

## SYMBOLIC MANIPULATION IN BUCKLING AND POSTBUCKLING ANALYSIS

N. RIZZI

Istituto di Scienza delle Costruzioni, Facoltà di Ingegneria, Università La Sapienza, 00184  
Roma, Italy

and

A. TATONE

Istituto di Scienza delle Costruzioni, Facoltà di Ingegneria, Università di L'Aquila, 67100  
L'Aquila, Italy

(Received 13 February 1984; in revised form 11 June 1984)

**Abstract**—A perturbation procedure for the buckling and postbuckling analysis of elastic structures is shown to be well suited to be implemented as an automatic symbolic manipulation procedure. The postbuckling analysis of a circular arch is considered as an example, and the asymptotic description of the bifurcated equilibrium path is given. The main purposes of the automatic procedure are to generate the representation of the Fréchet operator for the strain field and to perform integration by parts. This allows the manipulation of correct expressions of the basic relationships, as the strain-displacement one, without introducing any simplifying assumption or restriction. The perturbation equations are automatically generated and a solution procedure leads to parametric expressions for the coefficients of the asymptotic expansion of the bifurcated path. The symbolic manipulation system used is REDUCE.

### 1. INTRODUCTION

The asymptotic buckling and post-buckling analysis of elastic structures is a well established procedure [1]. It rests upon the perturbation analysis of three groups of differential equations: equilibrium, compatibility and constitutive equations [2]. Carrying out the analysis keeping these groups separate from each other has proved to be useful not only for the sake of clarity but also for imposing undeformability constraints or introducing nonlinear constitutive equations.

The starting point for a bifurcation analysis is a "fundamental" equilibrium path along which one is interested in looking for bifurcating paths. Unfortunately, even if the fundamental path is very simple, the representation of the Fréchet derivatives of the strain field along it involves manipulations of expressions which take a lot of time and hard work to be performed. When the first-order equations have been derived, even more work is needed, in the substitution of the compatibility and constitutive relations into the equilibrium equations, and so on. As a result, one has to check expressions again and again and is never sure about them.

A symbolic automatic manipulation allows to encompass these problems making it possible to derive error free expressions by defining simple procedures.

The objective of this work is to check the usefulness of the automatic symbolic manipulation in the asymptotic bifurcation analysis of elastic beams. In particular we will show an application of the algebraic manipulation system REDUCE [3, 4] to the: (a) generation of formal perturbation equa-

tions up to the desired order; (b) construction of a procedure, essentially interactive, which is a helpful tool in problem solving.

A specific problem will be considered in order to give an example of how the introduced procedures can be used. Due to the fact that for this problem it is possible to construct an exact solution to the perturbation equations, we will follow this way. It should be stressed, however, that the procedures used retain a more general value also in view of a numerical approach, which will not be pursued here.

A first assessment of the use of the Algebraic Manipulation Systems in structural mechanics can be found in [5], while an application to the solution of perturbation problems has recently been reported in [6].

### 2. ASYMPTOTIC BIFURCATION ANALYSIS

Let us consider a hyperelastic system acted upon by external conservative loads and let

$$\Pi(u, \epsilon; \lambda) := \Phi(\epsilon) - p(\lambda) f(u) \quad (1)$$

be the total potential energy function. Further let

$$\begin{aligned} \epsilon &= e(u), \\ \sigma &= s(\epsilon) := \Phi'(\epsilon), \end{aligned} \quad (2)$$

be the strain and stress field, respectively. The equilibrium equation is obtained by imposing (1) to be stationary, and reads

$$\Phi'(\epsilon) e'(u) \delta u - p(\lambda) f'(u) \delta u = 0 \quad (3)$$

or equivalently

$$\begin{aligned} \sigma e'(u) \delta u - p(\lambda) f'(u) \delta u &= 0, \\ \sigma &= s(\epsilon), \\ \epsilon &= e(u), \end{aligned} \tag{4}$$

where a prime denotes differentiation of each function with respect to its own argument.

Two regular branches are supposed to be solutions of eqns (4). We assume the first to be known in the form

$$\begin{aligned} u &= u^f(\lambda), \\ \epsilon &= \epsilon^f(\lambda), \\ \sigma &= \sigma^f(\lambda), \end{aligned} \tag{5}$$

and we are looking for a second one

$$\begin{aligned} u &= u^b(t) = u^f[\lambda(t)] + v(t), \\ \epsilon &= \epsilon^b(t) = \epsilon^f[\lambda(t)] + \gamma(t), \\ \sigma &= \sigma^b(t) = \sigma^f[\lambda(t)] + \tau(t), \end{aligned} \tag{6}$$

which is supposed to bifurcate from the first one at a point label led by  $t = 0$ .

For an introduction to these concepts and further details on the methods and notations used, the reader is referred to the essay of Budiansky [4]. Here we intend to point out that the aim of the perturbation method is to derive asymptotic expressions for the unknown ‘‘bifurcated’’ equilibrium path by using eqns (4) and (6). Perturbation equations up to the third order are written as follows

$$\begin{aligned} \sigma e'' \delta u + \tau e' \delta u - p f'' \delta u &= 0, \\ \dot{\tau} &= s' \dot{\gamma}, \\ \dot{\gamma} &= e' \dot{v}, \end{aligned} \tag{7}$$

$$\begin{aligned} (\sigma e'' \dot{v} + \tau e' - p f'' \dot{v}) \delta u &= \\ &= -(2\dot{\lambda} \hat{\sigma} e'' \dot{v} + 2\dot{\lambda} \hat{\tau} e' \dot{u} + 2\tau e'' \dot{v} + 2\dot{\lambda} \sigma e''' \dot{u} \\ &+ \sigma e''' \dot{v}^2 - 2\dot{\lambda} \hat{p} f'' \dot{v} - 2\dot{\lambda} p f''' \dot{u} \dot{v} - p f'' \dot{v}^2) \delta u, \\ \dot{\tau} &= s' \dot{\gamma} + 2\dot{\lambda} s'' \hat{\epsilon} \dot{\gamma} + s'' \dot{\gamma}^2, \\ \dot{\gamma} &= e' \dot{v} + 2\dot{\lambda} e'' \dot{u} \dot{v} + e'' \dot{v}^2, \end{aligned} \tag{8}$$

$$\begin{aligned} (\sigma e''' \ddot{v} + \tau e'' - p f''' \ddot{v}) \delta u &= \\ &= -(3\dot{\lambda}^2 \hat{\sigma} e'' \dot{v} + 3\dot{\lambda} \hat{\sigma} e''' \dot{v} + 3\hat{\tau} e'' \dot{v} + 3\dot{\lambda} \hat{\tau} e'' \dot{u} \\ &+ 3\dot{\lambda} \hat{\sigma} e'' \dot{v}^2 + 6\dot{\lambda}^2 \hat{\sigma} e''' \dot{u} \dot{v} + 3\hat{\tau} e''' \dot{v}^2 \\ &+ 6\dot{\lambda} \hat{\tau} e'' \dot{u} \dot{v} + 3\dot{\lambda}^2 \hat{\tau} e''' \dot{u}^2 + 3\dot{\lambda} \hat{\sigma} e'' \dot{v} + 3\hat{\tau} e'' \dot{v} \\ &+ 3\dot{\lambda}^2 \hat{\tau} e'' \dot{u} \dot{v} + 3\dot{\lambda} \hat{\tau} e'' \dot{u} + 3\dot{\lambda}^2 \sigma e''' \dot{u}^2 \dot{v} \\ &+ 3\dot{\lambda} \sigma e''' \dot{u} \dot{v}^2 + \sigma e''' \dot{v}^3 + 3\dot{\lambda} \sigma e''' \dot{u} \dot{v} \\ &+ 3\dot{\lambda}^2 \sigma e''' \dot{u} \dot{v} + 3\dot{\lambda} \sigma e''' \dot{u} \dot{v} + 3\sigma e''' \ddot{v} \\ &- 3\dot{\lambda}^2 \hat{p} f'' \dot{v} - 6\dot{\lambda}^2 \hat{p} f''' \dot{u} \dot{v} - 3\dot{\lambda} \hat{p} f'' \dot{v} \\ &- 3\dot{\lambda} \hat{p} f'' \dot{v}^2 - 3\dot{\lambda} \hat{p} f''' \ddot{v} - 3\dot{\lambda}^2 p f''' \dot{u}^2 \dot{v} \\ &- 3\dot{\lambda} p f''' \dot{u} \dot{v}^2 - p f''' \dot{v}^3 - 3\dot{\lambda} p f''' \dot{u} \dot{v} \\ &- 3\dot{\lambda}^2 p f''' \dot{u} \dot{v} - 3\dot{\lambda} p f''' \dot{u} \dot{v} - 3p f''' \ddot{v}) \delta u = 0, \end{aligned} \tag{9}$$

$$\begin{aligned} \ddot{\tau} &= s' \ddot{\gamma} + 3\dot{\lambda}^2 s'' e'' \dot{\gamma} + 3\dot{\lambda} s''' \hat{\epsilon} \dot{\gamma}^2 + s''' \dot{\gamma}^3 + 3\dot{\lambda} s'' \hat{\epsilon} \ddot{\gamma} \\ &+ 3\dot{\lambda}^2 s'' \hat{\epsilon} \dot{\gamma} + 3\dot{\lambda} s''' \hat{\epsilon} \dot{\gamma} + 3s''' \dot{\gamma} \dot{\gamma}, \\ \ddot{\gamma} &= e' \ddot{v} + 3\dot{\lambda}^2 e'' \dot{u}^2 \dot{v} + 3\dot{\lambda} e''' \dot{u} \dot{v}^2 + e''' \dot{v}^3 \\ &+ 3\dot{\lambda} e'' \dot{u} \ddot{v} + 3\dot{\lambda}^2 e'' \dot{u} \dot{v} + 3\dot{\lambda} e'' \dot{u} \dot{v} + 3e'' \ddot{v}, \end{aligned}$$

where  $\hat{\cdot}$  denotes differentiation with respect to  $\lambda$  and a superposed dot differentiation with respect to  $t$ . From eqns (8), by imposing Fredholm compatibility condition, we obtain the equation which determines the value of  $\dot{\lambda}$

$$\begin{aligned} s'' \dot{\gamma}^3 + 3\hat{\tau} e'' \dot{v}^2 + \sigma e''' \dot{v}^3 - p f''' \dot{v}^3 + 2\dot{\lambda} (s'' \hat{\epsilon} \dot{\gamma}^2 \\ + 2\hat{\tau} e'' \dot{u} \dot{v} + \hat{\sigma} e'' \dot{v}^2 + \sigma e''' \dot{u} \dot{v}^2 - \hat{p} f'' \dot{v}^2 \\ - p f''' \dot{u} \dot{v}^2) = 0. \end{aligned} \tag{10}$$

Again, the Fredholm compatibility condition for eqns (9) gives the value of  $\dot{\lambda}$  through the equation

$$\begin{aligned} s''' \dot{\gamma}^4 + 3s'' \dot{\gamma}^2 \dot{\gamma} + 3\hat{\tau} e'' \dot{v}^2 + 4\hat{\tau} e''' \dot{v}^3 + 6\hat{\tau} e'' \ddot{v} \\ + \sigma e''' \dot{v}^4 + 3\sigma e''' \dot{v}^2 \ddot{v} - p f''' \dot{v}^4 - 3p f''' \dot{v}^2 \ddot{v} \\ + 3\dot{\lambda} (s''' \hat{\epsilon} \dot{\gamma}^3 + s'' \hat{\epsilon} \dot{\gamma} \dot{\gamma} + \hat{\tau} e'' \dot{u} \dot{v} + \hat{\tau} e'' \dot{u} \dot{v} + \hat{\sigma} e''' \dot{v}^3 \\ + 3\hat{\tau} e'' \dot{u} \dot{v}^2 + \hat{\sigma} e'' \dot{v} \ddot{v} + \sigma e''' \dot{u} \dot{v}^3 + \sigma e''' \dot{u} \dot{v} \ddot{v} \\ - \hat{p} f''' \dot{v}^3 - \hat{p} f'' \dot{v} \ddot{v} - p f''' \dot{u} \dot{v}^3 - p f''' \dot{u} \dot{v} \ddot{v}) \\ + 3\dot{\lambda}^2 (s''' \hat{\epsilon}^2 \dot{\gamma}^2 + s'' \hat{\epsilon} \dot{\gamma}^2 + 2\hat{\tau} e'' \dot{u}^2 + \hat{\sigma} e'' \dot{v}^2 \\ + 2\hat{\sigma} e'' \dot{u} \dot{v}^2 + 2\hat{\tau} e'' \dot{u} \dot{v} + \sigma e''' \dot{u}^2 \dot{v}^2 + \sigma e''' \dot{u} \dot{v}^2 \\ - \hat{p} f'' \dot{v}^2 - 2\hat{p} f'' \dot{u} \dot{v}^2 - p f''' \dot{u}^2 \dot{v}^2 - p f''' \dot{u} \dot{v}^2) \\ + 3\dot{\lambda} (s'' \hat{\epsilon} \dot{\gamma}^2 + \hat{\sigma} e'' \dot{v}^2 + 2\hat{\tau} e'' \dot{u} + \sigma e''' \dot{u} \dot{v}^2 - \hat{p} f'' \dot{v}^2 \\ - p f''' \dot{u} \dot{v}^2) = 0. \end{aligned} \tag{11}$$

### 3. BEAM MODEL (1-D POLAR CONTINUUM IN 2-D)

In this section we briefly outline the continuum beam model which will be used in the sequel. In Fig. 1 the beam  $\mathcal{B}$  is depicted in its reference shape  $\kappa(\mathcal{B})$  and in the generic one  $\chi(\mathcal{B})$ , together with the set of directors,  $\mathbf{B}_i$  and  $\mathbf{b}_i$  ( $i = 1, 2$ ), respectively. Here the prime denotes differentiation with respect to the reference abscissa  $s$ .

Let the displacement vector  $\mathbf{u}$  and the rotation tensor  $\mathbf{R}$  be

$$\begin{aligned} \mathbf{u}(s) &:= \mathbf{x}(s) - \mathbf{X}(s) = u(s)\mathbf{b}_1 + v(s)\mathbf{b}_2, \\ \mathbf{R}(s) &: \mathbf{B}_i(s) \mapsto \mathbf{b}_i(s). \end{aligned} \tag{12}$$

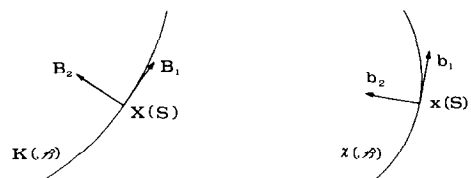


Fig. 1. Reference and generic beam shape.

Let the strain measures be defined as

$$\begin{aligned} \epsilon(s) &:= \mathbf{x}'(s) - \mathbf{R}(s) \mathbf{X}'(s) = \epsilon \mathbf{b}_1 + \gamma \mathbf{b}_2, \\ \mu(s) &:= \frac{1}{2}(\mathbf{R}'\mathbf{R}')_{\times} = \mu \mathbf{b}_3, \end{aligned} \tag{13}$$

$\times$  denoting the Gibbs cross and  $R'$  the transpose of  $R$ . The corresponding scalar compatibility relations read

$$\begin{aligned} \epsilon &= u' - (k + \theta')v + \cos \theta - 1, \\ \gamma &= v' + (k + \theta')u - \sin \theta, \\ \mu &= \theta'. \end{aligned} \tag{14}$$

Let  $\mathbf{n}(s)$ ,  $\mathbf{m}(s)$  denote internal force and couple vector fields. The components we will refer to are defined by

$$\begin{aligned} \mathbf{n} &= N\mathbf{b}_1 + T\mathbf{b}_2, \\ \mathbf{m} &= M\mathbf{b}_3. \end{aligned} \tag{15}$$

4. AUTOMATIC GENERATION OF THE PERTURBATION EQUATIONS FOR BEAMS

Let us examine now how to specialize the perturbation equations to the continuum beam model introduced in the previous section, and how this process can be performed automatically by some algebraic manipulation procedures.

From now on the vectors  $\mathbf{u}$ ,  $\epsilon$ ,  $\sigma$  appearing in eqn (2) will assume the form

$$\begin{aligned} \mathbf{u} &= [u(s), v(s), \theta(s)], \\ \epsilon &= [\epsilon(u), \gamma(u), \mu(u)], \\ \sigma &= [N(s), T(s), M(s)]. \end{aligned} \tag{16}$$

It is necessary to point out that, at this step of the analysis, the beam model just defined is essentially "kinematic" in the sense that nothing was said about the constitutive relation (4)<sub>2</sub>. Indeed, for the sake of generality, it is convenient to look at the constitutive relation as an element characterizing each particular problem as are the load functions  $p(\lambda)$ ,  $f(u)$  and the known equilibrium path. Consequently we do not specify any of them at this stage. Hence we will consider neither the terms in eqns (7)–(9) in which appears the external load, nor the asymptotic expansion of eqn (4)<sub>2</sub>.

For generating all the other terms in eqns (7)–(9) a REDUCE program has been implemented. The features of the main modules will be outlined here.

As it clearly appears looking at the asymptotic equations (7)–(9), what is needed first are the Gateaux differentials of functions (14) up to the fourth order, calculated along four different directions  $u_1$ ,  $u_2$ ,  $u_3$ ,  $u_4$ .

These are generated by the module GATEAUX, which is listed in Appendix A together with its output. This module is able to perform the differentiation process up to a specified order.

A little bit more sophisticated is the module that accomplishes the generation of the differential equilibrium equations starting from the terms.

$$\begin{aligned} \sigma e'' \dot{v} \delta u, \\ \dot{\tau} e' \delta u, \end{aligned} \tag{17}$$

appearing in eqn (7)<sub>1</sub> and from the corresponding ones at higher orders. We have to recall that terms (17) stand for scalar products which represent virtual works, that is

$$\begin{aligned} \dot{\tau} e' \delta u &\equiv (\dot{\tau}, e'(u) \delta u) \equiv \int_{\epsilon} \dot{\tau}(s) e'[u(s)] \delta u(s) ds, \\ \sigma e'' \dot{v} \delta u &\equiv (\sigma, e''(u) \dot{v} \delta u) \\ &\equiv \int_{\epsilon} \sigma(s) e''[u(s)] \dot{v}(s) \delta u(s) ds. \end{aligned} \tag{18}$$

In order to obtain the asymptotic equilibrium equations, we have to construct the adjoints of the differential operators

$$e'(u); \quad e''(u) \dot{v}. \tag{19}$$

Such operators can be obtained, together with the boundary conditions defining their domain, by performing successive integrations by parts. This can be done by using the module GREEN which is listed in Appendix A. It essentially makes use of the standard differentiation function DF, associated with the statement COEFF in order to isolate the coefficients of the derivatives of  $\delta u$ .

Once the load function and the constitutive relation have been specified, Gateaux differentiations and integration by parts on the rest of the terms in eqn (7) can be performed by modules which operate exactly as the modules GATEAUX and GREEN but on different expressions.

Before moving on to discuss how to solve a specific problem using the results of the previous steps, it is important to stress the fundamental role played by the output facilities of an algebraic manipulation system.

In REDUCE, where the output is essentially one-dimensional, the user has to face some difficulties when the variables that he is dealing with are not very few and their formal derivatives are to be generated. In fact, taking successive derivatives means nothing but generating new symbols in a recursive way, while for the sake of clarity these symbols must retain information about their origin, in order to be easily detectable as derivatives of the original variables. Using the expression DF(F(x), x) for denoting derivatives, previously defining F as an operator, is quite unsatisfactory. In order to obtain a representation for formal derivatives as short and clear as possible, a different notation has been devised. In this respect systems like SCRATCHPAD or MACSYMA, containing two-dimensional output facilities, should prove more suitable.

## 5. A SOLUTION PROCEDURE

As an application of the general procedure for curved beams, the bifurcation analysis of a circular arch under hydrostatic pressure, shown in Fig. 2, has been carried out.

The constitutive relations considered are the following

$$\begin{aligned} N &= A\epsilon, \\ T &= G\gamma, \\ M &= B\mu, \end{aligned} \quad (20)$$

where  $A$ ,  $G$ ,  $B$  are constants.

The external force field is defined by

$$\begin{aligned} p(\lambda) &= \lambda, \\ f(u) &= \int_{\mathcal{C}} p[\frac{1}{2}(k + \theta')(u^2 + v^2) \\ &\quad - v \cos \theta - u \sin \theta - u'v] ds \end{aligned} \quad (21)$$

while the only nonvanishing function along the fundamental path is

$$v = \frac{\lambda p R^2}{(\lambda p R + A)}. \quad (22)$$

It is possible now to complete the generation of the perturbation equations.

Looking at eqn (7)<sub>1</sub> the only term which has been neglected is

$$p f'' \dot{u} \delta u. \quad (23)$$

This can be constructed by performing Gateaux differentiations of  $f(u)$  as was done for  $e(u)$ . The subsequent integration by parts follows the same procedure used for the terms in (17). The eqn (7)<sub>2</sub> is trivially derived from (20) noting that  $s'(\epsilon)$  is represented by a diagonal matrix whose entries are  $A$ ,  $G$ ,  $B$ . The eqn (7)<sub>3</sub> can be evaluated from the first differential of  $e(u)$ , obtained as output of GATEAUX, by substituting  $\dot{u}$  for  $u_1$ . Now, if eqns (7)<sub>1</sub>, (7)<sub>2</sub>, (7)<sub>3</sub> reside simultaneously in a REDUCE environment, an automatic substitution is performed leading to a homogeneous system of three ordinary differential equations of the second order, with constant coefficients, defining an eigenvalue problem.

A standard method for solving this kind of differential equations, implemented as an automatic procedure, has been used to compute the solution given in Appendix B.

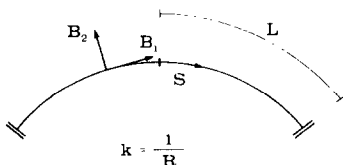


Fig. 2. Arch under hydrostatic pressure.

The next step of the analysis is the calculation of the expression for  $\dot{\lambda}_c$ . The module LAMBADADOT, which is listed in Appendix C, performs the task of constructing the expressions for the terms appearing in the orthogonality condition (10). After that each of them is to be integrated over the interval  $[-L, L]$ . Such integration is automatically performed by another module.

For the case we are dealing with  $\dot{\lambda}_c$  turns out to be zero.

The eqns (8) can then be obtained following the same line as for eqns (7), now getting to a system of ordinary linear differential equations with constant coefficients which are not homogeneous. It can easily be seen that the solution has the form

$$\ddot{u} = \ddot{u}_p + d \dot{u}, \quad (24)$$

where  $\dot{u}$  is the solution to the first-order equations,  $d$  is an arbitrary coefficient and  $\ddot{u}_p$  is any particular solution. An expression for  $\ddot{u}_p$  has been found by checking some trial functions and we omit writing it here for the sake of brevity.

Once such a solution has been found the last task to be performed is the evaluation of  $\dot{\lambda}_c$  through eqn (11), using the same technique as that used for evaluating  $\dot{\lambda}_c$ . The REDUCE output for the numerator and denominator or  $\dot{\lambda}_c$  is given in Appendix D.

Thus as a final result we have obtained a general parametric solution with no need to normalize the first (eigenvector) and the second-order solution vector, so retaining the possibility of dealing with a wide class of cases simply performing numerical substitutions. This is easily achieved by automatically translating the symbolic expressions of the results into FORTRAN statements.

It is worth noting here that, in spite of the difficulties in dealing with a sophisticated beam model, both the generation of the field equations and the construction of the solution have been performed using an algebraic manipulation system, thus avoiding hand calculations that would have been unmanageable.

## 6. PROGRAM STRUCTURE

As explained in the previous sections, the program is made up of modules to be called and executed in a REDUCE environment in a certain sequence. A called module interacts with the environment in the sense that it can introduce new identifiers, bind some others and so on.

For the sake of clarity each module contains not only the formal description of the actions to be performed, but also a documentation about the identifiers involved in the actions. To this end it has proved useful to divide the identifiers into the following groups: (1) input identifiers are identifiers the module needs in the sense that they must be present in the environment when the module is called; if operators they must be defined as well;

(2) output identifiers are identifiers which are assigned some value by the module. They can be (a) already present in the environment, then their value is modified; (b) new identifiers, then they are not only introduced but also assigned a value; (3) new identifiers are identifiers introduced by the module but not bound to any other identifier, nor assigned a value; (4) local identifiers are dummy identifiers which reside in the environment only for the time the module is running. They are cleared after the actions in which they are involved have been performed.

7. CONCLUSIONS

In this paper the use of an algebraic manipulation system in buckling problems for curved beams with exact kinematics has been outlined. As an application the bifurcation analysis for a circular arch under hydrostatic pressure, up to the second order, has been presented.

The use of an algebraic manipulation system has given evidence of how even a cumbersome procedure, such as the bifurcation analysis carried out, can be handled with a little effort while providing reliable results.

It is worth noting that, although it has been possible in the above application to find a closed form solution to the perturbation equations, this is not the case in general. Then it would be necessary to devise a procedure for finding an approximate solution. To this end the described symbolic procedures would play a relevant role, at least in deriving the perturbation equations in the integral form, as shown in the previous sections. Those equations should in fact be the starting point for constructing an approximate formulation. But a discussion on such a subject is beyond the scope of this work.

REFERENCES

1. B. Budiansky, Theory of Buckling and Postbuckling Behaviour of Elastic Structures, in *Advances in Applied Mechanics 14*, (Edited by Chia Shun Yih) Academic Press, New York (1974).
2. P. D'Asdia, A. Di Carlo and N. Rizzi, Nonlinear beam models in frame postbuckling analysis: An asymptotic approach. Istituto di Scienza delle Costruzioni Università di Roma Rep. II-204 (1977).
3. A. C. Hearn, REDUCE-2. A system and language for algebraic manipulation, ACM, *Proc. Second Symp. on Symbolic and Algebraic Manipulation*, Los Angeles (1971).
4. C. Griss, Using REDUCE on the UNIVAC 1110. Utah Symbolic Computation Group, Techn. Rep. TR-4 August, 1978.
5. A. K. Noor and C. M. Andersen, Computerized symbolic manipulation in structural mechanics—Progress and potential. *Comput. Structures 10*, 95–118 (1979).
6. A. K. Noor and C. D. Balch, Hybrid perturbation/Bubnov-Galerkin technique for nonlinear thermal analysis. *AIAA J. 22*, 287–294 (1984).

APPENDIX A

Modules for generating the perturbation equations

Identifiers defined as operators are used to allow a representation for derivatives of functions. Recalling that in REDUCE operators are represented through identifiers with subscripts in parentheses, we use the following rule: the first subscript (., ) is a number denoting the order of differentiation with respect to the curvilinear abscissa; the second subscript ( ,.), when a number, denotes generic direction in the Gateaux differentials, different numbers denoting different directions.

With reference to directions of particular interest, alphanumeric expressions will be used as subscripts. In particular

$$u(n, DOT) \text{ will stand for } \dot{u}^{(n)},$$

$$u(n, CFX) \text{ will stand for } u^{(n)},$$

$u$  being a generic function identifier.

Only for EPS, GAM, MU, the second subscript, when a number, denotes order of Gateaux differentiation.

```

1      %* * * * *
2      *
3      * Module   GATEAUX
4      *
5      * Computation of formal Gateaux derivatives of the strain measures.
6      * Components up to the order specified by "DifferrentiationOrder"
7      * variable.
8      *
9      *
10     *
11     * Input identifiers : eps(0),gam(0),mu(0)
12     *
13     * Output identifiers : eps(0,i),gam(0,i),mu(0,i)
14     *                      i:=1:DifferntiationOrder
15     *
16     * New identifiers   : u(m,i),v(m,i),te(m,i)
17     * "m" has the values already introduced in the
18     * expressions for the strain measures components
19     *                      i:=1:DifferntiationOrder
20     *
21     * Local identifiers : eps(0,0),gam(0,0),mu(0,0),a,b,c,j
22     *
23     * * * * *
24
25
26

```

```

27 DifferentiationOrder:=4;
28
29 for i:=1;DifferentiationOrder do
30   begin
31     for all m let df(u(m),gateaux)=u(m,i),
32       df(v(m),gateaux)=v(m,i),
33       df(te(m),gateaux)=te(m,i);
34
35     for all m,n let df(u(m,n),gateaux)=0,
36       df(v(m,n),gateaux)=0,
37       df(te(m,n),gateaux)=0;
38
39
40     let eps(0,0)=eps(0);
41     let gam(0,0)=gam(0);
42     let mu(0,0)=mu(0);
43
44     j:=i-1;
45     a:=df(eps(0,j),gateaux)*
46     b:=df(gam(0,j),gateaux)*
47     c:=df(mu(0,j),gateaux)*
48
49     eps(0,i):=a*
50     gam(0,i):=b*
51     mu(0,i):=c*
52   end;
53
54 %-----
55 - Clear substitutions and local identifiers -
56 -----
57
58
59 clear eps(0,0),gam(0,0),mu(0,0);
60 for all m clear df(u(m),gateaux),
61   df(v(m),gateaux),
62   df(te(m),gateaux);
63
64 for all m,n clear df(u(m,n),gateaux),
65   df(v(m,n),gateaux),
66   df(te(m,n),gateaux);
67
68 clear a,b,c;
69
70
71 %***** E * N * D * *****
72 >
1 %*****
2 *
3 * Module GREEN
4 *
5 * Evaluation of the following terms of the first order perturbation
6 * relationships :
7 * - equilibrium equations without terms involving external loads
8 * - kinematical relationships
9 *
10 *
11 * f(.,VAR) denotes the "variation" of f
12 * f(.,DOT) denotes the "dot" of f
13 *
14 *
15 * Input identifiers : eps(0,1),gam(0,1),mu(0,1),
16 * eps(0,2),gam(0,2),mu(0,2)
17 *
18 * Output identifiers : eqz1,eqz2,eqz3,BoundaryTerms
19 *
20 * New identifiers : n(.),n(.,DOT),q(.),q(.,DOT),m(.),m(.,DOT)
21 * u(.,DOT),v(.,DOT),te(.,DOT)
22 *
23 * Local identifiers : alfa,beta,gamma (used in COEFF),
24 * iord,j,a,b,c,expression,boundary
25 *
26 *
27 %*****
28
29
30 operator n,q,m $
31
32 for all m let u(m,1)=u(m,VAR),v(m,1)=v(m,VAR),te(m,1)=te(m,VAR),
33   u(m,2)=u(m,DOT),v(m,2)=v(m,DOT),te(m,2)=te(m,DOT) $
34
35 %-----
36 - First order expression for the contact forces power density -
37 -----
38
39 expression:= n(0)*eps(0,2)+q(0)*gam(0,2)+m(0)*mu(0,2)
40   +n(0,DOT)*eps(0,1)+q(0,DOT)*gam(0,1)+m(0,DOT)*mu(0,1) $
41
42 % Initialization to 0 for identifiers used as accumulators $
43
44 eqz1:=eqz2:=eqz3:=0$
45 BoundaryTerms:=0$

```

```

46
47 %- -----
48 - Differentiation pattern through substitutions -
49 -----
50
51 for all k let
52   df(u(k),s)=u(k+1),
53   df(v(k),s)=v(k+1),
54   df(te(k),s)=te(k+1),
55   df(n(k),s)=n(k+1),
56   df(q(k),s)=q(k+1),
57   df(m(k),s)=m(k+1),
58   df(u(k,DOT),s)=u(k+1,DOT),
59   df(v(k,DOT),s)=v(k+1,DOT),
60   df(te(k,DOT),s)=te(k+1,DOT),
61   df(n(k,DOT),s)=n(k+1,DOT),
62   df(q(k,DOT),s)=q(k+1,DOT),
63   df(m(k,DOT),s)=m(k+1,DOT) ;
64
65 % iord := differentiation order with respect to the abscissa $
66
67 integer iord;
68 iord := 0;
69
70
71 %- -----
72 - Integration by parts and evaluation of the boundary terms -
73 -----
74
75 while expression neq 0 do
76   begin
77     alfa0:=alfa1:=beta0:=beta1:=gamma0:=gamma1:=0$
78     coeff(expression,u(iord,VAR),alfa)$
79     coeff(alfa0,v(iord,VAR),beta)$
80     coeff(beta0,te(iord,VAR),gamma)$
81
82     expression:=gamma0;
83
84     a:=df(alfa1,s,iord);
85     b:=df(beta1,s,iord);
86     c:=df(gamma1,s,iord);
87
88     eqz1:=eqz1+(-1)**iord#a;
89     eqz2:=eqz2+(-1)**iord#b;
90     eqz3:=eqz3+(-1)**iord#c;
91
92
93     boundary := for j:=1:iord sum
94       (-1)**(iord-1)*df(alfa1,s,j-1)*u(iord-j,1)+
95       (-1)**(iord-1)*df(beta1,s,j-1)*v(iord-j,1)+
96       (-1)**(iord-1)*df(gamma1,s,j-1)*te(iord-j,1)$
97
98     BoundaryTerms:=BoundaryTerms + boundary;
99
100    iord:=iord+1;
101    write "iord := ",iord
102  end$
103
104
105 %- -----
106 - Clear substitutions and local identifiers -
107 -----
108
109 for all k clear
110   df(u(k),s) ,df(u(k,DOT),s),
111   df(v(k),s) ,df(v(k,DOT),s),
112   df(te(k),s) ,df(te(k,DOT),s),
113   df(n(k),s) ,df(n(k,DOT),s),
114   df(q(k),s) ,df(q(k,DOT),s),
115   df(m(k),s) ,df(m(k,DOT),s);
116
117 iord:=0;
118 clear a,b,c,boundary,expression$
119 for all m clear u(m,1),v(m,1),te(m,1),u(m,2),v(m,2),te(m,2) $
120
121
122 % * * * * E * N * D * * * * *
>

```

When  $\text{eps}(0, 0)$ ,  $\text{gam}(0, 0)$ ,  $\text{mu}(0, 0)$  are given the expressions (14) the module Gateaux produces the following differentials:

```

EPS(0,1):=U(1,1) - V(0,1)*TE(1) - V(0,1)*K - SIN(TE(0))*TE(0,1) - TE(1,1)*V(0)
EPS(0,2):= - (V(0,1)*TE(1,2) + TE(0,1)*TE(0,2)*COS(TE(0)) + TE(1,1)*V(0,2))
EPS(0,3):=SIN(TE(0))*TE(0,1)*TE(0,2)*TE(0,3)
EPS(0,4):=TE(0,1)*TE(0,2)*COS(TE(0))*TE(0,3)*TE(0,4)
GAM(0,1):=U(0,1)*TE(1) + U(0,1)*K + V(1,1) + U(0)*TE(1,1) - TE(0,1)*COS(TE(0))
GAM(0,2):=U(0,1)*TE(1,2) + U(0,2)*TE(1,1) + SIN(TE(0))*TE(0,1)*TE(0,2)
GAM(0,3):=TE(0,1)*TE(0,2)*COS(TE(0))*TE(0,3)
GAM(0,4):= - SIN(TE(0))*TE(0,1)*TE(0,2)*TE(0,3)*TE(0,4)
MU(0,1):=TE(1,1)
MU(0,2):=0
MU(0,3):=0
MU(0,4):=0
    >
    
```

Once the required differentials of the expressions (14) have been derived the module Green produces the following results:

```

EQZ1:=Q(0,DDT)*TE(1) + Q(0,DDT)*K + Q(0)*TE(1,DDT) - N(1,DDT)
EQZ2:= - (N(0,DDT)*TE(1) + N(0,DDT)*K + Q(1,DDT) + TE(1,DDT)*N(0))
EQZ3:= - Q(0,DDT)*COS(TE(0)) - Q(0,DDT)*U(1) + V(1,DDT)*N(0) + N(1)*V(0,DDT) - U(1,DDT)*Q(0) - Q(1)*U(0,DDT) - N(0,DDT)*SIN(TE(0)) + N(0,DDT)*V(1) - Q(1,DDT)*U(0) + Q(0)*SIN(TE(0))*TE(0,DDT) - M(1,DDT) - TE(0,DDT)*COS(TE(0))*N(0) + V(0)*N(1,DDT)
BOUNDARYTERMS:=Q(0,DDT)*U(0)*TE(0,VAR) + Q(0,DDT)*V(0,VAR) + N(0,DDT)*U(0,VAR) - N(0,DDT)*V(0)*TE(0,VAR) + U(0,DDT)*Q(0)*TE(0,VAR) - V(0,DDT)*TE(0,VAR)*N(0) + TE(0,VAR)*M(0,DDT)
    >
    
```

**APPENDIX B**

*Solution to the first-order perturbation equations*

The bifurcation points along the fundamental solution are defined by the values of  $\lambda$  which are roots of the equation

$$\{(G + 3A)\lambda^3 R^3 B + [3(G + A)B + R^2 A^2]\lambda^2 R^2 A + [(3G + A)B + R^2 G A]\lambda R A^2 + \lambda^4 R^4 B + G B A^3\} / [(\lambda R + A)^2 R^2 G B A] = h^2,$$

where the value of  $h$  is such that either  $\cos(hL) = 0$  or  $\sin(hL) = 0$ . If we consider the case  $hL = \pi/2$  the corresponding eigensolution to the first-order equations is

$$\dot{u} = a_1 \cos(hs)$$

$$\dot{v} = -a_2 \sin(hs)$$

$$\dot{\theta} = a_3 \cos(hs)$$

where  $a_1, a_2, a_3$  stand for the following expressions:

$$a_1 = R[-\lambda^3 R^3 - \lambda^2 R^2(G + 2A) + \lambda R A(h^2 R^2 G - 2G - A) - G A^2],$$

$$a_2 = hR^2 A^2(\lambda R + G),$$

$$a_3 = [\lambda^3 R^3 + \lambda^2 R^2(G + 2A) - \lambda R A(h^2 R^2 G - 2G - A) + G A^2(1 - h^2 R^2)].$$



APPENDIX C

Module for constructing the expression for  $\lambda_c$ .

```

1      %* * * * *
2      *
3      * Module LAMBADADOT
4      *
5      * Evaluation of the terms appearing in the expression for lambdadot
6      * and containing internal force and couple densities.
7      * (Here hyperelastic constitutive relationships are assumed).
8      *
9      *
10     *          SigE3VD3                      (numerator)
11     *    3*TauDE2VD2                        (numerator)
12     *          SigE3UFVD2                   (denominator)
13     *          SigFE2VD2                    (denominator)
14     *    2*TauDE2UFVD                       (denominator)
15     *
16     *
17     *
18     *Input identifiers : n(0),q(0),m(0),
19     *                   n(0,CFX),t(0,CFX),m(0,CFX),
20     *                   n(0,DDT),q(0,DDT),m(0,DDT),
21     *                   u(0,CFX),v(0,CFX),t(0,CFX),
22     *                   u(0,DDT),v(0,DDT),te(0,DDT),
23     *                   eps(0,2),gam(0,2),mu(0,2),
24     *                   eps(0,3),gam(0,3),mu(0,3)
25     *
26     *Output identifiers : SigE3VD3, TauDE2VD2, SigE3UFVD2,
27     *                   SigFE2VD2,TauDE2UFVD
28     *
29     *New identifiers   : none
30     *
31     *Local identifiers : none
32     *
33     *
34     *N.B. Being the output identifiers densities of scalar product,the
35     * corresponding expressions have to be integrated
36     *
37     %* * * * *
38
39
40     %- -----
41     - Substitution pattern :   f(.,i) ==> f(.,DDT)   (i=1..3)   -
42     -----
43
44     for i:=1:3 do
45     begin
46         for all m let u(m,i)=df(u(0,DDT),s,m),
47             v(m,i) =df(v(0,DDT),s,m),
48             te(m,i)=df(te(0,DDT),s,m)
49     end;
50
51
52     SigE3VD3:=n(0)*eps(0,3)+q(0)*gam(0,3)+m(0)*mu(0,3) $
53
54     TauDE2VD2:=n(0,DDT)*eps(0,2)+q(0,DDT)*gam(0,2)+m(0,DDT)*mu(0,2) $
55
56     SigFE2VD2:=n(0,CFX)*eps(0,2)+q(0,CFX)+m(0,CFX) $
57
58
59     %- -----
60     - Substitution pattern :   f(.,1) ==> f(.,CFX)   -
61     ..                        f(.,j) ==> f(.,DDT)   (j=2,3)   -
62     -----
63
64     for all m let u(m,1)=df(u(0,CFX),s,m),
65             v(m,1)=df(v(0,CFX),s,m),
66             te(m,1)=df(te(0,CFX),s,m)
67
68
69     SigE3UFVD2:=n(0)*eps(0,3)+q(0)*gam(0,3)+m(0)*mu(0,3) $
70
71     TauDE2UFVD:=n(0,DDT)*eps(0,2)+q(0,DDT)*gam(0,2)+m(0,DDT)*mu(0,2) $
72
73
74     %- -----
75     - Clear Substitutions
76     -----
77
78     for all m,i clear u(m,i),v(m,i),te(m,i) $
79
80
81     %* * * * * E * N * D * * * * *
82     >

```

APPENDIX D

Expressions for numerator and denominator of  $\lambda_c$ .

$$\begin{aligned}
 \text{NUMp2} &:= 3A^{**2}L * (2a1^{**2}a3^{**2}R^{**2}h^{**2} + 2a1d4a3R^{**2}h \\
 &+ 4a1d4a2R^{**2}h^{**2} + 3a1a3^{**3}R + 4a1a3^{**2}a2R \\
 &*h - a1a3d1R^{**h} + a1a3d2R^{**2}h^{**2} - 3a1a3d2 - 5a \\
 &a1a3d3R^{**2}h^{**2} - a1a3d3 + 2d4a3a2R + 4d4a2^{**2} \\
 &*R*h) + 3a3^{**4}R^{**2} + 6a3^{**3}a2R^{**2}h - 2a3^{**2}d1R^{**2} \\
 &*h + 8a3^{**2}a2^{**2}R^{**2}h^{**2} - 3a3^{**2}d2R - a3^{**2}d3R \\
 &+ a3d1a2 - 3a3a2d2R^{**h} - 9a3a2d3R^{**h}) / (4AR^{**2} \\
 &+ 4p0R^{**3}) + 3AG*L * (- 4a1^{**2}d4R^{**h} - 4a1d4a3R \\
 &R^{**2}h - 4a1d4a2R^{**2}h^{**2} - 4a1a3^{**3}R - 4a1a3^{**2} \\
 &a2R^{**h} + 3a1a3d1R^{**h} + 3a1a3d2 + a1a3d3 - d4a3a \\
 &a2R - 4a3^{**4}R^{**2} - 6a3^{**3}a2R^{**2}h + 3a3^{**2}d1R^{**2} \\
 &*h - 2a3^{**2}a2^{**2}R^{**2}h^{**2} + 3a3^{**2}d2R + a3^{**2}d3R \\
 &+ 3a3d1a2R^{**2}h^{**2} - a3d1a2 + a3a2d2R^{**h} + 3a3a \\
 &a2d3R^{**h}) / (4AR^{**2} + 4p0R^{**3}) + 3Ap0L * (- 2a1^{**} \\
 &2d4R^{**2}h + 2a1^{**2}a3^{**2}R^{**2}h^{**2} - 2a1d4a3R^{**2}h + \\
 &4a1d4a2R^{**2}h^{**2} + 2a1a3^{**3}R^{**3}h^{**2} + 2a1a3^{**3}R \\
 &R + 4a1a3^{**2}a2R^{**h} + a1a3d2R^{**2}h^{**2} - 3a1a3d2 \\
 &- 5a1a3d3R^{**2}h^{**2} - a1a3d3 + 4d4a3a2R^{**3}h^{**2} \\
 &+ 3d4a3a2R + 6d4a2^{**2}R^{**h} + 2a3^{**4}R^{**2} + 6a3^{**3} \\
 &a2R^{**2}h + 8a3^{**2}a2^{**2}R^{**2}h^{**2} + a3^{**2}d3R^{**3}h^{**2} \\
 &- 3a3^{**2}d2R - 5a3^{**2}d3R^{**3}h^{**2} - a3^{**2}d3R + a3 \\
 &d1a2 - 4a3a2d2R^{**h} - 12a3a2d3R^{**h}) / (4AR + 4p0R \\
 &R^{**2}) + 3GBp0L * (- 4a1^{**2}d4R^{**h} - 4a1d4a3R^{**2}h \\
 &- 4a1d4a2R^{**2}h^{**2} - 4a1a3^{**3}R - 4a1a3^{**2}a2R \\
 &h + 3a1a3d1R^{**h} + 3a1a3d2 + a1a3d3 - d4a3a2R \\
 &- 4a3^{**4}R^{**2} - 6a3^{**3}a2R^{**2}h + 3a3^{**2}d1R^{**2}h \\
 &- 2a3^{**2}a2^{**2}R^{**2}h^{**2} + 3a3^{**2}d2R + a3^{**2}d3R + \\
 &3a3d1a2R^{**2}h^{**2} - a3d1a2 + a3a2d2R^{**h} + 3a3a2 \\
 &d3R^{**h}) / (4AR + 4p0R^{**2}) + 3p0^{**2}R*L * (- 2a1^{**2} \\
 &d4h - 2a1d4a3R^{**h} - a1a3^{**3} + a1a3d1h + d4a3a2 \\
 &+ 2d4a2^{**2}h - a3^{**4}R + a3^{**2}d1R^{**h} - a3a2d2h - 3 \\
 &a3a2d3h) / (4A + 4p0R) ; \\
 \\
 \text{DENp2} &:= 3A^{**3}L * (- a1^{**2} - a1a3R^{**3}h^{**2} - 2a1a3R - 2 \\
 &a1a2R^{**h} - a3^{**2}R^{**2} - 3a3a2R^{**2}h - a2^{**2}) / (A^{**3}R \\
 &+ 3A^{**2}p0R^{**2} + 3Ap0^{**2}R^{**3} + p0^{**3}R^{**4}) + 3A \\
 &^{**2}GBa3L * (a1 + a3R + a2R^{**h}) / (A^{**3} + 3A^{**2}p0R + 3A \\
 &p0^{**2}R^{**2} + p0^{**3}R^{**3}) + 3A^{**2}p0L * (- 3a1^{**2} - \\
 &a1a3R^{**3}h^{**2} - 5a1a3R - 6a1a2R^{**h} - a3^{**2}R^{**4}h \\
 &^{**2} - 2a3^{**2}R^{**2} - 6a3a2R^{**2}h - 3a2^{**2}) / (A^{**3} + 3A \\
 &^{**2}p0R + 3Ap0^{**2}R^{**2} + p0^{**3}R^{**3}) + 3AGa3 \\
 &p0R * L * (a1 + a3R + a2R^{**h}) / (A^{**3} + 3A^{**2}p0R + 3Ap0 \\
 &^{**2}R^{**2} + p0^{**3}R^{**3}) + 3Ap0^{**2}R * L * (- 3a1^{**2} - 5a \\
 &a1a3R - 6a1a2R^{**h} - 2a3^{**2}R^{**2} - 5a3a2R^{**2}h - 3 \\
 &a2^{**2}) / (A^{**3} + 3A^{**2}p0R + 3Ap0^{**2}R^{**2} + p0^{**3}R \\
 &^{**3}) + 3p0^{**3}R^{**2}L * (- a1^{**2} - 2a1a3R - 2 \\
 &a1a2R^{**h} - a3^{**2}R^{**2} - 2a3a2R^{**2}h - a2^{**2}) / (A^{**3} \\
 &+ 3A^{**2}p0R + 3Ap0^{**2}R^{**2} + p0^{**3}R^{**3}) ;
 \end{aligned}$$

>

The symbol  $p0$  stands for  $\lambda_c$ , while  $h = \pi/2L$ . The symbols  $a_1, a_2, a_3$  are constants appearing in the first-order solution, while  $d_1, d_2, d_3, d_4$  are constants appearing in the second-order solution. The expressions for them, as well as the expression for  $\lambda_c$ , are the intermediate results of the symbolic procedure. It is convenient to think of them as identifiers which take a numerical value, simply by evaluating the corresponding expressions.