_1' + 2A_1') w_1] \] 

(38)

\[ 2cn^2 (w_1' + A_1') \delta f_1 + 4cn^2 (w_1' + A_1') \delta f_1' + 2cn^2 (w_1' + A_1') \delta f_1'' + 4c^2 \frac{R}{t} \delta w_0 + 4c\lambda \delta w_0' + 4c(A_0' + w_0') \delta \lambda + \frac{c}{2} \frac{t}{R} \delta w_0^\IV + (2c^2 n^2 (w_1' + A_1') + 2cn^2 f_1') \delta w_1 + 2cn^2 f_1' \delta w_1 + 2cn^2 f_1' \delta w_1' = -2cn^2 [(w_1' + A_1') f_1 + 2 (w_1' + A_1') f_1' + (w_1 + A_1) f_1''] - 4c^2 \frac{R}{t} w_0 - 4c\lambda (A_0' + w_0') - c^2 n^2 (2A_1 + w_1) w_1 \] 

(39)

\[ 4cn^2 (w_0' + A_0') \delta f_1^\IV - 4cn^2 (w_1' + A_1') \delta f_2^\IV + 8cn^2 (w_1' + A_1') \delta f_2 + 8cn^2 (w_1' + A_1') \delta f_2^\IV + 4cn^2 (w_1' + A_1') \delta f_2^\IV \]

\[ + [n^4 \frac{t}{R} + 2cn^2 f_2' + 4cn^2 w_0 + \frac{c}{R} c^2 [3w_1 w_1' + 6w_1 A_1' + 2A_1 A_1'] \] \( \delta w_1 \)

\[ + 8cn^2 f_2' \delta w_1 + (-2 \frac{t}{R} n^2 + 4c\lambda + 8cn^2 f_2) \delta w_1' + 4c(A_1' + w_1') \delta \lambda + \frac{c}{R} \delta w_1^\IV = -4cn^2 (w_0' + A_0') f_1 - 4c \frac{R}{t} f_1' - 8cn^2 (w_1' + A_1') f_2 - 8cn^2 (w_1' + A_1') f_2' - 4c^2 n^2 (w_1 + A_1) w_0 - n^4 \frac{t}{R} w_1 \]

\[-n^4 \frac{t}{R} c^2 [(A_1 + w_1)(2A_1 + w_1) w_1] + 2 \frac{c}{R} n^2 w_1' - 4c\lambda (A_1' + w_1') - \frac{t}{R} w_1^\IV \] 

(40)
\[ \delta \lambda + \frac{c_v R}{L} \int_0^L \left[ \delta \omega_0^+ + \frac{t_n^2}{2R} \delta \omega_1^+ (w_1^+ + A_1^+) \right] \, dx \]

\[ + \frac{ct}{2L} \int_0^L \left[ 2\delta \omega_0^+ (w_0^+ + A_1^+) + \delta \omega_1^+ (w_1^+ + A_1^+) \right] \, dx + \frac{t_n^2}{4LR} \int_0^L \delta \omega_0^+ \, dx \]

\[ = \delta \lambda - \frac{c_v R}{L} \int_0^L \left[ \omega_0^+ + \frac{t_n^2}{4R} \omega_1^+ (w_1^+ + 2A_1^+) \right] \, dx \]

\[ - \frac{ct}{2L} \int_0^L \left[ \omega_0^+ (w_0^+ + 2A_1^+) + \frac{1}{2} \omega_1^+ (w_1^+ + 2A_1^+) \right] \, dx - \frac{t_n^2}{4LR} \int_0^L \delta \omega_0^+ \, dx \]  

(41)

Starting with some initial values (see par. 3.5), equations (37-41) are used in an iterative scheme converging to a solution of the nonlinear 2-point boundary value problem (30, 34).

3.2 Discretization of the linearized equations

In order to discretize the linearized equations (37, 41), let us choose, between \( \bar{x} = 0 \) and \( \bar{x} = \frac{L}{R} \), a grid of \( N+1 \) points, spaced at a constant distance \( h \) apart. Using the well known trapezoidal rule of numerical integration we can write for the integral of a variable \( g(\bar{x}) \) over the interval \( 0 < \bar{x} < \frac{L}{R} \).

\[ \int_0^{\frac{L}{R}} g(\bar{x}) \, d\bar{x} = h \left( \frac{1}{2} g_0^+ \sum_{i=1}^{N-1} g_i^+ + \frac{1}{2} g_n^+ \right) \]  

(42)

where \( g_i^+ = g(ih) \). The global error for equations (42) is of the order \( h^2 \). Application of the trapezoidal rule of numerical integration to equation (41) yields, introducing an error of the order of \( h^2 \).

\[ \delta \lambda + \frac{c_v R h}{L} \left[ \frac{1}{2} \delta \omega_0^+ + \frac{t_n^2}{4R} \delta \omega_1^+ (w_1^+ + A_1^+) \right] + \sum_{i=1}^{N-1} \left[ \delta \omega_0^+ (w_0^+ + A_1^+) \right] + \delta \omega_1^+ (w_1^+ + A_1^+) \]

\[ + \frac{1}{2} \delta \omega_1^N \left[ w_1^N + A_1^N \right] + \frac{ct}{2L} \left[ \delta \omega_0^+ (w_0^+ + A_1^+) \right] + \frac{1}{2} \delta \omega_1^N \left( w_1^N + A_1^N \right) \]

\[ + \sum_{i=1}^{N-1} \left[ 2\delta \omega_0^+ (w_0^+ + A_1^+) + \delta \omega_1^+ (w_1^+ + A_1^+) \right] + \delta \omega_1^+ (w_1^+ + A_1^+) + \frac{1}{2} \delta \omega_1^N \left( w_1^N + A_1^N \right) \]

\[ + \frac{t_n^2}{4LR} \left[ \frac{1}{2} \delta \omega_0^+ + \sum_{i=1}^{N-1} \delta \omega_1^+ + \frac{1}{2} \delta \omega_0^N \right] = \delta \lambda - \frac{c_v R h}{L} \left[ \frac{1}{2} \omega_0^+ + \frac{t_n^2}{8R} \omega_1^+ (w_1^+ + 2A_1^+) \right] \]
\[
\begin{align*}
N-1 & \sum_{i=1}^{N-1} \left[ w_{i0} + \frac{t^n}{4R} w_{i1} (w_{i1} + 2A_{i1}) \right] + \frac{1}{2} w_{0N} + \frac{t^n}{8R} w_{N1} (w_{N1} + 2A_{N1}) \right] - \frac{ct}{2L} \frac{1}{2} w_{o0} (w_{o0} + 2A_{o0}) \\
+ \frac{1}{4} w_{10} (w_{10} + 2A_{10}) & + \sum_{i=1}^{N-1} \left[ w_{0i} (w_{0i} + 2A_{0i}) \right] + \frac{1}{2} w_{1i} (w_{1i} + 2A_{1i}) \\
+ \frac{1}{2} w_{Nw} (w_{Nw} + 2A_{Nw}) & + \frac{1}{4} w_{N1} (w_{N1} + 2A_{N1}) \right] - \frac{t^n}{4LR} \sum_{i=1}^{N-1} \left[ \frac{1}{2} w_{oi} + \frac{1}{2} w_{o0} \right] \\
(43)
\end{align*}
\]

The linearized equations for the correction terms (37-40 and 43) are converted to finite difference equations by using central difference formulae for derivatives of the correction term variables. The linearized equations are then written as matrix equations. In order to solve the resulting central difference equations it was decided to use Arbocz' modification of Potters' method (see par. 5), which application is possible if the matrix central difference equations include values at only three points. The central difference formulae for first and second derivatives involve values of the variables at 3 different points. However, the central difference formula for fourth derivative involves values for 5 different points. It is therefore necessary to carry along second derivatives of variables in the solution vectors of the central difference equations. Fourth derivatives are then computed as second derivatives of the second derivative variables. Using central difference formulae the first and fourth derivatives at a point \( i \) of a variable \( g \) are written.

\[
g'_i = \frac{g_{i+1} - g_{i-1}}{2h} (44)
\]

\[
g''_i = \frac{g_{i+1} - 2g_i + g_{i-1}}{h^2} (45)
\]

The error for these equations (44) and (45) is also of the order of \( h^2 \). Using the formulae (44) and (45) on first and fourth derivatives of the correction term variables in equations (37-40) and (43) yields five central difference equations

\[
\begin{align*}
\frac{1}{2} \delta f_{i1} & + n^4 \delta f_{i1} - 2n^2 \delta f_{i1}' - \frac{2}{2} \delta f_{i1}'' - \frac{ct}{R} n^2 (w_{i1} + A_{i1}) \delta w_{i1}' \\
& - \frac{ct}{R} n^2 (w_{i1} + A_{i1}) \delta w_{i1} - c \delta w_{i1}' + \frac{1}{h^2} \delta f_{i1}' & = -n^4 f_{i1} + 2n^2 f_{i1}' \\
-\frac{ct}{R} n^2 \left[(w_{i1} + A_{i1}) w_{i0}' + A_{i1}' w_{i1} \right] + c \delta w_{i1}'
(46)
\end{align*}
\]
\[ \begin{align*}
\frac{1}{h^2} \delta f_{1}''' & - \frac{c}{2h R} n^2 (w_1'' + A_1') \delta w_{1}''_{i-1} + 16 n^4 \delta f_{2} - (8n^2) \\
\frac{2}{h^2} \delta f_{1}'' & - \frac{c}{2R} n^2 (w_1'' + A_1'') \delta w_{1}'' - \frac{c}{2} \frac{Oh}{R} n^2 (w_1'' + A_1) \delta w_{1}'' \\
\frac{1}{h^2} \delta f_{2}'' & + \frac{c}{2R} n^2 (w_1'' + A_1) \delta w_{1}''_{i+1} = -16n^4 f_{2} + 8n^2 f_{1}'' \\
-\frac{c}{2} \frac{Oh}{R} n^2 (w_1 + A_1) w_1'' + A_1' w_1 - (w_1'' + 2A_1') w_1 \\
- \frac{2cn}{h} (w_1 + A_1) \delta f_{1}'' + \frac{c}{hR} \delta w_{0}''_{i-1} - \frac{2cn^2}{h} f_{1}'' \delta w_{0}'' - 2cn^2 (w_1'' + A_1') \delta f_{1}'' + \\
+2cn^2 (w_1 + A_1) \delta f_{1}'' + 4c^2 \frac{R}{c} \delta w_{0}'' + 4c \delta w_{0}''_{i-1} \\
- \frac{2c}{hR} \delta w_0'' + [2cn^2 (w_1 + A_1) + 2cn^2 f_{1}''_{i}] \delta w_0 + 2cn^2 f_{1}'' \delta w_1'' \\
+ \frac{2cn^2}{h} (w_1 + A_1) \delta f_{1}''_{i+1} + \frac{c}{hR} \delta w_0''_{i+1} + \frac{2cn^2}{h} f_{1}'' \delta w_1''_{i+1} + 4c (A_0'' + w''_{0}) \lambda \\
= -2cn^2 [(w_1'' + A_1') f_{1}''_{i+1} + 2 (w_1'' + A_1') f_{1}''_{i} + (w_1 + A_1) f_{1}''_{i}] - 4c^2 \frac{R}{c} \delta w_0'' \\
- \frac{c}{h} \delta w_0''_{i-1} - 4c \lambda (A_0'' + w''_{0}) - c^2 n^2 (2A_1 + w_1) w_1'' \\
- \frac{4cn^2}{h} (w_1'' + A_1') \delta f_{2}''_{i-1} - \frac{4cn^2}{h} f_{1}'' \delta w_1''_{i-1} + \frac{c}{h^2} \delta w_1''_{i-1} + 4c n^2 (w_1'' + A_1') \delta f_{1}''_{i-1} \\
+ 4c \frac{R}{c} \delta f_{1}'' + 8cn^2 (w_1'' + A_1') \delta f_{2}'' + 2cn^2 (w_1 + A_1) \delta f_{1}'' \\
+ 4c^2 n^2 (w_1 + A_1) \delta w_0'' + 4cn^2 f_{1}'' \delta w_0'' + [n^4 \frac{c}{h} + 2cn^2 f_{2}'' + 4c^2 n^2 w_0'' ] \\
+ n^4 \frac{c}{h} [3w_1^2 + 6w_1 A_1 + 2A_1^2] \delta w_1 + (-2c \frac{R}{c} n^2 + 4c \lambda + 8cn^2 f_{2}'' \\
- \frac{2c}{h^2} \delta w_{1}' + \frac{4cn^2}{h} (w_1'' + A_1') \delta f_{1}'' + \frac{4cn^2}{h} f_{2}'' \delta w_1''_{i+1} \\
+ \frac{c}{h^2} \delta w_1''_{i+1} + 4c (A_1'' + w_1') \lambda = -4cn^2 (w_1'' + A_1') f_{1}''_{i} \\
- 4c \frac{R}{c} f_{1}''_{i-1} - 8cn^2 (w_1'' + A_1') f_{2}'' - 8cn^2 (w_1'' + A_1') f_{2}'' \\
- 2cn^2 (w_1 + A_1) f_{1}''_{i-1} - 4c^2 n^2 (w_1 + A_1) w_0'' - n^4 \frac{c}{h} w_1'' \\
\end{align*} \]
\[-n^4 \frac{t}{R} c^2 (A_{1_1} + w_{1_1})(2A_{1_1} + w_{1_1}) w_{1_1} + 2 \frac{t}{R} n^2 w_{1_1}^2 - 4c\lambda (A_{1_1} + w_{1_1}) \]

\[-t \frac{\text{R}^{\text{IV}}}{w_{1_1}} \]

\[-\frac{ct}{4L} (w_{0_0} + A_{0_0}) \delta w_{0_1} - \frac{ct}{8L} (w_{1_0} + A_{1_0}) \delta w_{1_1} + \frac{[cvRh]}{2L} \]

\[-\frac{ct}{2L} (w_{0_0} + A_{0_0}) \delta w_{0_0} + \frac{[cvtn^2h]}{4L} (w_{1_0} + A_{1_0}) - \frac{ct}{4L} (w_{1_1} + A_{1_1}) \delta w_{1_1} + \frac{t \cdot \text{h}q}{8LR} \delta w_{0_0} \]

\[+ \frac{[cvRh]}{L} + \frac{ct}{4L} [w_{0_0} + A_{0_0} - 2 (w_{0_2} + A_{0_2})] \delta w_{0_2} + \frac{[cvtn^2h]}{2L} (w_{1_1} + A_{1_1}) \]

\[+ \frac{ct}{4L} [\frac{1}{2} (w_{0_1} + A_{0_1}) - (w_{1_2} + A_{1_2})] \delta w_{1_2} + \frac{t \cdot \text{h}q}{4LR} \delta w_{0_1} + \Sigma [\frac{[cvRh]}{L}] \]

\[+ \frac{ct}{2L} (w_{0_1} + A_{0_1} - w_{1_1} + A_{1_1}) \delta w_{0_1} + \frac{[cvtn^2h]}{2L} (w_{1_1} + A_{1_1}) \]

\[+ \frac{ct}{4L} (w_{1_1} + A_{1_1} - w_{1_1} + A_{1_1}) \delta w_{1_1} + \frac{t \cdot \text{h}q}{4LR} \delta w_{0_1} + \frac{[cvRh]}{L} \]

\[+ \frac{ct}{4L} [2(w_{0_N-2} + A_{0_N-2}) - w_{0_0} - A_{0_0}] \delta w_{0_0} + \frac{cvtn^2h}{2L} (w_{1_N-1} + A_{1_N-1}) \]

\[+ \frac{ct}{4L} [2(w_{1_N-2} + A_{1_N-2}) - \frac{1}{2} (w_{1_1} + A_{1_1})] \delta w_{1_1} + \frac{t \cdot \text{h}q}{4LR} \delta w_{0_1} + \frac{[cvRh]}{2L} \]

\[\delta - \frac{\text{h}q}{8LR} \delta w_{0_1} + \frac{ct}{4L} (w_{0_0} + A_{0_0}) \delta w_{0_0} + \frac{ct}{8L} (w_{1_1} + A_{1_1}) \delta w_{1_1} + \delta \lambda \]

\[= \delta \lambda - \frac{cvRh}{L} \frac{1}{2} w_{0_0} + \frac{t^n_2}{8R} w_{1_0} (w_{1_1} + 2A_{1_1}) + \Sigma [w_{0_0} + \frac{t^n_2}{4R} w_{1_1} (w_{1_1} + 2A_{1_1})] \]

\[+ \frac{1}{n} w_{0_n} + \frac{t^n_2}{8R} w_{1_n} (w_{1_n} + 2A_{1_n}) \] - \[\frac{ct}{2L} \frac{1}{2} w_{0_0} (w_{0_0} + 2A_{0_0}) \]

\[+ \frac{1}{4} w_{1_0} (w_{1_0} + 2A_{1_0}) + \Sigma [w_{0_0} (w_{0_0} + 2A_{0_0}) + \frac{1}{2} w_{1_0} (w_{1_0} + 2A_{1_0})] \]

\[+ \frac{1}{2} w_{0_0} (w_{0_0} + 2A_{0_0}) + \frac{1}{4} w_{1_0} (w_{1_0} + 2A_{1_0}) - \frac{t^n_2}{8R} \frac{1}{2} w_{0_0} + \Sigma [w_{0_0} (w_{0_0} + 2A_{0_0})] \]

Since correction term second derivatives are carried along as unknowns, four additional equations are needed.

These additional equations are the identities between the second derivative correction variables and the second derivatives of the corresponding variables con-
structured by the central difference formulae.

\[ b_i'' = \frac{g_{i+1} - 2g_i + g_{i-1}}{h^2} \]  

(51)

The error for this equation (51) is also of the order of \( h^2 \). The additional equations are

\[-\frac{1}{2} \delta w_{i-1} + \frac{2}{h^2} \delta w_i + \frac{1}{2} \delta w_{i+1} = -w_i'' + \frac{1}{2} (w_{i-1} - 2w_i + w_{i+1}) \]  

(52)

\[-\frac{1}{2} \delta w_{i-1} + \frac{2}{h^2} \delta w_i + \frac{1}{2} \delta w_{i+1} = -w_i'' + \frac{1}{2} (w_{i-1} - 2w_i + w_{i+1}) \]  

(53)

\[-\frac{1}{2} \delta f_{i-1} + \frac{2}{h^2} \delta f_i + \frac{1}{2} \delta f_{i+1} = -f_i'' + \frac{1}{2} (f_{i-1} - 2f_i + f_{i+1}) \]  

(54)

\[-\frac{1}{2} \delta f_{i-1} + \frac{2}{h^2} \delta f_i + \frac{1}{2} \delta f_{i+1} = -f_i'' + \frac{1}{2} (f_{i-1} - 2f_i + f_{i+1}) \]  

(55)

The central difference equations (46-50) and (52-55) are applied to all points from 0 to \( N \). The resulting system of \( N+1 \) eight dimensional matrix equations and one vector equation with \( N+1 \) eight dimensional vector unknowns and one scalar unknown are written in terms of the single matrix equation.

\[
\begin{bmatrix}
L_0 & D_0 & U_0 & 0 \\
L_1 & D_1 & U_1 & 0 \\
& & \ddots & \ddots \\
0 & 0 & L_{N-1} & D_{N-1} & U_{N-1} \\
& & & \ddots & \ddots \\
0 & 0 & 0 & L_N & D_N & U_N \\
& & & & \ddots & \ddots \\
\delta_1 & \delta_0 & \delta_{N-1} & \delta_N & \delta_{N+1} & \delta_0^t \\
\end{bmatrix} 
\begin{bmatrix}
\xi_0 \\
\xi_1 \\
\vdots \\
\xi_{N-1} \\
\xi_N \\
\xi_{N+1} \\
\end{bmatrix} = 
\begin{bmatrix}
\xi_0 \\
\xi_1 \\
\vdots \\
\xi_{N-1} \\
\xi_N \\
\xi_{N+1} \\
\end{bmatrix}
\begin{bmatrix}
\delta_1 \\
\delta_0^t \\
\delta_1^t \\
\delta_{N-1}^t \\
\delta_N^t \\
\delta_{N+1}^t \\
\end{bmatrix}
\]

(56)

where \( L, \ D, \) and \( U \) are \((8\times8)\) matrices, \( \xi, \ \phi, \ \chi \) and \( \eta \) are 8 dimensional vectors and \( \delta \), \( \chi \), and \( \eta \) are scalars.

It remains to omit 16 unknowns by enforcement of the boundary conditions (see also Appendix B).
3.3 Enforcement of the boundary conditions

Since only non-derivative and second derivative correction terms are carried along as unknowns in the equations (46-50) and (52-55) the boundary conditions have to be expressed in terms of these unknowns only. Using the central difference formula (44) the first derivative correction terms are related to the corresponding non-derivative correction terms. The third derivative correction terms are related to the corresponding second derivative correction terms by using the central difference formula

\[
g^{'''}_{i} = \frac{g^{''}_{i+1} - g^{''}_{i-1}}{2h}
\]  

(57)

The error for this equation (57) is also of the order of \( h^2 \). Application of the equations (44) and (57) to the boundary conditions (21-29) at the points \( i=0 \) and \( i=N \) yields

\[
\delta w_0 = -\frac{v}{c} \delta \lambda \\
\delta w^\prime_0 = q \delta \lambda \\
\delta f^1_{-1} = \delta f^1_1 \\
\delta f^2_{-1} = \delta f^2_1 \\
\delta w_0 = \delta w^\prime_0 = \delta f^1_0 = \delta f^2_0 = 0
\]

and

\[
\delta w_0 = -\frac{v}{c} \delta \lambda \\
\delta w^\prime_0 = q \delta \lambda \\
\delta f^1_{N+1} = \delta f^1_{N-1} \\
\delta f^2_{N+1} = \delta f^2_{N-1} \\
\delta w_0 = \delta w^\prime_0 = \delta f^1_0 = \delta f^2_0 = 0
\]

(58)
SS-2: \[ \delta \omega_0 = - \frac{\nu}{c} \delta \lambda \]
\[ \delta \omega_{0}^{\prime} = - q \delta \lambda \]
\[ \delta f_{0}^{\prime} = \delta f_{1}^{\prime} \]
\[ \delta f_{1}^{\prime} = \delta f_{1}^{\prime} + cn^2 \frac{t}{R} A_{1/0} (\delta \omega_{0}^{\prime} - \delta \omega_{0}^{\prime}) + c(\delta \omega_{1}^{\prime} - \delta \omega_{1}^{\prime}) \]
\[ \delta f_{2}^{\prime} = \delta f_{2}^{\prime} + c \frac{n}{2} \frac{t}{R} A_{1/0} (\delta \omega_{1}^{\prime} - \delta \omega_{1}^{\prime}) \]
\[ \delta \omega_{1}^{\prime} = \delta \omega_{1}^{\prime} = 0 \]

and

(59)

\[ \delta \omega_{0} = - \frac{\nu}{c} \delta \lambda \]
\[ \delta \omega_{0}^{\prime} = - q \delta \lambda \]
\[ \delta f_{0}^{\prime} = \delta f_{N+1}^{\prime} \]
\[ \delta f_{1}^{\prime} = \delta f_{N+1}^{\prime} - cn^2 \frac{t}{R} A_{1/0} (\delta \omega_{0}^{\prime} - \delta \omega_{0}^{\prime}) - c(\delta \omega_{1}^{\prime} - \delta \omega_{1}^{\prime}) \]
\[ \delta f_{2}^{\prime} = \delta f_{N+1}^{\prime} - c \frac{n}{2} \frac{t}{R} A_{1/0} (\delta \omega_{1}^{\prime} - \delta \omega_{1}^{\prime}) \]
\[ \delta \omega_{1}^{\prime} = \delta \omega_{1}^{\prime} = 0 \]

SS-3: \[ \delta \omega_0 = - \frac{\nu}{c} \delta \lambda \]
\[ \delta \omega_{0}^{\prime} = - q \delta \lambda \]
\[ \delta \omega_{1}^{\prime} = \delta \omega_{1}^{\prime} = \delta f_{1}^{\prime} = \delta f_{2}^{\prime} = \delta f_{2}^{\prime} = 0 \]

and

(60)
\begin{align}
\delta w_0 &= -\frac{v}{c} \delta \lambda \\
\delta w_0' &= \bar{q} \delta \lambda \\
\delta w_1 &= \delta w_1' = \delta f_1 = \delta f_1' = \delta f_2 = \delta f_2' = 0 \\
\delta f_{1,0} &= \delta f_{1,0}' = -n^2 v \delta f_{1,0} \\
\delta f_{2,0} &= \delta f_{2,0}' = -4n^2 v \delta f_{2,0} \\
\delta w_1 &= \delta w_1' = 0
\end{align}

and

\begin{align}
\delta w_N &= -\frac{v}{c} \delta \lambda \\
\delta w_N' &= \bar{q} \delta \lambda \\
\delta f_{1,0}' &= -n^2 v \delta f_{1,0} \\
\delta f_{1,N} &= \delta f_{1,N}'' = -n^2 (2+v)(\delta f_{1,N-1} - \delta f_{1,N}) - cn^2 \frac{\epsilon}{R} A_{1N}(\delta w_{0,N} - \delta w_{0,N+1}) \\
&\quad\quad - c(\delta w_{1,N-1} - \delta w_{1,N+1}) \\
\delta f_{2,N} &= \delta f_{2,N}' = -4n^2 v \delta f_{2,N} \\
\delta f_{2,N+1} &= \delta f_{2,N}'' = -4n^2 (2+v)(\delta f_{2,N-1} - \delta f_{2,N}) - \frac{1}{2} cn^2 \frac{\epsilon}{R} A_{1N}(\delta w_{1,N} - \delta w_{1,N+1})
\end{align}
\[ \delta w_1 = \delta w'_1 = 0 \]

\[ \delta w_{0\,N} = -\frac{\nu}{c} \delta \lambda \]

\[ \delta w_{0\,1} = \delta w_{0\,1} \]

\[ \delta w_{1\,1} = \delta w_{1\,1} \]

\[ \delta f_{1\,1} = \delta f_{1\,1} \]

\[ \delta f_{2\,1} = \delta f_{2\,1} \]

\[ \delta w_{0\,0} = \delta f_{1\,0} = \delta f_{2\,0} = 0 \]

and

\[ \delta w_{0\,N} = \frac{\nu}{c} \delta \lambda \]

\[ \delta w_{0\,N+1} = \delta w_{0\,N-1} \]

\[ \delta w_{1\,N+1} = \delta w_{1\,N-1} \]

\[ \delta f_{1\,N+1} = \delta f_{1\,N-1} \]

\[ \delta f_{2\,N+1} = \delta f_{2\,N-1} \]

\[ \delta w_{1\,N} = \delta f_{1\,N} = \delta f_{2\,N} = 0 \]

\[ \delta w_{0\,0} = -\frac{\nu}{c} \delta \lambda \]

\[ \delta w_{0\,-1} = \delta w_{0\,-1} \]

\[ \delta w_{1\,-1} = \delta w_{1\,-1} \]

\[ \delta f_{1\,-1} = \delta f_{1\,-1} \]
\[ \delta f_1'' = \delta f_1'' \]
\[ \delta f_2'' = \delta f_2'' \]
\[ \delta f_1' = \delta f_1' \]
\[ \delta w_1^0 = 0 \]

and

\[ \delta w_0^N = -\frac{\nu}{c} \delta \lambda \]
\[ \delta w_0^{N+1} = \delta w_{N+1}^{N+1} \]
\[ \delta w_1^{N+1} = \delta w_{N+1}^{N+1} \]
\[ \delta f_1'' = \delta f_1'' \]
\[ \delta f_2'' = \delta f_2'' \]
\[ \delta f_1' = \delta f_1' \]
\[ \delta f_2' = \delta f_2' \]
\[ \delta w_1^N = 0 \]

C-3: \[ \delta w_0^0 = -\frac{\nu}{c} \delta \lambda \]
\[ \delta w_0^{-1} = \delta w_{01}^{-1} \]
\[ \delta w_1^{-1} = \delta w_{11}^{-1} \]
\[ \delta w_1^0 = \delta f_{10}^0 = \delta f_{10}'' = \delta f_{20}^0 = \delta f_{20}'' = 0 \]

and

\[ (63) \]
\[ (64) \]
\[ \delta w_0 = -\frac{\gamma \delta \lambda}{c} \]

\[ \delta w_{0, N+1} = \delta w_{0, N-1} \]

\[ \delta w_{1, N+1} = \delta w_{1, N-1} \]

\[ \delta w_{1} = \delta f_{1} = \delta f_{1}' = \delta f_{2} = \delta f_{2}' = 0 \]

\[ \delta f_{10}' = -n^2 v \delta f_{10} \]

\[ \delta f_{10}' = \delta f_{10}' + n^2 (2+v) (\delta f_{10} - \delta f_{10}) \]

\[ \delta f_{20}' = -4n^2 v \delta f_{20} \]

\[ \delta f_{21}' = \delta f_{21}' + 4n^2 (2+v) (\delta f_{21} - \delta f_{21}) \]

\[ \delta w_{0} = 0 \]

and

\[ \delta w_{0} = -\frac{\gamma \delta \lambda}{c} \]

\[ \delta w_{0, N+1} = \delta w_{0, N-1} \]

\[ \delta w_{1, N+1} = \delta w_{1, N-1} \]

\[ \delta f_{10}' = -n^2 v \delta f_{10} \]

\[ \delta f_{10}' = \delta f_{10}' - n^2 (2+v) (\delta f_{10} - \delta f_{10}) \]

\[ \delta f_{20}' = -4n^2 v \delta f_{20} \]
\[ \delta \xi_{n+1}^2 = \delta \xi_{n-1}^2 - 4n^2(2+n)(\delta \xi_{n-1}^2 - \delta \xi_{n+1}^2) \]

Symm.: \[ \delta \omega_{0-1} = \delta \omega_{01} \]
\[ \delta \omega_{0-1}'' = \delta \omega_{01}'' \]
\[ \delta \omega_{1-1} = \delta \omega_{11} \]
\[ \delta \omega_{1-1}'' = \delta \omega_{11}'' \]
\[ \delta \xi_{1-1} = \delta \xi_{11} \]
\[ \delta \xi_{1-1}'' = \delta \xi_{11}'' \]
\[ \delta \xi_{2-1} = \delta \xi_{21} \]
\[ \delta \xi_{2-1}'' = \delta \xi_{21}'' \]

and

\[ \delta \omega_{n+1} = \delta \omega_{n-1} \]
\[ \delta \omega_{n+1}'' = \delta \omega_{n-1}'' \]
\[ \delta \omega_{n+1} = \delta \omega_{n-1} \]
\[ \delta \omega_{n+1}'' = \delta \omega_{n-1}'' \]
\[ \delta \xi_{n+1} = \delta \xi_{n-1} \]
\[ \delta \xi_{n+1}'' = \delta \xi_{n-1}'' \]
\[ \delta \xi_{n+1} = \delta \xi_{n-1} \]
\[ \delta \xi_{n+1}'' = \delta \xi_{n-1}'' \]

(66)
Omitting 16 unknowns out of the matrix equation (56) by enforcement of the chosen boundary conditions, this system is rewritten in terms of the following bordered block tridiagonal matrix equation $A\mathbf{x} = \mathbf{y}$:

$$
\begin{bmatrix}
  D_0 & U_0 & 0 & & \\
  L_1 & D_1 & U_1 & & \\
  & L_{N-1} & D_{N-1} & U_{N-1} & \\
  & & L_N & D_N &
  \\
 n_0 & t & t & 0 & t &
\end{bmatrix}
\begin{bmatrix}
  x_0 \\
  x_1 \\
  \vdots \\
  x_{N-1} \\
  x_t
\end{bmatrix}
= 
\begin{bmatrix}
  y_0 \\
  y_1 \\
  \vdots \\
  y_{N-1} \\
  y_t
\end{bmatrix}
$$

(67)

All local errors in equation (67) are of the order of $h^2$, hence the global error of this system is also of the order of $h^2$.

As mentioned before, in order to solve the system (67), Arbocz' modification of Potters' method (see par. 4) is used.

### 3.4 Completion of the solution

The solution vector $\mathbf{x}$ as in equation (67) is composed of correction term variables and second derivatives of the correction term variables. The non-derivative and second derivative correction term variables omitted by enforcement of the chosen boundary conditions are obtained using (58-66).

After addition of all those correction term variables to the corresponding variables, first and fourth derivative variables are computed from the non-derivative and second derivative variables using the formulae (44-45).

### 3.5 Initial values

Starting with some initial values, the system (67) is used in an iterative scheme converging to a solution of the nonlinear 2-point boundary value problem (30, 34).

These initial values must satisfy the chosen boundary conditions and, in order to get convergence to the desired solution, may not differ too much from this
solution. Hence for low values of the loading parameter $\lambda$ and the "end shortening" parameter $\delta$, for which the nonlinear terms in the equations (46-50) and (52-55) have just a small effect in the solution and for which the limit point is not yet reached, appropriate initial values are obtained using the linear theory performed by setting all initial values (and so all nonlinear terms) in the equations (46-50) and (52-55) equal to zero. Doing this, the vectors $y_L$ in the system (67) are equal to zero, and $y_L$ is equal to $\delta$.

For values of $\delta$ and $\lambda$ close to or beyond the limit point, the solution for adjacent values of $\delta$ and $\lambda$ is used as starting value, as can be done anywhere.

3.6 Further remarks

In order to obtain an iterative scheme converging to a solution of the nonlinear 2-point boundary value problem (30, 34) when using increments in $\delta_{NL}$ instead of increments in $\delta$, one just has to omit the loading parameter $\lambda$ and its correction term variable $\delta\lambda$ out of equation (50), and to replace the parameter $\delta$ by $\delta_{NL}$ resulting in a system of the same appearance as equation (67).

When solving the nonlinear 2-point boundary value problem (30, 34) using increments in loading instead of "end shortening", the correction term $\delta\lambda$ in the equations (46-49) is equal to zero, hence the vector $\bar{y}$ in the system (67) can be set to zero. In addition one has to replace the correction term variable $\delta\lambda$ in equation (50) by the correction term variable $-\delta\delta$. However it is clear that using increments in loading, it is advantageous to solve the nonlinear 2-point boundary value problem (30) in a iterative scheme. This scheme is represented by the block tridiagonal partition of the system (67)

\[
\begin{bmatrix}
D_0 & U_0 \\
L_1 & D_1 & U_1 \\
& & \ddots & \ddots \\
& & & D_{N-1} & U_{N-1} \\
& & & & D_N
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
\vdots \\
x_{N-1} \\
x_N
\end{bmatrix}
= 
\begin{bmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{N-1} \\
y_N
\end{bmatrix}
\]

(68)

and may be solved using Potters' method (see par. 4). If desired, next, after the asked level of convergence is reached, the corresponding "end shortening" can be obtained using equation (50) where all the correction term variables are equal to zero and only $\delta$ is unknown.
4. POTTER'S METHOD

The block tridiagonal system (68) may be solved using Potters' method (Ref. 11). Let us assume that the solution vector \( x_i \) at the \( i^{th} \) point can be written in terms of the corresponding solution vector \( x_{i+1} \) at the \( i+1^{th} \) point by

\[
x_i = A_i x_{i+1} + b_i
\]  

(69)

Substituting this expression into the matrix equation

\[
L_i x_{i-1} + D_i x_i + U_i x_{i+1} = y_i
\]  

(70)

yields

\[
(L_i A_i x_{i-1} + D_i) x_i + U_i x_{i+1} + (L_i b_{i-1} - y_i) = 0
\]  

(71)

which gives

\[
x_i = -(L_i A_i x_{i-1} + D_i)^{-1} U_i x_{i+1} - (L_i A_i x_{i-1} + D_i)^{-1} (L_i b_{i-1} - y_i)
\]  

(72)

providing that the inverse exists. Comparison of the equations (69) and (72) yields the recurrence relations

\[
A_i = -(L_i A_i x_{i-1} + D_i)^{-1} U_i
\]  

(73)

\[
b_i = -(L_i A_i x_{i-1} + D_i)^{-1} (L_i b_{i-1} - y_i)
\]  

(74)

To start the solution of the recurrence relations it is necessary to determine starting \( A_0 \) and \( b_0 \) values.

Therefore the first equation in system (68)

\[
D_0 x_0 + U_0 x_1 = y_0
\]  

(75)

will be solved for \( x_0 \) in terms of \( x_1 \) giving values for \( A_0 \) and \( b_0 \). Providing \( D_0 \) is regular, equation (75) gives.
\[ x_0 = -D_0^{-1} U_0 x_1 + D_0^{-1} x_0 \]  
\[ a_0 = -D_0^{-1} u_0 \]  
\[ b_0 = D_0^{-1} x_0 \]  

Comparison of the equations (69) and (76) yields:

Starting with these values \((A_1, \ldots, A_{N-1})\) and \((b_1, \ldots, b_{N-1})\) are computed via the recurrence relations (73) and (74).

Next the last equation in the system (68) must be solved for \(x_n\) in order to perform the backwards sweep obtaining all solution vectors \(x_i\).

\[ L_N x_{N-1} + D_N x_N = y_N \]  

Substituting the Potters' equation (69) at \(i=N\).

\[ x_{N-1} = A_{N-1} x_N + b_{N-1} \]  

into equation (79) yields

\[ (L_N A_{N-1} + D_N) x_N + L_N b_{N-1} - x_N = 0 \]  

This gives

\[ x_N = -(L_N A_{N-1} + D_N)^{-1} (L_N b_{N-1} - x_N) \]  

providing the inverse exists. Next the solution vectors \((x_{N-1}, \ldots, x_0)\) are found using the Potters' equation (69).
5. ARBOCZ' MODIFICATION OF POTTERS' METHOD

Because the block tridiagonal partition of the system (67) is singular at the limit point, the system (67) can not be solved by using a standard profile method or a partitioning method of solution. Therefore Arbocz developed a modified scheme for Potters' method taking account of the additional row and column. Let us assume that the solution vector $\mathbf{x}_i$ at the $i^{\text{th}}$ point of the system (67) can be written in terms of the corresponding vector $\mathbf{x}_{i+1}$ at the $i+1^{\text{th}}$ point by

$$\mathbf{x}_i = A_1 \mathbf{x}_{i+1} + b_1 \mathbf{x}_t + c_1$$ (83)

substituting this equation into the matrix equation

$$L_1 \mathbf{x}_{i-1} + D_1 \mathbf{x}_i + U_1 \mathbf{x}_{i+1} + E_1 \mathbf{x}_t = Y_1$$ (84)

yields

$$\left( L_1 A_{i-1} + D_1 \right) \mathbf{x}_i + U_1 \mathbf{x}_{i+1} + \left( L_1 b_{i-1} + E_1 \right) \mathbf{x}_t + L_1 c_{i-1} = Y_1$$ (85)

which gives

$$\mathbf{x}_i = - \left( L_1 A_{i-1} + D_1 \right)^{-1} U_1 \mathbf{x}_{i+1} - \left( L_1 A_{i-1} + D_1 \right)^{-1} \left( L_1 b_{i-1} + E_1 \right) \mathbf{x}_t$$

$$- \left( L_1 A_{i-1} + D_1 \right)^{-1} L_1 c_{i-1} - Y_1$$ (86)

providing that the inverse exists. Comparison of the equations (83) and (86) yields the recurrence relations

$$A_1 = - \left( L_1 A_{i-1} + D_1 \right)^{-1} U_1$$ (87)

$$b_1 = - \left( L_1 A_{i-1} + D_1 \right)^{-1} \left( L_1 b_{i-1} + E_1 \right)$$ (88)

$$c_1 = - \left( L_1 A_{i-1} + D_1 \right)^{-1} L_1 c_{i-1} - Y_1$$ (89)

To start the solution of the recurrence relations it is necessary to determine starting $A_1$, $b_1$ and $c_1$ values. Therefore the first equation in the system (67)
\[ D_0 \mathbf{X}_0^+ U_1 \mathbf{X}_1^+ \mathbf{I}_0 \mathbf{X}_t = \mathbf{I}_0 \]  \hfill (90)

will be solved for \( \mathbf{X}_1 \) in terms of \( \mathbf{X} \) giving values for \( A_1, b_1 \) and \( c_1 \). Providing \( D_1 \) is regular, equation (90) gives

\[ \mathbf{X}_0 = -D_0^{-1} U_0 \mathbf{X}_1 - D_0^{-1} \mathbf{I}_0 \mathbf{X}_t + D_0^{-1} \mathbf{I}_0 \]  \hfill (91)

Comparison of the equations (83) and (91) yields:

\[ A_0 = -D_0^{-1} U_0 \]  \hfill (92)

\[ b_0 = -D_0^{-1} \mathbf{I}_0 \]  \hfill (93)

\[ c_0 = D_0^{-1} \mathbf{I}_0 \]  \hfill (94)

Starting with these values (92-94) \( (A_1, \ldots, A_{N-1}), (b_1, \ldots, b_{N-1}) \) and \( (c_1, \ldots, c_{N-1}) \) are computed via the recurrence relations (87-89).

Next the last two equations in the system (67) must be solved for \( \mathbf{X}_{N-1} \) and \( \mathbf{X}_t \) in order to perform the backwards sweep obtaining all solution vectors \( \mathbf{X}_i \).

\[ L_N \mathbf{X}_{N-1} + D_N \mathbf{X}_N + \mathbf{I}_N \mathbf{X}_t = \mathbf{I}_N \]  \hfill (95)

\[ \alpha_0^t \mathbf{X}_0 + \alpha_1^t \mathbf{X}_1 + \cdots + \alpha_N^t \mathbf{X}_N + d_t \mathbf{X}_t = \mathbf{y}_t \]  \hfill (96)

All of the \( \mathbf{X}_i \) terms except for \( \mathbf{X}_N \) must be written in terms of \( \mathbf{X}_N, \mathbf{X}_t \) and the computed \( A_1, b_1 \) and \( c_1 \) terms.

Using the modified Potters equation (83) for the points \( i = N-1 \) and \( i = N-2 \) we get

\[ \mathbf{X}_{N-1} = A_{N-1} \mathbf{X}_N + b_{N-1} \mathbf{X}_t + c_{N-1} \]  \hfill (97)

\[ \mathbf{X}_{N-2} = A_{N-2} \mathbf{X}_{N-1} + b_{N-2} \mathbf{X}_t + c_{N-2} \]  \hfill (98)

Substitution of equation (97) into equation (98) yields

\[ \mathbf{X}_{N-2} = A_{N-2} \mathbf{A}_{N-1} \mathbf{X}_N^+ + \left( A_{N-2} b_{N-1} + b_{N-2} \right) \mathbf{X}_t \]
+ \mathcal{A}_{N-2} \xi_{N-1}^{\leftarrow} \xi_{N-2} \tag{99}

For the points \( i=0, \ldots, N-1 \) the general expression is

\[
\xi_{N-j} = \left( \prod_{k=1}^{j} A_{N-k} \right) \xi_N + \left[ \prod_{l=1}^{j-1} \left( \sum_{k=1}^{l} A_{N-k} b_{N-l} \right) + b_{N-j} \right] x_t \\
+ \left( \prod_{l=1}^{j-1} \left( \sum_{k=1}^{l} A_{N-k} \right) \xi_{N-l} \right) + \xi_{N-j} \tag{100}
\]

which can be represented as

\[
\xi_i = \bar{A}_i \xi_N + \bar{b}_i x_t + \bar{c}_i \tag{101}
\]

where

\[
\bar{A}_i = A_i \bar{A}_{i+1} \tag{102}
\]

\[
\bar{b}_i = A_i \bar{b}_{i+1} + b_i \quad \text{for } i=0, \ldots, N-2 \tag{103}
\]

\[
\bar{c}_i = A_i \bar{c}_{i+1} + c_i \tag{104}
\]

\[
\bar{A}_{N-1} = A_{N-1} \tag{105}
\]

\[
\bar{b}_{N-1} = b_{N-1} \tag{106}
\]

\[
\bar{c}_{N-1} = c_{N-1} \tag{107}
\]

Substitution of equation (101) into equation (95) and (96) yields the following two equations

\[
\left( L_N \bar{A}_{N-1} + \bar{b}_N \right) \xi_N + \left( L_N \bar{b}_{N-1} + \bar{c}_N \right) x_t + \left( L_N \bar{c}_{N-1} - \xi_N \right) = 0 \tag{108}
\]

\[
\left[ \sum_{i=0}^{N-1} \left( \bar{c}_i \bar{A}_i + c_i \right) \xi_N^{\leftarrow} \right] x_t^{\leftarrow} + \left[ \sum_{i=0}^{N-1} \left( \bar{b}_i \bar{A}_i + b_i \right) \xi_N^{\leftarrow} \right] x_t^{\leftarrow} + \left( \sum_{i=0}^{N-1} \left( \bar{c}_i \bar{c}_i + c_i \right) \xi_N^{\leftarrow} \right) - \gamma_t = 0 \tag{109}
\]

which must be solved for \( \xi_N \) and \( x_t^{\leftarrow} \).

Equation (108) gives
\[ x_N = \bar{b}_N x_t + \bar{c}_N \]  

where

\[ \bar{b}_N = b_N = -(I_N A_{N-1} + D_N)^{-1} (I_N \bar{b}_{N-1} + r_N) \]  

\[ \bar{c}_N = c_N = -(I_N A_{N-1} + D_N)^{-1} (I_N \bar{c}_{N-1} - x_N) \]

providing that the inverse exists. Substitution of equation (110) into equation (109) yields

\[ (\bar{a}_N \bar{b}_N + \bar{d}_t) x_t + \bar{a}_N \bar{c}_N + \bar{c}_N + \bar{y}_t = 0 \]

where

\[ \bar{a}_N = \sum_{i=1}^{N-1} a_i \] \[ \bar{b}_N = \sum_{i=1}^{N-1} b_i \]

\[ \bar{d}_t = \sum_{i=1}^{N-1} d_i \]

\[ \bar{y}_t = \sum_{i=1}^{N-1} y_i \]

Equation (113) gives

\[ x_t = -(\bar{a}_N \bar{b}_N + \bar{d}_t)^{-1} (\bar{a}_N \bar{c}_N + \bar{y}_t) \]

Providing that the inverse exists.

Next the solution vector \( x_N \) is found using equation (110), and the solution vectors \( (x_0, \ldots, x_{N-1}) \) are found using equation (101). Note that the solution vectors \( (x_0, \ldots, x_N) \) can also be obtained using equation (83).
6 NUMERICAL RESULTS

The solution scheme for the nonlinear 2-point boundary value problem (30, 34), presented in this report, was coded into the computer program FREIA. A description of the program is given in Appendix C.

Using the program FREIA some test runs were made for the following case:
\[ \frac{L}{R} = 1; \frac{R}{t} = 1000; \nu = 0,3; \]

Boundary conditions C-3 at \( \bar{x} = 0 \) and \( \bar{x} = L/R; \bar{q} = 0; \)
\[ A_0(\bar{x}) = 0,5 \cos(2\pi \bar{x}) \]
\[ A_1(\bar{x}) = -0,05 \sin(\pi \bar{x}) \]
\( n = 13 \)

With the following input parameters:
\[ NI = 50 \quad ; \quad ITMAX = 10 \quad ; \quad NEPSI = 7 \quad ; \quad NPREC = 5 \]
\[ XLDIR = 0,5; \quad RDIT = 1000 \quad ; \quad XNU = 0,3 \quad ; \]
\[ IBC0 = 7 \quad ; \quad IBCNI = 9 \quad ; \quad Q = 0 \]
\[ NCW = 13; \]

Using increments in \( \delta \) and \( \delta_{NL} \).
The numerical results are given in Table 1 and Fig. 2. A close look at Table 1 reveals that it is only possible to get around the limit point by using increments in \( \delta_{NL} \).
7. CONCLUSIONS

The aim of this study was to devise an efficient numerical method for finding an approximate solution to the Donnell-type nonlinear shell equations. By incrementing $\delta_{NL}$ one may get around the limit point which is, by definition, the collapse load of the imperfect shell.
REFERENCES


APPENDIX A

The approximate solution of Donnell's equation for an imperfect cylindrical shell assumes that the initial imperfection surface is represented by

\[ \tilde{w} = tA_0(x) \cdot tA_1(x) \cos nx \]  \hspace{1cm} (1)

The equilibrium state of the axially loaded cylinder is approximated as:

\[ w(x, y) = \frac{t \nu \lambda}{c} + tw_0(x) + tw_1(x) \cos ny \]  \hspace{1cm} (2)

\[ F(x, y) = \frac{E R t^2}{c} \left[ - \frac{\lambda}{2} y^2 + f_0(x) + f_1(x) \cos ny \right. \]

\[ \left. + f_2(x) \cos 2ny \right] \]  \hspace{1cm} (3)

In order to facilitate the substitution of (1), (2) and (3) into the Donnell-type equations, write the spacial derivatives in the Donnell equation in non-dimensional form by using

\[ \frac{1}{R} \frac{d}{dx} = \frac{d}{dx}, \quad \frac{1}{R} \frac{d}{dy} = \frac{d}{dy} \]  \hspace{1cm} (4)

The Donnell shell equations are then

\[ \frac{1}{R^4 E t} \nabla^4 F - \frac{1}{R^3} \frac{1}{2R} L_{NL} (w, w + 2\tilde{w}) = 0 \]  \hspace{1cm} (5)

\[ \frac{E t^3}{12R^4 (1-\nu^2)} \nabla^4 w + \frac{1}{R^3} F, \quad \frac{1}{R^4} L_{NL} (F, w + \tilde{w}) = 0 \]  \hspace{1cm} (6)

where the nonlinear operator \( L_{NL} \) is defined by

\[ L_{NL}(S, T) = S, \quad \frac{\partial}{\partial x} T, \quad \frac{\partial}{\partial y} 2S, \quad \frac{\partial}{\partial x} T, \quad \frac{\partial}{\partial y} T, \quad \frac{\partial}{\partial y} S, \quad \frac{\partial}{\partial x} T, \quad \frac{\partial}{\partial y} \]  \hspace{1cm} (7)

and the two-dimensional biharmonic operator \( \nabla^4 \) is

\[ \nabla^4 = \frac{\partial^4}{\partial x^4} + 2 \frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4} \]  \hspace{1cm} (8)
Performing the various operations on $\nabla$, $W$, and $F$ in equations (5) and (6), using the assumed forms (1), (2) and (3)

$$
\nabla^4 F = \frac{\text{Ert}}{c} \left\{ f^{iv}_0 + f^{iv}_1 \cos ny + f^{iv}_2 \cos 2ny - 2n^2 f^{iv}_1' \cos ny \
- 8n^2 f^{iv}_2' \cos 2ny + n^4 f^{iv}_1 \cos ny + 16n^4 f^{iv}_2 \cos 2ny \right\}
$$

(9)

$w, \frac{\partial}{\partial x} = t[w''_0 + w''_1 \cos ny]$

(10)

$$
L_{NL}(w, w + 2\bar{w}) = -t^2 n^2 (w''_0 + w''_1 \cos ny)(w_1 + 2A_1) \cos ny
$$

$$
-2t^2 n^2 (w'_1 \cos ny + 2A_1') \sin^2 ny
$$

$$
-2t^2 n^2 (w'_1 + 2A_1) \cos ny + w'_1(2A_1' + 2A_1') \cos^2 ny}
$$

(11)

$$
\nabla^4 w = t\{w^{iv}_0 + w^{iv}_1 \cos ny - 2n^2 w'_1 \cos ny + n^4 w_1 \cos ny\}
$$

(12)

$$
F, \frac{\partial}{\partial x} = \frac{\text{Ert}}{c} \left\{ f^{iv}_0 + f^{iv}_1 \cos ny + f^{iv}_2 \cos 2ny \right\}
$$

(13)

$$
L_{NL}(F, w + \bar{w}) = -\frac{\text{Ert}}{c} \left\{ n^2 (w_1 + A_1)(f^{iv}_0 \cos ny + f^{iv}_1 \cos 2ny)
\right. \\
\left. + f^{iv}_2 \cos ny \cos 2ny\right\}
$$

$$
-2 \frac{\text{Ert}}{c} \left\{ (w'_1 + A_1') (f^{iv}_1 \sin ny + 2f^{iv}_2 \sin ny \sin 2ny)
\right. \\
\left. - \frac{\text{Ert}}{c} ((w''_0 + A''_0) + (w''_1 + A''_1) \cos ny) (\lambda + n^2 f^{iv}_1 \cos ny + 4n^2 f^{iv}_2 \cos 2ny)\right\}
$$

(14)

where $' = d/dx$.

Substituting the trigonometric identities

$$
\sin^2 \alpha = \frac{1}{2}(1 - \cos 2\alpha)
$$

(15)
\[ \cos^2 \alpha = \frac{1}{2}(1 + \cos 2\alpha) \] (16)

Into equation (11) yields

\[
L_{NL}(w, w + 2w) = -t^2 n^2 \left( \frac{w''}{2} (w_1 + 2A_1) + w_1 (w'_1 + 2A'_1) + \frac{w_1}{2}(w''_1 + 2A''_1) \right)
- t^2 n^2 (w'_0 (w_1 + 2A_1) + w_1 (w''_0 + 2A''_0)) \cos ny
- t^2 n^2 \left( \frac{w''}{2} (w_1 + 2A_1) - w_1 (w'_1 + 2A'_1) + \frac{w_1}{2}(w''_1 + 2A''_1) \right) \cos 2ny
\] (17)

Substituting the terms (9), (10) and (17) into the compatibility equation (5) yields

\[
\left\{ \begin{array}{l}
\left[ -t \frac{f^IV}{Rc} - t \frac{w''}{R} \right] \cos ny \\
+ \left[ \frac{t}{Rc} \left( \frac{1}{2} n^2 (w_1 + 2A_1) + w_1 (w'_1 + 2A'_1) \right) \right] \cos ny \\
+ \left[ \frac{t}{Rc} \left( 8n^2 f''_2 + 16n^4 f_2 \right) - t \frac{2}{R} \left( \frac{w'_1}{2} (w_1 + 2A_1) \right. \right. \\
- w'_1 (w'_1 + 2A'_1) + \frac{w_1}{2}(w''_1 + 2A''_1) \right] \cos 2ny \\
= 0
\end{array} \right.
\] (18)

Multiplying through by \( \frac{R^3 c}{t} \) and equating coefficients of like terms yields the following 3 equations

\[
f^IV_0 - c w''_0 = \frac{c}{2} n^2 \left[ A_1 w''_1 + (2A'_1 + w_1) w'_1 + w_1 w''_1 \right] = 0 \] (19)

\[
f^IV_1 - 2n^2 f''_1 + n^4 f_1 - c w''_1 = \frac{ct}{R} n^2 \left[ A'_0 w'_1 + (A'_1 + w_1) w'_0 \right] = 0 \] (20)

\[
f^IV_2 - 2(2n)^2 f''_2 + (2n)^4 f_2 - \frac{c t}{2 R} n^2 \left[ A_1 w''_1 + A'_1 w_1 \right. \right.
- (2A'_1 + w'_1) w'_1 + w_1 w''_1 \left. \right] = 0 \] (21)
Substituting the trigonometric identities (15) and (16) into equation (14) yields

\[
L_{NL}(F, \omega + \omega) = - \frac{E \pi^3}{c} \frac{n^2}{2} \left[ \frac{1}{2} \left( w_0 + A_1 \right) f_1'' + \left( w_1' + A_1' \right) f_1' \right] + \left( w_0'' + A_1'' \right) \frac{\lambda}{n^2} \cos \alpha
\]

\[
- \frac{E \pi^3}{c} \frac{n^2}{2} \left[ \left( w_0 + A_1 \right) f_1' + \left( w_0' + A_0' \right) f_1 + \left( w_1' + A_1' \right) \frac{\lambda}{n^2} \cos \alpha \right]
\]

\[
- \frac{E \pi^3}{c} \frac{n^2}{2} \left[ \left( w_1 + A_1 \right) f_1'' - \left( w_1' + A_1' \right) f_1 + 4(w_0'' + A_0') f_2 \right] + \frac{1}{2} \left( w_1' + A_1' \right) f_1 \cos \alpha
\]

\[
- \frac{E \pi^3}{c} \frac{n^2}{2} \left( w_1 + A_1 \right) f_2' + 4(w_1' + A_1') f_2 \cos \alpha \cos \alpha
\]

\[
- \frac{E \pi^3}{c} \frac{n^2}{2} \left( 4(w_1' + A_1') f_2 \sin \alpha \sin \alpha \sin \alpha \right)\]  

(22)

Substituting the terms (12), (13) and (22) into the equilibrium equation (6) yields

\[
\left\{ \frac{E \pi^4}{12R^4(1-v^2)} w_0'' + \frac{E \pi^2}{c \pi^2} \right. f_0' + \frac{E \pi^3}{c \pi^3} \frac{n^2}{2} \left[ \frac{1}{2} \left( w_0 + A_1 \right) f_1'' \right]
\]

\[
+ \left( w_1' + A_1' \right) f_1' + \left( w_0' + A_0' \right) \frac{\lambda}{n^2} + \frac{1}{2} \left( w_1' + A_1' \right) f_1 \right\}
\]

\[
+ \left\{ \frac{E \pi^4}{12R^4(1-v^2)} \left( w_0'' - 2n^2 w_0' + 4 w_0 \right) + \frac{E \pi^2}{c \pi^2} \right. f_0'' \right.
\]

\[
+ \frac{E \pi^3}{c \pi^3} \frac{n^2}{2} \left[ \left( w_0 + A_1 \right) f_0' + \left( w_0' + A_0' \right) f_1 \right]
\]

\[
+ \left( w_0'' + A_0' \right) \frac{\lambda}{n^2} \cos \alpha
\]

\[
+ \frac{E \pi^2}{c \pi^2} \right. f_1'' + \frac{E \pi^3}{c \pi^3} \frac{n^2}{2} \left[ \frac{1}{2} \left( w_1 + A_1 \right) f_1'' - \left( w_1' + A_1' \right) f_1' + 4(w_0'' + A_0') f_2 \right]
\]

\[
+ \frac{1}{2} \left( w_1' + A_1' \right) f_1 \right\} \cos \alpha \cos \alpha
\]

\[
+ \frac{E \pi^3}{c \pi^3} \frac{n^2}{2} \left( w_1 + A_1 \right) f_2' + 4(w_1' + A_1') f_2 \cos \alpha \cos \alpha
\]

\[
+ \frac{E \pi^3}{c \pi^3} \frac{n^2}{2} \left( 4(w_1' + A_1') f_2 \sin \alpha \sin \alpha \sin \alpha \right)\]  

(23)
The right-hand side of the equation is non-zero because the equilibrium equation is not necessarily satisfied exactly by the assumed form of the solution. Applying the Galerkin procedure, the following integrals are evaluated.

\[
\int 0 \leq y \leq 2\pi \varepsilon \, dy = 0 \tag{24}
\]

\[
\int 0 \leq y \leq 2\pi \varepsilon \cos (ny) \, dy = 0
\]

Using equation (23) to evaluate the integral (24) yields

\[
\omega_0^4 + 4c \frac{R^2}{t^2} f_0'' + 4c \frac{R}{t} \lambda (A_0'' + \omega_0'') + \frac{2cR}{t} \pi^2 \left( (A_1'' + \omega_1'') f_1 + (A_1 + \omega_1) f_1'' + 2(A_1 + \omega_1) f_1' \right) \tag{25}
\]

Again using equation (23) to evaluate the integral (25) yields

\[
\omega_1^4 - 2n^2 \omega_1'' + n^4 \omega_1 + 4c \frac{R^2}{t^2} f_1'' + 4c \frac{R}{t} \lambda (A_1'' + \omega_1'') + \frac{2cR}{t} \pi^2 \left[ 2(A_1'' + \omega_1'') f_2 + 4(A_1'' + \omega_1') f_2 + (A_1 + \omega_1) f_2' + 4(A_1 + \omega_1) f_1'' \right] = 0 \tag{26}
\]

In addition let us define "unit end shortening" as

\[
\varepsilon = -\frac{1}{2\pi RL} \int_0^L \left( u, x - q w, xx \right) \, dx \, dy \tag{27}
\]

where

\[
u_x = \varepsilon_x - \frac{1}{2} \left( w, x x + \frac{2w}{x} \right) \tag{28}
\]

\[
\varepsilon_x = \frac{1}{Et} \left( F, yy - vF, xx \right) \tag{29}
\]

Substituting these expressions into equation (27) and introducing the nondimensional parameters \( \bar{x}, \bar{y} \) and \( \bar{q} = \frac{4cR}{t^2} q \) yields
\[
\varepsilon = -\frac{1}{2\pi LR} \int_0^L \int_0^R \left( \frac{1}{E} (F_{yy} - \nu F_{xx}) - \frac{1}{2} \varepsilon \left( \frac{\varepsilon}{\varepsilon} + 2\frac{\varepsilon}{\varepsilon} \right) \right) dxdy
\]

Substituting for \( w \), \( \varepsilon \) and \( F \) from equations (1-3) into equation (30) yields

\[
\varepsilon = \frac{c}{2\pi L} \int_0^L \int_0^R \left\{ \frac{1}{c} \left[ \lambda + \nu f''_0 + (\mu^2 + \nu) f_1 \cos \gamma + (4n^2 + \nu) f_2 \cos 2\gamma \right] \right. \\
+ \frac{c}{R} \left\{ \frac{1}{2} w' \left( w_0^2 + 2A_0^2 \right) + \left( w_0 w_1 + w_0 A_1 + w_1 A_0 \right) \cos \gamma + \frac{1}{2} w_1 \left( w_1^2 + 2A_1^2 \right) \cos 2\gamma \right\} \\
+ \frac{c^2}{4R^2} \left( w_0'' + \gamma_1'' \right) \right\} dxdy
\]

Carrying out the \( \gamma \) integration yields

\[
\varepsilon = \frac{c}{L} \int_0^L \left[ \frac{1}{c} \left( \frac{\mu}{c} f''_0 + \frac{c}{2R} w_1 \left( w_0^2 + 2A_0^2 \right) + \frac{c}{4R^2} w_1 \left( w_1^2 + 2A_1^2 \right) + \frac{c^2}{4R^2} \gamma_1'' \right) \right] dx
\]

which is rewritten as

\[
\varepsilon = \frac{c}{R} \lambda + \frac{c^2 \nu}{L} \int_0^L f''_0 \ dx + \frac{c^2}{2LR} \int_0^L \left[ w_0'' w_0 + 2A_0^2 \right] dx + \frac{c^4}{4cLR^2} \left( w_0'' \right) \int_0^L dx
\]

(33)
APPENDIX B

The matrix equation (56)

For the elements of the solution vector \( x \) is written for \( i = 0, 1, \ldots, N \):
\[
x_i = (\delta w_0 \delta w_1 \delta f_1 \delta f_2 \delta w_0' \delta w_1' \delta f_1' \delta f_2')
\]
and further
\[
x_t = \delta \lambda
\]

For the element of the right hand side vector \( y \) is written for \( i = 0, 1, \ldots, N \):
\[
y_i(1) = h^2 n^2 c \frac{t}{R} \left[ (w_{11} + A_{11}) \frac{w_{11}'' + A_{01}'' w_{11}}{0_{11}} + h^2 c w_{11}'
\right] 
\]
\[- h^2 n^4 f_{11} + 2h^2 n^2 f_{11}'' - h^2 f_{11}^{IV}
\]
\[
y_i(2) = \frac{1}{2} h^2 n^2 c \frac{t}{R} \left[ A_{11}'' w_{11} - (w_{11}'' + 2A_{11}) w_{11}' + (w_{11} + A_{11}) w_{11}''
\right]
\[- 16h^2 n^4 f_{21} + 8 h^2 n^2 f_{21}'' - h^2 f_{21}^{IV}
\]
\[
y_i(3) = -4h^2 c \frac{R}{t} w_{01} - \frac{t}{R} h^2 w_{01}^{IV} - h^2 n^2 c (2A_{11} + w_{11}) w_{11}
\]
\[-2h^2 n^2 c \left[ (w_{11}' + A_{11}) f_{11} + 2 (w_{11}'' + A_{11}'') f_{11}' + (w_{11} + A_{11}) f_{11}''
\right]
\[-4h^2 c \lambda (A_{01}' + w_{01}')
\]
\[
y_i(4) = -4h^2 n^2 c (w_{11} + A_{11}) w_{01} - h^2 n^4 \frac{t}{R} w_{11}
\]
\[-h^2 n^2 c \frac{t}{R} (A_{11} + w_{11})(2A_{11} + w_{11}) w_{11} + 2h^2 n^2 \frac{t}{R} w_{11}'
\]
\[-h^2 \frac{t}{R} w_{11}^{IV} - 4h^2 n^2 c (w_{01}' + A_{01}') f_{11} + 4h^2 c \frac{R}{t} f_{11}^{IV}
\]
\[-8h^2 n^2 c (w_{11}' + A_{11}') f_{21} + 8h^2 n^2 c (w_{11} + A_{11}) f_{21}'
\]
\[-2h^2 n^2 c (w_{11} + A_{11}) f_{21}' - 4h^2 c \lambda (A_{11}' + w_{11}')
\]
\[
y_i(5) = -h^2 w_{01}'' + w_{01}''_{11} - 2w_{01} + w_{01}_{11}
\]
\[
y_i(6) = -h^2 w_{11}'' + w_{11}''_{11} - 2w_{11} + w_{11}_{11}
\]
\[ y_i(7) = -h^2 f''_{i1} + f'_{i1} + 2f_{i1} + f_{i1-1} \]
\[ y_i(8) = -h^2 f''_{i1} + f'_{i1} + 2f_{i1} + f_{i1-1} \]

and further

\[ y_t = \delta - \lambda - \frac{1}{2} \hbar c \frac{R}{L} w_{00} - \frac{1}{8} h n^2 c v \frac{t}{L} (w_{00} + 2A_{10}) w_{10} \]
\[ - \frac{1}{4} h c \frac{t}{L} (w_{00}' + 2A_{10}') w_{00}' - \frac{1}{8} h c \frac{t}{L} (w_{00}' + 2A_{10}') w_{00}' - \frac{1}{8} \eta q \frac{t^2}{L R} w_{00}' \]
\[ + \sum_{i=1}^{N-1} \left[ - h c v \frac{R}{L} w_{01} - \frac{1}{4} h n^2 c v \frac{t}{L} (w_{11} + 2A_{11}) w_{11} \right] \]
\[ - \frac{1}{2} h c \frac{t}{L} (w_{01}' + 2A_{11}') w_{01}' - \frac{1}{4} h c \frac{t}{L} (w_{11}' + 2A_{11}') w_{11}' - \frac{1}{4} \eta q \frac{t^2}{L R} w_{01}' \]
\[ - \frac{1}{2} h c v \frac{R}{L} w_{0N} - \frac{1}{8} h n^2 c v \frac{t}{L} (w_{1N} + 2A_{1N}) w_{1N} \]
\[ - \frac{1}{4} h c \frac{t}{L} (w_{0N}' + 2A_{1N}') w_{0N}' - \frac{1}{8} h c \frac{t}{L} (w_{1N}' + 2A_{1N}') w_{1N}' - \frac{1}{8} \eta q \frac{t^2}{L R} w_{0N}' \]

Hence all the coefficients of the coefficient matrix A are zero with the exception of:

for \( i = 0, 1, \ldots, N \)

\[ L_i(5,1) = -1 \]
\[ L_i(2,2) = -\frac{1}{2} h n^2 c \frac{t}{R} (w_{11}' + A_{11}') \]
\[ L_i(3,2) = -2h n^2 c f'_{11} \]
\[ L_i(4,2) = -4h n^2 c f'_{21} \]
\[ L_i(6,2) = -1 \]
\[ L_i(3,3) = -2h n^2 c (w_{11}' + A_{11}') \]
\[ L_i(7,3) = -1 \]
\[ L_1(4, 4) = -4h^2c \left( w_{1\,1}^{\prime\prime} + A_{1\,1}^{\prime\prime} \right) \]

\[ L_1(8, 4) = -1 \]

\[ L_1(3, 5) = t/R \]

\[ L_1(4, 6) = t/R \]

\[ L_1(1, 7) = 1 \]

\[ L_1(2, 8) = 1 \]

\[ D_1(3, 1) = 4h^2c^2 \frac{r}{t} \]

\[ D_1(4, 1) = 4h^2n^2c^2 \left( w_{1\,1}^{\prime} + A_{1\,1}^{\prime} \right) \]

\[ D_1(5, 1) = 2 \]

\[ D_1(1, 2) = -h^2n^2c \frac{t}{R} \left( w_{0\,1}^{\prime\prime} + A_{0\,1}^{\prime\prime} \right) \]

\[ D_1(2, 2) = -\frac{h^2n^2c}{2} \frac{t}{R} \left( w_{1\,1}^{\prime\prime} + A_{1\,1}^{\prime\prime} \right) \]

\[ D_1(3, 2) = 2h^2n^2c^2 \left( w_{1\,1}^{\prime} + A_{1\,1}^{\prime} \right) + 2h^2n^2c f_{1\,1}^{\prime\prime} \]

\[ D_1(4, 2) = 4h^2n^2c^2 w_{0\,1}^{\prime} + h^2n^4 c^2 \frac{t}{R} \left( 3w_{1\,1}^{2} + 6w_{1\,1}^{\prime} + 2A_{1\,1}^{2} \right) + 2h^2n^2c f_{2\,1}^{\prime\prime} + h^2n^4 \frac{t}{R} \]

\[ D_1(6, 2) = 2 \]

\[ D_1(1, 3) = h^2n^4 \]

\[ D_1(3, 3) = 2h^2n^2c \left( w_{1\,1}^{\prime\prime} + A_{1\,1}^{\prime\prime} \right) \]

\[ D_1(4, 3) = 4h^2n^2c \left( w_{0\,1}^{\prime\prime} + A_{0\,1}^{\prime\prime} \right) \]

\[ D_1(7, 3) = 2 \]
\[ D_i(2,4) = 16h^2n^4 \]
\[ D_i(4,4) = 8h^2n^2c \left( \psi_{11}^1 + \psi_{11}^{1'} \right) \]
\[ D_i(8,4) = 2 \]
\[ D_i(1,5) = -h^2n^2c \frac{R}{\varepsilon} \left( \psi_{11}^1 + \psi_{11}^{1'} \right) \]
\[ D_i(3,5) = 4h^2c\lambda - 2 \frac{R}{\varepsilon} \]
\[ D_i(4,5) = 4h^2n^2c f_{11} \]
\[ D_i(5,5) = h^2 \]
\[ D_i(1,6) = -h^2c \]
\[ D_i(2,6) = -\frac{1}{2} h^2n^2c \frac{R}{\varepsilon} \left( \psi_{11}^1 + \psi_{11}^{1'} \right) \]
\[ D_i(3,6) = 2h^2n^2c f_{11} \]
\[ D_i(4,6) = 4h^2c\lambda - 2 \frac{R}{\varepsilon} - 2h^2n^2c \frac{R}{\varepsilon} + 8h^2n^2c f_{21} \]
\[ D_i(6,6) = h^2 \]
\[ D_i(1,7) = -2h^2n^2-2 \]
\[ D_i(3,7) = 2h^2n^2c \left( \psi_{11}^1 + \psi_{11}^{1'} \right) \]
\[ D_i(4,7) = 4h^2c \frac{R}{\varepsilon} \]
\[ D_i(7,7) = h^2 \]
\[ D_i(2,8) = -8h^2n^2-2 \]
\[ D_i(4,8) = 2h^2n^2c \left( \psi_{11}^1 + \psi_{11}^{1'} \right) \]
\[ D_i(8,8) = h^2 \]
\[ U_i(5,1) = -1 \]
\[ U_1(2, 2) = \frac{1}{2} \hbar n^2 c (w_1 + A'_1) \]

\[ U_1(3, 2) = 2\hbar n^2 c f'_1 \]

\[ U_1(4, 2) = 4\hbar n^2 c f'_2 \]

\[ U_1(6, 2) = -1 \]

\[ U_1(3, 3) = 2\hbar n^2 c (w'_1 + A'_1) \]

\[ U_1(7, 3) = -1 \]

\[ U_1(4, 4) = 4\hbar n^2 c (w'_1 + A'_1) \]

\[ U_1(8, 4) = -1 \]

\[ U_1(3, 5) = \frac{t}{R} \]

\[ U_1(4, 6) = \frac{t}{R} \]

\[ U_1(1, 7) = 1 \]

\[ U_1(2, 8) = 1 \]

\[ r_1(3) = 4\hbar^2 c (A''_0 + w''_0) \]

\[ r_1(4) = 4\hbar^2 c (A''_1 + w''_1) \]

and further

\[ o_{-1}(1) = -\frac{1}{4} c \frac{E}{L} (w'_{0,0} + A'_{0,0}) \]

\[ o_{-1}(2) = -\frac{1}{8} c \frac{E}{L} (w'_{1,0} + A'_{1,0}) \]

\[ o_0(1) = \frac{1}{2} \hbar c v \frac{R}{L} - \frac{1}{2} c \frac{E}{L} (w'_{0,1} + A'_{0,1}) \]

\[ o_0(2) = \frac{1}{4} \hbar n^2 c v \frac{E}{L} (w'_{1,0} + A'_{1,0}) - \frac{1}{4} c \frac{E}{L} (w'_{1,1} + A'_{1,1}) \]
\[ o_0(5) = \frac{1}{8} h^2 \frac{e^2}{LR} \]

\[ o_1(1) = \hbar c v \frac{R}{L} + \frac{1}{4} c \frac{e}{L} [w_{0,2} + A_{0}^{'} - 2 (w_{2,2}^{'} + A_{2}^{''})] \]

\[ o_1(2) = \frac{1}{2} \hbar n^2 c v \frac{e}{L} (w_{1,1} + A_{1} + \frac{1}{8} c \frac{e}{L} [w_{1,0}^{'} + A_{1}^{'} - 2 (w_{1,2}^{'} + A_{2}^{''})] \]

\[ o_1(5) = \frac{1}{4} \hbar q \frac{e^2}{LR} \]

for \( i = 2, 3, \ldots, N-2 \)

\[ o_1(1) = \hbar c v \frac{R}{L} + \frac{1}{2} c \frac{e}{L} (w_{i-1}^{'} + A_{i-1}^{'} - w_{i}^{'} + A_{i}^{'} - A_{i+1}^{''}) \]

\[ o_1(2) = \frac{1}{2} \hbar n^2 c v \frac{e}{L} (w_{1,1} + A_{1} + \frac{1}{4} c \frac{e}{L} [w_{1,1-1}^{'} + A_{1-1}^{'} - w_{1,2}^{'} - A_{2}^{''}]) \]

\[ o_1(5) = \frac{1}{4} \hbar q \frac{e^2}{LR} \]

and further

\[ o_{N-1}(1) = \hbar c v \frac{R}{L} + \frac{1}{4} c \frac{e}{L} [2(w_{N-2}^{'} + A_{N-2}^{''}) - w_{N}^{'} + A_{N}^{''}] \]

\[ o_{N-1}(2) = \frac{1}{2} \hbar n^2 c v \frac{e}{L} (w_{N} + A_{N}) + \frac{1}{4} c \frac{e}{L} [2(w_{N-2}^{'} + A_{N-2}^{''}) - w_{N}^{'} - A_{N}^{''}] \]

\[ o_{N-1}(5) = \frac{1}{4} \hbar q \frac{e^2}{LR} \]

\[ o_{N}(1) = \frac{1}{2} \hbar c v \frac{R}{L} + \frac{1}{2} c \frac{e}{L} (w_{N-1}^{'} + A_{N-1}^{''}) \]

\[ o_{N}(2) = \frac{1}{4} \hbar n^2 c v \frac{e}{L} (w_{N} + A_{N}) + \frac{1}{4} c \frac{e}{L} (w_{N-1}^{'} + A_{N-1}^{''}) \]

\[ o_{N}(5) = \frac{1}{8} \hbar q \frac{e^2}{LR} \]

\[ o_{N+1}(1) = \frac{1}{4} c \frac{e}{L} (w_{N}^{'} + A_{N}^{''}) \]

\[ o_{N+1}(2) = \frac{1}{8} c \frac{e}{L} (w_{N}^{'} + A_{N}^{''}) \]

\[ d_t = 1 \]
By enforcement of the boundary conditions it is possible to omit 8 unknowns at $x = 0$ and 8 unknowns at $x = L/R$ out of the matrix equation (56). Rewriting, the vector unknowns $x_0$ and $x_N$ become for the boundary conditions

**SS-1:**
\[
\begin{align*}
X_0 &= (\delta w_0, \delta w_1, \delta w_0', \delta w_1', \delta f', 1, 1, 10, 2_0, 2_0) \\
X_N &= (\delta w_0, \delta w_1, \delta w_0', \delta w_1', \delta f', 1, 1, 10, 2_N, 2_N)
\end{align*}
\]

**SS-2:**
\[
\begin{align*}
X_0 &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_0) \\
X_N &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_N)
\end{align*}
\]

**SS-3:**
\[
\begin{align*}
X_0 &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_0) \\
X_N &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_0)
\end{align*}
\]

**SS-4:**
\[
\begin{align*}
X_0 &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_0) \\
X_N &= (\delta w_0, \delta w_1, \delta f_1, 0, \delta w_0', \delta w_1', \delta f', 1, 1_0, 2_0)
\end{align*}
\]

**C-1:**
\[
\begin{align*}
X_0 &= (\delta w_0', \delta w_1', \delta f', 1, 1, 2_1, 2_1, 0, 1_0, 2_0) \\
X_N &= (\delta w_0', \delta w_1', \delta f', 1, 1, 2_1, 2_1, 0, 1_0, 2_0)
\end{align*}
\]

**C-2:**
\[
\begin{align*}
X_0 &= (\delta w_0', \delta w_1', \delta f, 0, 0, 0, 0, 0, 10, 2_0) \\
X_N &= (\delta w_0', \delta w_1', \delta f, 0, 0, 0, 0, 0, 10, 2_0)
\end{align*}
\]

**C-3:**
\[
\begin{align*}
X_0 &= (\delta f_1, 0, \delta f_2, 0, 0, 0, 0, 0, 10, 2_0) \\
X_N &= (\delta f_1, 0, \delta f_2, 0, 0, 0, 0, 0, 10, 2_0)
\end{align*}
\]
\begin{align*}
\text{C-4:} & \quad x_0 = (\delta f_{10}, \delta f_{20}, \delta f_{10}, \delta f_{20}, \delta \omega_{10}, \delta \omega_{10}, \delta \omega_{10}, \delta \omega_{10}) \\
& \quad x_N = (\delta f_{N+1}, \delta f_{N+1}, \delta f_{N+1}, \delta f_{N+1}, \delta \omega_{N+1}, \delta \omega_{N+1}, \delta \omega_{N+1}, \delta \omega_{N+1}) \\
\text{Symm.:} & \quad x_0 = (\delta \omega_{10}, \delta \omega_{10}, \delta f_{20}, \delta f_{20}, \delta \omega_{10}, \delta \omega_{10}, \delta f_{20}, \delta f_{20}) \\
& \quad x_N = (\delta \omega_{N1}, \delta \omega_{N1}, \delta f_{2N}, \delta f_{2N}, \delta \omega_{N1}, \delta \omega_{N1}, \delta f_{2N}, \delta f_{2N})
\end{align*}

Doing this, we may rewrite the system (56) in the system (67).
APPENDIX C

The computerprogram FREIA.

The in this report presented efficient approximate solution method for predicting the buckling load of axially compressed imperfect isotropic shells was coded into the computerprogram FREIA. The use of this program makes it possible to investigate how the axial load level at the limit point of the nonlinear prebuckling equilibrium states is affected by the following factors: the choice of inplane boundary conditions, the prebuckling growth caused by radial edge restraint, the location of the load application point, the orientation and shape of the axisymmetric and asymmetric imperfection components, and the finite length of the shell.

The program FREIA is written in FORTRAN IV; its various tasks of computation are divided in a main program and several subroutines. A brief description of the main program, its subroutines and its in-and output is given in the following paragraphs. The various types of variables are defined by the ("IBM")FORTRAN implicit "first letter" type convention statement: IMPLICIT REAL*8 (A-H, O-Z), INTEGER*4 (I-N), being stored in 8 and 4 bytes respectively.

The main program

Purpose

To control the program FREIA as an entire integrated unit, to check the convergence and accuracy of the solution, and to take care of the in- and output.

Subroutines and function subprograms required

FDAT1, FHV, FAYLD, FDAT2D, ARBPOT, FAYNL, CDAT2D, IMPERF.

Declaration of the arrays

Using the DIMENSION statement, all arrays are declared as one dimensional arrays. Before running the program one next ascertain that,

array A has at least (N+1).208 elements,
array DAT1 has at least (N+1).8 elements,
array DAT2 has at least (N+3).8 elements, and the
array Y has at least (N+1).8 elements

The declaration of the remaining arrays must remain unchanged.
Input (using DD-number 5)
The used formats are fixed and can occupy 72 columns:
integer numbers are read with the format statement I5, real numbers are read
with the format statement D15.8, and cards with only alfanumerical data are
read with the format statement 9A8.
A description of the input cards is given on page 53. The last input card
can be repeated as often as wished. The input parameters have the following
meaning:
IDDIN  =  DD-number of the dataset from which the initial data are read; if
         IDDIN = 0 the initial data are computed from linear theory.
IDDOUT = DD-number of the dataset in which after the last iteration the
         solution is stored. If IDDOUT = 0, the solution is not stored.
IWRITE = number for selection of the output
         if IWRITE = 0: the corrected solution is only written out after
         the last iteration.
         if IWRITE = 1: after each iteration the corrected solution is
         written out.
NI     =  N
ITEMAX = maximum number of iterations allowed
NEPSI  =  number that defines the relative precision EPSI which the solution
         of the recurrence relations in Arboicz' modification of Potters'
         method has to meet as EPSI = 0.1^NEPSI
NPREC  =  number that defines the relative precision PREC which all elements
         of the correction vector have to meet: PREC = 0.1^NPREC
XLDIR  =  L/R
RDIT   =  R/t
XNU    =  ν
IBC0   =  number for the selection of the boundary condition at x = 0;
         if IBC0 = 1: boundary condition SS-1 at x = 0
         if IBC0 = 2: boundary condition SS-2 at x = 0
         if IBC0 = 3: boundary condition SS-3 at x = 0
         if IBC0 = 4: boundary condition SS-4 at x = 0
         if IBC0 = 5: boundary condition C-1 at x = 0
         if IBC0 = 6: boundary condition C-2 at x = 0
         if IBC0 = 7: boundary condition C-3 at x = 0
         if IBC0 = 8: boundary condition C-4 at x = 0
         if IBC0 = 9: boundary condition sym. at x = 0
\[ IBCNI = \text{number for selection of the boundary condition at } \bar{x} = L/R; \]
\[ \quad \text{if } IBCNI = 1: \text{boundary condition SS-1 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 2: \text{boundary condition SS-2 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 3: \text{boundary condition SS-3 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 4: \text{boundary condition SS-4 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 5: \text{boundary condition C-1 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 6: \text{boundary condition C-2 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 7: \text{boundary condition C-3 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 8: \text{boundary condition C-4 at } \bar{x} = L/R \]
\[ \quad \text{if } IBCNI = 9: \text{boundary condition sym. at } \bar{x} = L/R \]

\[ Q = q \]

\[ NCW = n \]

\[ IPARM = \text{number for selection of the incremental parameter:} \]
\[ \quad \text{if } IPARM = 1: \delta \text{ is the incremental parameter} \]
\[ \quad \text{if } IPARM = 2: \delta_{NL} \text{ is the incremental parameter} \]

\[ PARM = \text{value of the incremental parameter} \]

In addition one has to write the subroutine IMPERF in order to define the imperfection data \( A_0(\bar{x}) \) and \( A_1(\bar{x}) \)

Output (using DD-number 6)

Successively are written out: the input, the computed imperfection data, the initial values and the solution. The solution has one of the following five levels:

Solution level 0: all solution terms which have a corresponding element in the correction vector \( \bar{x} = A^{-1} \bar{y} \) meet the relative precision \( 0.1^{NPREC} \)

Solution level 1: the computation is stopped because ITEMAX is reached before the solution meet any other solution level.

Solution level 2: the computation is stopped because the norm of the correction vector \( \bar{x} = A^{-1} \bar{y} \) does not converge anymore, yet the norm of that part of the solution term vector which corresponds with the correction vector \( \bar{x} = A^{-1} \bar{y} \) meets the relative precision \( 0.1^{NPREC} \)

Solution level 3: the computation is stopped because the norm of the correction vector \( \bar{x} = A^{-1} \bar{y} \) does not converge any more, even the norm of that part of the solution term vector which corresponds with the correction vector \( \bar{x} = A^{-1} \bar{y} \) does not meet the relative precision \( 0.1^{NPREC} \).
Solution level 4: the computation is stopped because the initial values are bad: at the first iteration the norm of the correction vector $x = A^{-1}y$ is larger than the norm of the corresponding part of the initial value vector.

In addition GPREC is written out being the ratio of the norm of the last computed correction vector $x = A^{-1}y$ over the norm of the corresponding part of the uncorrected solution term vector.
Subroutine FDAT1

Purpose
To compute the imperfection data matrix DAT1, and the imperfection data
variables Al0 and AlNI

Usage
call fdat1 (DAT1, XLDIR, Al0, AlNI, NI)

Description of the parameters
DAT1 = output matrix on return containing imperfection data. The imperfec-
tion data are stored columnwise in the following form:
\[ \text{DAT1}_i = (x_i, A_0^i, A_1^i, A'_0^i, A'_1^i, 2A''_1^i) \] for i = 0, 1, ..., N
XLDIR = L/R, input variable
Al0 = \( A_0 \), output variable
AlNI = \( A'_1 \), output variable
NI = N, input variable

Remarks
None

Subroutines and function subprograms required
IMPERF

Method
Using subroutine IMPERF, \( A_0 \) and \( A_1 \) are computed at the grid points \( I = -1, ..., N+1 \). The first and second derivatives of \( A_0 \) and \( A_1 \) are calculated using the central difference formulae.
Subroutine FHVE

Purpose

To compute the vectors HVEL, HVED and HVBC

Usage

CALL FHVE (HVEL, HVED, HVBC, XLDIR, RDIT, XNU, Q, AIØ AINI, NCW, NI)

Description of the parameters

HVEL = output vector on return containing the resultants of some expressions needed to set up that part of the iterative solution scheme that does not involve the concept of "end shortening" nor the enforcement of the boundary conditions.

HVEL (1) = t/R
HVEL (2) = 2h
HVEL (3) = 2hn^2c
HVEL (4) = 4hn^2c
HVEL (5) = \frac{1}{2}hn^2c t/R
HVEL (6) = 2t/R
HVEL (7) = h^2
HVEL (8) = h^2 t/R
HVEL (9) = h^2c
HVEL (10) = 4h^2c
HVEL (11) = 4h^2c R/t
HVEL (12) = 4h^2c^2 R/t
HVEL (13) = 2hn^2
HVEL (14) = 8h^2 n^2
HVEL (15) = 2h^2 n^2 t/R
HVEL (16) = 2h n^2 + 2
HVEL (17) = 8h^2 n^2 + 2
HVEL (18) = 2h^2n^2c
HVEL (19) = 4h^2n^2c
HVEL (20) = 8h^2 n^2 c
HVEL (21) = h^2 n^2 c t/R
HVEL (22) = \frac{1}{2} h^2 n^2 c t/R
HVEL (23) = h^2 n^2 c^2
HVEL (24) = 2h^2n^2 c^2
HVEL (25) = 4h^2 n^2 c^2
HVEL (26) = h^2 n^4
HVEL (27) = 16\pi^2n^4
HVEL (28) = \pi^4t/R
HVEL (29) = \pi^2n^2c^2t/R
HVEL (30) = reserved for further calculation
HVEL (31) = reserved for further calculation
HVEL (32) = reserved for further calculation

**HVED** = Output vector on return containing the resultants of some expressions needed to bring in the concept of "end shortening" in the iterative solution scheme

\[
\begin{align*}
\text{HVED (1)} &= \frac{1}{4}h(q - \frac{t}{L})^2 \\
\text{HVED (2)} &= \frac{1}{8}h(q - \frac{t}{L})tL \\
\text{HVED (3)} &= \frac{1}{2}ct/L \\
\text{HVED (4)} &= \frac{1}{4}ct/L \\
\text{HVED (5)} &= \frac{1}{8}ct/L \\
\text{HVED (6)} &= \frac{1}{2}ht/L \\
\text{HVED (7)} &= \frac{1}{4}ht/L \\
\text{HVED (8)} &= \frac{1}{8}ht/L \\
\text{HVED (9)} &= hcr R/L \\
\text{HVED (10)} &= \frac{1}{2}hcv R/L \\
\text{HVED (11)} &= \frac{1}{2}\pi^2n^2cv t/L \\
\text{HVED (12)} &= \frac{1}{4}\pi^2n^2cv t/L \\
\text{HVED (13)} &= \frac{1}{8}\pi^2n^2cv t/L
\end{align*}
\]

**HVEBC** output vector on return containing the resultants of some expressions needed to make it possible to bring in the enforcement of all the boundary conditions that are free to be chosen, in the iterative solution scheme

\[
\begin{align*}
\text{HVEBC (1)} &= q \\
\text{HVEBC (2)} &= c \\
\text{HVEBC (3)} &= v/c \\
\text{HVEBC (4)} &= n^2c \\
\text{HVEBC (5)} &= 4n^2c \\
\text{HVEBC (6)} &= n^2(v+c)
\end{align*}
\]
HVEBC (7) = \(4n^2(v+2)\)
HVEBC (8) = \(n^2cA_{1}t/R\)
HVEBC (9) = \(\frac{1}{2}n^2cA_{1}^{0}t/R\)
HVEBC (10) = \(n^2cA_{1}^{0}t/R\)
HVEBC (11) = \(\frac{1}{2}n^2cA_{1}^{N}t/R\)

XLDIR = L/R, input variable
RDIT = R/t, input variable
XNU = v, input variable
Q = q, input variable
A1Ø = A_{1}^{0}, input variable
ALNI = A_{1}^{0}, input variable
NI = N, input variable

Remarks
None

Method

The elements of the vectors HVEL, HVED and HVEBC are computed as the resultants of some expressions.
Subroutine FAYLD

Purpose

To set up the bordered block tridiagonal matrix equation \( Ax = y \) according to the linear theory, with enforcement of the boundary conditions.

Usage

CALL FAYLD (A, DATI, Y, HVEL, HVED, HVEBC, DT, YT, PARM, IPARM, NIP1, IBC0, IBCNI)

Description of the parameters

A = output matrix on return containing the bordered block tridiagonal matrix \( A \) with the exception of the element \( d_L \). The matrix is filled columnwise in the following compressed form:

\[ L_1, D_1, U_1, L_2, D_2, \ldots, L_N \]

DATI = input matrix containing imperfection data

Y = output vector on return containing the right hand side vector \( y \), with the exception of \( y_L \).

HVEL = input vector containing the resultants of some expressions.

HVED = input vector containing the resultants of some expressions.

HVEBC = input vector containing the resultants of some expressions.

DT = \( d_L \), output variable

YT = \( y_L \), output variable

PARM = incremental parameter, input variable

IPARM = input variable for selection of the incremental parameter

NIP1 = \( N+1 \), input variable

IBC0 = input variable for selection of the boundary condition at \( x = 0 \)

IBCNI = input variable for selection of the boundary condition at \( x = L/R \)

Remarks

This subroutine may be used in order to obtain appropriate initial values from the linear theory.

Subroutines and function subprograms required

None

Method

Block equation by block equation the matrix \( A \) is filled.

The vector \( y \) with the exception of \( y_L \) is set equal to zero.
Subroutine FDAT2D

Purpose
To store with enforcement of the boundary conditions the solution computed
from linear theory into the matrix DAT2 and the variable XLABDA.

Usage
CALL FDAT2D (DAT2, Y, HVEBC, YT, XLABDA, NIPI, IBCO, IBCNI)

Description of the parameters

DAT2  = output matrix, on return containing the solution computed from
       linear theory, with the exception of \( \lambda \). The solution is stored
       with enforcement of the boundary conditions, columnwise in the
       following form:

\[
\begin{bmatrix}
    w_0 & w_1 & f_1 & f_2 & w_0' & w_1' & f_1' & f_2'
\end{bmatrix}
\]

for \( i = -1, 0, \ldots, N+1 \)

Y  = inputvector containing the solution vector \( \mathbf{x} = A^{-1}Y \) computed from
linear theory, with the exception of \( x_L = \lambda \).

HVEBC  = inputvector containing the resultants of some expressions.

YT  = \( x_L = \lambda \), input variable

XLABDA  = \( \lambda = x_L \), output variable

IBCO  = input variable for selection of the boundary condition at \( x = 0 \)

IBCNI  = input variable for selection of the boundary condition at \( x = L/R \)

Remarks
None

Subroutines and subprograms required
None

Method
Element by element the solution vector \( \mathbf{x} \) is stored into the matrix DAT2 and
the variable XLABDA with enforcement of the boundary conditions.
Subroutine ARBPOT

Purpose

To solve a system of simultaneous equations \( Ax = y \) with a block tridiagonal coefficient matrix being bordered at the right and below by an additional column and row, respectively.

Usage

CALL ARBPOT (A, Y, HMFD, HMDU, HVP, HVORN, HVCONB, HVYINB, YT, DT, EPSI, NB, N, NWAL, NWA2, IER, IB)

Description of the parameters

- **A** = given bordered block tridiagonal coefficient matrix from NB simultaneous block equations and one additional equation, with the exception of the last additional diagonal element. Each block equation exists of N simultaneous equations. The coefficients are stored columnwise in compressed form in NB successive blocks, each containing a N by N lower diagonal block, a N by N diagonal block, a N by N upper diagonal block, a N dimensional column from the right, and a N dimensional row from below respectively. The coefficients of the first lower diagonal block and the last upper diagonal block have no meaning at all. The matrix A is destroyed in computation.

- **Y** = given right hand side vector with the exception of the last element. On return Y contains the solution of the equation with the exception of the last element.

- **HMFD** = N by N matrix needed as workspace

- **HMDU** = N by N matrix needed as workspace

- **HVP** = N dimensional vector needed as workspace

- **HVORN** = N dimensional vector needed as workspace

- **HVCONB** = N dimensional vector needed as workspace

- **HVYINB** = N dimensional vector needed as workspace

- **YT** = given last element of the right hand side vector on return containing the last element of the solution vector.

- **DT** = given last additional diagonal coefficient of the bordered block tridiagonal matrix A, being destroyed in computation.

- **EPSI** = input 'constant for the relative precision which the solution of the recurrence relations have to meet.

- **NB** = number of block equations, N must be larger than 1.
N = number of equations per block, N must be larger than 1.

NWA1 = resulting warning indicator on return containing the number of times that only the norm of a solution vector of the recurrence relations meets the relative precision EPSI.

NWA2 = resulting warning indicator on return containing the number of times that even the norm of a solution vector of the recurrence relations does not meet the relative precision EPSI.

IER = resulting error indicator coded as follows:
IER = 0: no error; each component of the solution vectors of the recurrence relations meets the relative precision EPSI.
IER = 1: no result because a solution vector of the recurrence relations at block equation IB has no meaning at all; the recurrence relations at block equation IB are ill-conditioned.
IER = 2: no result because:
    if IB = 0 : input error, NB<2 or N<2
    if 1<IB<NB : the recurrence relations at block equation IB are singular
    if IB = NB+1: the recurrence relation for solving YT is singular

IB = resulting indicator, on return containing the number of block equations until what the solution scheme did proceed.

Remarks
This subroutine is especially fitted for solving central difference equations. On a computer with 8-bytes memory locations, EPSI should be set to about $1.10^{-7}$ in order to solve the recurrence relations with roughly single precision accuracy.

Subroutines and function subprograms required
None.

Method
Arbocz' modification of Potter's method. The recurrence relations are solved using Crout's method with equilibrated partial pivoting followed by iterative refinement.
Subroutine FAYNLD

Purpose

To set up the bordered block tridiagonal matrix equation \( Ax = y \) according to the nonlinear theory, with enforcement of the boundary conditions.

Usage

CALL FAYNLD (A, DAT1, DAT2, Y, HVEL, HVED, HVEBC, DT, YT, XLABDA, PARM, IPARM, NIPI, IBC0, IBCNI)

Description of the parameters

A = output matrix on return containing the bordered block tridiagonal matrix \( A \) with the exception of the element \( d_L \). The matrix is filled columnwise in the following compressed form:
\[
L_i^0, D_i^1, U_i^1, V_i^1, O_i^1 \quad \text{for } i = 0, 1, \ldots, N
\]

DAT1 = input matrix containing imperfection data

DAT2 = input matrix containing the uncorrected solution with exception of \( \lambda \)

Y = output vector on return containing the right hand side vector \( y \) with the exception of \( y_L \)

HVEL = input vector containing the resultants of some expressions.

HVED = input vector containing the resultants of some expressions.

HVEBC = input vector containing the resultants of some expressions.

DT = \( d_L \), output variable

YT = \( y_L \), output variable

XLABDA = \( \lambda \), uncorrected input variable

PARM = incremental parameter, input variable

IPARM = input variable for selection of the incremental parameter

NIPI = NI + 1, input variable

IBC0 = input variable for selection of the boundary condition at \( x = 0 \)

IBCNI = input variable for selection of the boundary condition at \( x = L/R \).

Remarks

Starting with approximate initial values, this subroutine may be used in an iterative scheme in order to obtain the solution from the nonlinear theory.

Subroutines and subprogram required

None.
Method

Block equation by block equation the matrix \( A \) and the vector \( Y \) are filled. The first and fourth derivative variables are computed from the non-derivative and second derivative variables using the central difference formulæ.
Subroutine CDAT2D

Purpose

To add with enforcement of the boundary conditions, the correction vector \( \bar{x} \) computed from nonlinear theory to the uncorrected solution stored in the vector DAT2 and the variable XLABDA, and to check the accuracy of the solution.

Usage

CALL CDAT2D (DAT2, Y, HVEBC, YT, XLABDA, PREC, XNORMY, XNPDT2, NIP1, IBC0, IBCNI, ICC)

Description of the parameters

- **DAT2** = in- and output matrix containing the uncorrected solution with the exception of \( \lambda \). On return the correction vector \( \bar{x} \) with the exception of \( x_\lambda = \delta \lambda \) is added to the uncorrected solution with enforcement of the boundary conditions. The solution is stored columnwise in the following form:
  \[
  \text{DAT2}_i = (w_0, w_1, f_1, w_2, w', w'', f_1', f'') \quad \text{for } i = -1, 0, \ldots, N+1
  \]

- **Y** = input vector containing the correction vector \( \bar{x} \) with the exception of \( x_\lambda = \delta \lambda \).

- **HVEBC** = input vector containing the resultants of some expressions.

- **YT** = \( x_\lambda = \delta \lambda \), input variable

- **XLABDA** = in- and output variable containing the uncorrected solution for \( \lambda \). On return the correction term \( x_\lambda = \delta \lambda \) is added to the uncorrected solution.

- **PREC** = input variable indicating the relative precision which all terms of the correction vector have to meet.

- **XNORMY** = output variable, on return containing the norm of the correction vector \( \bar{x} = A^{-1}Y \).

- **XNPDT2** = output variable, on return containing the norm of that part of the uncorrected solution term vector which corresponds with the correction vector \( \bar{x} = A^{-1}Y \).

- **NIP1** = \( N+1 \), input variable

- **IBC0** = input variable for selection of the boundary conditions at \( \bar{x} = 0 \).

- **IBCNI** = input variable for selection of the boundary conditions at \( \bar{x} = L/R \).
ICC = output variable coded as follows:

ICC = 0; all terms of the correction vector meet the relative precision PREC.

ICC = 1; not all terms of the correction vector meet the relative precision PREC.

Remarks
None.

Subroutines and function subprograms required
None

Method
Element by element the correction vector \( \mathbf{x} \) is added to the matrix DAT2 and the variable XLABDA with enforcement of the boundary conditions and a check for the relative precision.
Subroutine IMPERF

Purpose
To compute the imperfection data variables $A_0$ and $A_1$ for given $\bar{x}$.

Usage
CALL IMPERF (X, A0, A1)

Description of the parameters
\begin{align*}
X &= \bar{x}, \text{ input variable} \\
A0 &= A_0(\bar{x}), \text{ output variable} \\
A1 &= A_1(\bar{x}), \text{ output variable}
\end{align*}

Remarks
The subroutine IMPERF has to be written by the user of the program FREIA in order to define the imperfections.

Subroutines and function subprograms required
None.

Method
To be chosen by the user.
Table 1: Numerical results of the test runs

<table>
<thead>
<tr>
<th>δ</th>
<th>δ_{NR}</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60000</td>
<td>-0.000879</td>
<td>0.60088</td>
</tr>
<tr>
<td>0.62500</td>
<td>-0.000681</td>
<td>0.62568</td>
</tr>
<tr>
<td>0.65000</td>
<td>-0.000264</td>
<td>0.65026</td>
</tr>
<tr>
<td>0.66000</td>
<td>0.000103</td>
<td>0.65989</td>
</tr>
<tr>
<td>0.66400</td>
<td>0.000350</td>
<td>0.66365</td>
</tr>
<tr>
<td>0.66600</td>
<td>0.000521</td>
<td>0.66547</td>
</tr>
<tr>
<td>0.66800</td>
<td>0.000753</td>
<td>0.66724</td>
</tr>
<tr>
<td>0.66900</td>
<td>0.000910</td>
<td>0.66809</td>
</tr>
<tr>
<td>0.67000</td>
<td>0.001122</td>
<td>0.66887</td>
</tr>
<tr>
<td>0.67100</td>
<td>0.001473</td>
<td>0.66952</td>
</tr>
<tr>
<td>0.67120</td>
<td>0.001600</td>
<td>0.66960</td>
</tr>
<tr>
<td>0.67131</td>
<td>0.001700</td>
<td>0.66961</td>
</tr>
<tr>
<td>0.67138</td>
<td>0.001800</td>
<td>0.66958</td>
</tr>
<tr>
<td>0.67142</td>
<td>0.001900</td>
<td>0.66952</td>
</tr>
<tr>
<td>0.67106</td>
<td>0.002500</td>
<td>0.66856</td>
</tr>
<tr>
<td>0.66895</td>
<td>0.003500</td>
<td>0.66545</td>
</tr>
<tr>
<td>0.65326</td>
<td>0.007500</td>
<td>0.64576</td>
</tr>
<tr>
<td>0.64137</td>
<td>0.010000</td>
<td>0.63137</td>
</tr>
</tbody>
</table>