



THE USE OF MATHEMATICA FOR THE ANALYSIS OF STRONGLY NONLINEAR TWO-DEGREE-OF-FREEDOM SYSTEMS BY MEANS OF THE MODIFIED LINDSTEDT–POINCARÉ METHOD

C. FRANCIOSI AND S. TOMASIELLO

DiSGG, Faculty of Engineering, University of Basilicata, Via della Tecnica 3, 85100, Potenza, Italy

(Received 16 December 1996, and in final form 2 May 1997)

An automatic code is proposed for the perturbation analysis of strongly non-linear two-degree-of-freedom systems with cubic non-linearities. The recently proposed [1] modified Lindstedt–Poincaré method is adopted, both because of its excellent performance and of its straightforward implementation. The symbolic software *Mathematica* is used in order to speed up all the cumbersome algebra which is inherent to every perturbation method.

© 1998 Academic Press Limited

1. INTRODUCTION

Since the pioneering works by Green [2] and Liouville [3], classical perturbation methods have been the subject of countless papers and books, from the classic Kevorkian and Cole [4] to the book by Linch [5] and the recent monograph by Holmes [6]. Many different approaches have been proposed, such as the Lindstedt–Poincaré (LP) method [7, 8], the powerful multiple scales approach [9] and the Krylov–Bogoliubov–Mitropolsky (KBM) technique [10] with its equivalent Galerkin method [11]. All of these approaches were originally limited to weak non-linearities and one-degree-of-freedom oscillators. Consequently, satisfactory results could be obtained only in limited parametric ranges, and internal resonance phenomena could not be investigated. More recently, the importance of modal interactions has begun to be understood, and a great deal of effort has been devoted to n -degree-of-freedom systems which exhibit internal resonances. For a complete review, the reader can consult the paper by Nayfeh and Balachandran [12], in which a lot of physical systems are illustrated where one-to-one, two-to-one and three-to-one internal resonances can occur. In reference [12] the multiple scales method was employed, but a LP method or a KBM technique could also be used.

If the non-linearities turn out to be too strong, the classical methods cannot give satisfactory results, and a number of improvements have been proposed. Basically, two different techniques can be adopted. In the first case, a classical perturbation method can be used to give an approximate solution, and then this solution can be conveniently used for a subsequent Galerkin analysis. Following these steps, Geer and Andersen [13] obtained very interesting results both for one-degree-of-freedom oscillators and for multiple-degree-of-freedom systems.

In the second case, the classical LP method is used, and a new perturbation parameter is defined, which remains small even if the original parameter grows without bound. The original idea can be traced back to Jones [14], even if it was greatly extended by Burton [15, 16], and deeply modified by Cheung and coworkers [17]. Quite recently this method has been generalized to strongly non-linear two-degree-of-freedom systems in the presence of internal resonance [1].

One of the major drawbacks of the perturbation methods lies in the cumbersome algebra which is inherent to the successive perturbation steps. For example, in an effort to obtain a seventh order approximation to the moon's orbit, Delaunay obtained an equation which covered 170 (printed) pages (quoted from Hagedorn [18], p. 2). Therefore, the recently developed computerized symbolic softwares have been immediately seen as ideal tools for such perturbation problems, and an early version of MACSYMA was used by Rand and Armbruster [19] for a pioneering investigation in perturbation theory. Nowadays, much more sophisticated and versatile symbolic programs are available, such as *Maple* [20] or *Mathematica* [21]; for example, all the calculations in the quoted paper by Geer and Anderson [13] were performed with the aid of *Mathematica*.

In this paper a computer-oriented version of the Chen and Cheung analysis for two-degree-of-freedom oscillators [1] is given, a computer program written in *Mathematica* is illustrated, and some examples are performed.

The advantages of the proposed approach seem to be twofold: first of all, parametric investigations become immediate, and the influence of various parameter changes can be easily detected. Moreover, with small efforts, the program can be modified to cope with other kinds of non-linearities, and finally, other kinds of external excitations can be studied. For example, a study in the presence of quasi-periodic forcing terms is currently under development.

2. METHOD OF ANALYSIS

Consider a general two-degree-of-freedom oscillator with cubic non-linearities, whose equations of motion can be written as

$$d^2x/dt^2 + \omega_{10}^2 x + \varepsilon(Ax^3 + Bx^2y + Cxy^2 + Dy^3) = \varepsilon p_1 \cos \omega t, \quad (1)$$

$$d^2y/dt^2 + \omega_{20}^2 y + \varepsilon(Ex^3 + Fx^2y + Gxy^2 + Hy^3) = \varepsilon p_2 \cos \omega t, \quad (2)$$

where ω_{10} , ω_{20} are the linear frequencies, A, \dots, H are the non-linearity coefficients, p_1 , p_2 are the excitation amplitudes, ω is the frequency of the forcing terms, and ε is the perturbation parameter. In order to study the internal resonance phenomena, one assumes that

$$\omega_{20}^2 = \omega_{10}^2 (\lambda_2/\lambda_1)^2 + \varepsilon \sigma_2 \quad (3)$$

if the fundamental resonance occurs when $\omega \approx \omega_{10}$, or

$$\omega_{10}^2 = \omega_{20}^2 (\lambda_1/\lambda_2)^2 + \varepsilon \sigma_1 \quad (4)$$

if $\omega \approx \omega_{20}$. In the first case (henceforth case (a)) it is also convenient to define $\omega_0 = \omega_{10}$, whereas in the other case (henceforth case (b)) it will be $\omega_0 = \omega_{20}$. The integers λ_1 and λ_2 are relatively prime, and σ_1 , σ_2 are the so-called detuning parameters.

The preliminary steps can be carried out as suggested by Chen and Cheung [1]. Therefore, ω^2 is expanded as a power series of ε ,

$$\omega^2 = \omega_0^2 + \sum_{k=1}^{\infty} \varepsilon^k \omega_k, \quad (5)$$

a new perturbation parameter α is defined as

$$\alpha = \varepsilon \omega_1 / (\omega_0^2 + \varepsilon \omega_1) \quad (6)$$

so that ω^2 turns out to be given by

$$\omega^2 = \frac{\omega_0^2}{1 - \alpha} \left(1 + \sum_{k=2}^{\infty} \alpha^k \delta_k \right), \quad (7)$$

and finally, the Lagrangian co-ordinates are expressed as series in the parameter α :

$$x = \sum_{k=0}^{\infty} \alpha^k x_k, \quad y = \sum_{k=0}^{\infty} \alpha^k y_k. \quad (8)$$

Assume for the moment that case (a) occurs. If equations (3, 6–8) are inserted into the equations of motion, then it is possible to write

$$\begin{aligned} eq1 &= (1 + \delta_2 \alpha^2 + \delta_3 \alpha^3 + \dots) \ddot{x} + (1 - \alpha)x + (\alpha/\omega_1)(Ax^3 + Bx^2y + Cxy^2 + Dy^3) \\ &\quad - (\alpha/\omega_1)p_1 \cos \tau = 0, \end{aligned} \quad (9)$$

$$\begin{aligned} eq2 &= (1 + \delta_2 \alpha^2 + \delta_3 \alpha^3 + \dots) \ddot{y} + (1 - \alpha)(\lambda_2^2/\lambda_1^2)y + (\alpha/\omega_1)\sigma_2 y \\ &\quad + (\alpha/\omega_1)(Ex^3 + Fx^2y + Gxy^2 + Hy^3) - (\alpha/\omega_1)p_2 \cos \tau = 0, \end{aligned} \quad (10)$$

where the dot denotes differentiation with respect to the scaled time parameter $\tau = \omega t$. Equations (9, 10) are equivalent to the Chen and Cheung equations, but they are set in a way which is more suitable to successive automatic manipulations.

A single command can expand the previous equations into a power series in α , and the coefficients of like powers can be immediately extracted. For example, the following line expands the first equation up to the quadratic α terms,

$$sseq1 = \text{Series}[eq1, \{\alpha, 0, 3\}], \quad (11)$$

and the coefficients will be given by

$$\begin{aligned} ss0eq1 &= \text{Coefficient}[sseq1, \alpha, 0], \\ ss1eq1 &= \text{Coefficient}[sseq1, \alpha, 1], \\ ss2eq1 &= \text{Coefficient}[sseq1, \alpha, 2]. \end{aligned} \quad (12)$$

A similar set of commands should be used to isolate the coefficients in the second equation, obtaining the quantities $ss0eq2$, $ss1eq2$ and $ss2eq2$.

The first order perturbation can be solved with the aid of the command

$$\begin{aligned} \text{lin} &= \text{DSolve}[\{ss0eq1 = = 0, ss0eq2 = = 0, \\ &\quad x0[0] = = x00, x0'[0] = = 0, y0[0] = = y00, y0'[0] = = 0\}, \\ &\quad \{x0[t], y0[t]\}, t] \end{aligned} \quad (13)$$

which finds the functions $x_0(t)$ and $y_0(t)$ with the initial conditions:

$$x_0(0) = x_{00}, \quad \dot{x}_0(0) = 0, \quad y_0(0) = y_{00}, \quad \dot{y}_0(0) = 0. \quad (14)$$

These solutions can be inserted into the coefficients of the linear part of the expansion, `ss1eq1` and `ss1eq2`.

Care should be taken, in order to obtain the desired results, because powers of the cosine must be avoided. The following command can do the job:

$$\text{ss1eq1} = \text{Simplify}[\text{ComplexToTrig}[\text{Expand}[\text{ss1eq1}, \text{Trig} \rightarrow \text{True}]]], \quad (15)$$

with a similar line for the other coefficient. The software automatically uses the previously obtained results, so that the user must merely manipulate the various terms. The outputs to the previous commands are given by

$$\begin{aligned} \text{ss1eq1} = & (1/4\omega_1)/[(3Ax_{00}^3 + x_{00}^2y_{00}B + 2x_{00}y_{00}^2C - 4x_{00}\omega_1 - 4p_1) \cos(t) \\ & + (Ax_{00}^3 + 2x_{00}^2y_{00}B + 3y_{00}^3D) \cos(3t) + (x_{00}^2y_{00}B + x_{00}y_{00}^2C) \cos(5t) \\ & + x_{00}y_{00}^2C \cos(7t) + y_{00}^3D \cos(9t)] + x_1(t) + \dot{x}_1(t), \end{aligned} \quad (16)$$

$$\begin{aligned} \text{ss2eq1} = & (1/4\omega_1)[(3x_{00}^3E + x_{00}^2y_{00}F + 2x_{00}y_{00}^2G - 4p_2) \cos(t) \\ & + (x_{00}^3E + 2x_{00}^2y_{00}F + 3y_{00}^3H - 36y_{00}\omega_1 + 4y_{00}\sigma_2) \cos(3t) \\ & + (x_{00}^2y_{00}F + x_{00}y_{00}^2G) \cos(5t) + x_{00}y_{00}^2G \cos(7t) \\ & + y_{00}^3H \cos(9t)] + 9y_1(t) + \dot{y}_1(t). \end{aligned} \quad (17)$$

The secular terms can now be easily extracted, because the form of the previous formulae is well adapted. One has

$$\text{coefcoseq1} = \text{Coefficient}[\text{Numerator}[\text{ss1eq1}], \text{Cos}[t]], \quad (18)$$

$$\text{coefcoseq2} = \text{Coefficient}[\text{Numerator}[\text{ss1eq2}], \text{Cos}[3 t]]. \quad (19)$$

From the first output, ω_1 can be found, from

$$\text{Solve}[\text{coefcoseq1} = 0, \text{omega1}], \quad (20)$$

whereas the second line gives the cubic relationship which is equivalent to equation (3.6) in reference [1]. It is

$$\omega_1 = (1/4x_{00})[3Ax_{00}^3 + x_{00}^2y_{00}B + 2x_{00}y_{00}^2C - 4p_1] \quad (21)$$

$$y_{00}^3(3H - 18C) - 9x_{00}y_{00}^2B + y_{00}(2x_{00}^2F - 27Ax_{00}^2 + 36p_1/x_{00}) + 4\sigma_2 + x_{00}^3E = 0. \quad (22)$$

The second order perturbation problem can be solved with the same steps. If the initial conditions are

$$x_1(0) = 0, \quad \dot{x}_1(0) = 0, \quad y_1(0) = y_{10}, \quad \dot{y}_1(0) = 0, \quad (23)$$

then a simple application of `DSolve` (see equation (13)) gives

$$\begin{aligned} x_1(t) = & K_{11} \cos(t) + K_{13} \cos(3t) + K_{15} \cos(5t) + K_{17} \cos(7t) + K_{19} \cos(9t), \\ y_1(t) = & K_{21} \cos(t) + K_{23} \cos(3t) + K_{25} \cos(5t) + K_{27} \cos(7t) + K_{29} \cos(9t), \end{aligned} \quad (24)$$

with:

$$K_{11} = (1/4\omega_1) \left(\frac{Ax_{00}^3}{8} + \frac{7x_{00}^2 y_{00} B}{24} + \frac{x_{00} y_{00}^2 C}{16} + \frac{31y_{00}^3 D}{80} \right),$$

$$K_{13} = - (1/4\omega_1) \left(\frac{Ax_{00}^3}{8} + \frac{x_{00}^2 y_{00} B}{4} + \frac{3y_{00}^3 D}{8} \right),$$

$$K_{15} = - (1/4\omega_1) \left(\frac{x_{00}^2 y_{00} B}{24} + \frac{x_{00} y_{00}^2 C}{24} \right), \quad K_{17} = - (1/4\omega_1) \frac{x_{00} y_{00}^2 C}{48},$$

$$K_{19} = - (1/4\omega_1) \frac{y_{00}^3 D}{80}, \quad (25)$$

$$K_{21} = (1/4\omega_1) \left(\frac{3x_{00}^3 E}{8} + \frac{x_{00}^2 y_{00} F}{8} + \frac{x_{00} y_{00}^2 G}{4} - p_2/2 \right),$$

$$K_{23} = (1/4\omega_1) \left(4\omega_1 y_{10} - \frac{3x_{00}^3 E}{8} - \frac{x_{00}^2 y_{00} F}{16} - \frac{13x_{00} y_{00}^2 G}{80} + \frac{y_{00}^3 H}{72} + p_2/2 \right),$$

$$K_{25} = - (1/4\omega_1) \left(\frac{x_{00}^2 y_{00} F}{16} + \frac{x_{00} y_{00}^2 G}{16} \right), \quad K_{27} = - (1/4\omega_1) \frac{x_{00} y_{00}^2 G}{40},$$

$$K_{29} = - (1/4\omega_1) \frac{y_{00}^3 H}{72}, \quad (26)$$

after the results have been expanded and simplified.

Because of the initial conditions,

$$K_{11} + K_{13} + K_{15} + K_{17} + K_{19} = 0, \quad K_{21} + K_{23} + K_{25} + K_{27} + K_{29} = y_{10}. \quad (27)$$

It is interesting to note that the previously obtained formulae resemble equation (3.7) in reference [1], but now the $\cos(t)$ coefficient in the first equation is not equal to zero. It is also worth noting that the above results should merely be introduced in the quadratic terms of the expansion, and therefore they can also not be displayed at all.

The following pair of commands perform a job similar to equation (15):

$$\text{ss2eq1} = \text{Simplify}[\text{ComplexToTrig}[\text{Expand}[\text{ss2eq1}, \text{Trig} \rightarrow \text{True}]]], \quad (28)$$

$$\text{ss2eq2} = \text{Simplify}[\text{ComplexToTrig}[\text{Expand}[\text{ss2eq2}, \text{Trig} \rightarrow \text{True}]]], \quad (29)$$

whereas the lines

$$\text{coef2coseq1} = \text{Coefficient}[\text{Numerator}[\text{ss2eq1}], \text{Cos}[t]], \quad (30)$$

$$\text{coef2coseq2} = \text{Coefficient}[\text{Numerator}[\text{ss2eq2}], \text{Cos}[3 t]], \quad (31)$$

extract the secular terms. They are

$$\begin{aligned} \text{coef2coseq1} = & (1/4\omega_1)[K_{11}(9Ax_{00}^2 + 2x_{00}y_{00}B + 2y_{00}^2C - 4\omega_1) \\ & + K_{13}[3Ax_{00}^2 + 4x_{00}y_{00}B] + K_{15}(2x_{00}y_{00}B + y_{00}^2C) + y_{00}^2K_{17}C \\ & + K_{21}(3x_{00}^2B + 2x_{00}y_{00}C + 6y_{00}^2D) + K_{23}(x_{00}^2B + 4x_{00}y_{00}C) \\ & + K_{25}(2x_{00}y_{00}C + 3y_{00}^2D) + 3y_{00}^2K_{27}D] - x_{00}\delta_2, \end{aligned} \quad (32)$$

$$\begin{aligned} \text{coef2coseq2} = & (1/4\omega_1)[K_{11}(3x_{00}^2E + 4x_{00}y_{00}F) + K_{13}(6x_{00}^2E + 3y_{00}^2G) \\ & + K_{15}(3x_{00}^2E + 2x_{00}y_{00}F) + 2x_{00}y_{00}K_{17}F + y_{00}^2K_{19}G \\ & + K_{21}(x_{00}^2F + 4x_{00}y_{00}G) + K_{23}(2x_{00}^2F + 9y_{00}^2H - 36\omega_1 + 4\sigma_2) \\ & + K_{25}(x_{00}^2F + 2x_{00}y_{00}G) + 2x_{00}y_{00}K_{27}G + 3y_{00}^2K_{29}H] - 9y_{00}\delta_2, \end{aligned} \quad (33)$$

Finally, the unknowns δ_2 and y_{10} can be calculated by means of

$$\text{Solve}\{\{\text{coef2coseq1} = = 0, \text{coef2coseq2} = = 0\}, \{\text{delta2}, y_{10}\}\}. \quad (34)$$

The pattern is now obvious. Each perturbation step can be performed with the following lines:

$$\begin{aligned} & \text{DSolve}\{\{\text{ssieq1} = = 0, \text{xi}[0] = = 0, \text{xi}'[0] = = 0\}, \text{xi}[t], t\}, \\ & \text{DSolve}\{\{\text{ssieq2} = = 0, \text{yi}[0] = = y_{1i}, \text{yi}'[0] = = 0\}, \text{yi}[t], t\}, \end{aligned} \quad (35)$$

$$\begin{aligned} \text{ssjeq1} = & \text{Simplify}[\text{ComplexToTrig}[\text{Expand}[\text{ssjeq1}, \text{Trig} \rightarrow \text{True}]]], \\ \text{ssjeq2} = & \text{Simplify}[\text{ComplexToTrig}[\text{Expand}[\text{ssjeq2}, \text{Trig} \rightarrow \text{True}]]], \end{aligned} \quad (36)$$

$$\begin{aligned} \text{coefjcoseq1} = & \text{Coefficient}[\text{Numerator}[\text{ssjeq1}], \text{Cos}[t]], \\ \text{coefjcoseq2} = & \text{Coefficient}[\text{Numerator}[\text{ssjeq1}], \text{Cos}[3t]], \end{aligned} \quad (37)$$

$$\text{Solve}\{\{\text{coefjcoseq1} = = 0, \text{coefjcoseq2} = = 0\}, \{\text{delta}j, y_{1j}\}\}. \quad (38)$$

Here j should be equal to $i + 1$. A more general approach can also be used, introducing some arrays and matrices, but it is probably useless, because the number of perturbation steps is always rather limited.

3. THE CASE OF ω NEAR ω_{20}

With some minor changes, the above described procedure can also be applied to case (b), where $\omega \approx \omega_{20}$. First of all, the equations of motion become

$$\begin{aligned} eq1 = & (1 + \delta_2\alpha^2 + \delta_3\alpha^3 + \dots)\ddot{x} + (1 - \alpha)(\lambda_1^2/\lambda_2^2)x + (\alpha/\omega_1)\sigma_1x \\ & + (\alpha/\omega_1)(Ax^3 + Bx^2y + Cxy^2 + Dy^3) - (\alpha/\omega_1)p_1 \cos \tau = 0, \end{aligned} \quad (39)$$

$$\begin{aligned} eq2 = & (1 + \delta_2\alpha^2 + \delta_3\alpha^3 + \dots)\ddot{y} + (1 - \alpha)y + (\alpha/\omega_1)(Ex^3 + Fx^2y + Gxy^2 + Hy^3) \\ & - (\alpha/\omega_1)p_2 \cos \tau = 0. \end{aligned} \quad (40)$$

The coefficients of the linear part of the expansion, $ss1eq1$ and $ss1eq2$ can be calculated with the same commands as for case (a). They are

$$\begin{aligned} ss1eq1 = & (1/36\omega_1)[(27Ax_0^3 + 9x_0^2y_0B + 18x_0y_0^2C - 4x_0\omega_1 + 36x_0\sigma_1) \cos(t/3) \\ & + (9Ax_0^3 + 18x_0^2y_0B + 27y_0^3D - 36p_1) \cos(t) \\ & + (9x_0^2y_0B + 9x_0y_0^2C) \cos((5t)/3) + 9x_0y_0^2C \cos((7t)/3) + 9y_0^3D \cos(3t)] \\ & + \frac{1}{9}x_1(t) + \ddot{x}_1(t), \end{aligned} \quad (41)$$

$$\begin{aligned} ss1eq2 = & (1/4\omega_1)[(3x_0^3E + x_0^2y_0F + 2x_0y_0^2G) \cos(t/3) \\ & + (x_0^3E + 2x_0^2y_0F + 3y_0^3H - 4y_0\omega_1 - 4p_2) \cos(t) + (x_0^2y_0F + x_0y_0^2G) \\ & \times \cos((5t)/3) + x_0y_0^2G \cos(7t/3) + y_0^3H \cos(3t)] + y_1(t) + \ddot{y}_1(t). \end{aligned} \quad (42)$$

The secular terms can be extracted as follows (see equations (18, 19))

$$coefcoseq1 = \text{Coefficient}[\text{Numerator}[ss1eq1], \text{Cos}[t/3]] \quad (43)$$

$$coefcoseq2 = \text{Coefficient}[\text{Numerator}[ss1eq2], \text{Cos}[t]] \quad (44)$$

Equation (20) allows one to find ω_1 :

$$\omega_1 = \frac{9}{4}[3Ax_0^2 + x_0y_0B + 2y_0^2C + 4\sigma_1], \quad (45)$$

whereas the cubic relationship for y_0 is given by

$$y_0^3(3H - 18C) - 9y_0^2x_0B - y_0(27Ax_0^2 - 2x_0^2F + 36\sigma_1) - 4p_2 + x_0^3E = 0. \quad (46)$$

Once again, $x_1(t)$ and $y_1(t)$ can be obtained by means of simple manipulations, giving:

$$x_1(t) = K_{11} \cos(t/3) + K_{13} \cos(t) + K_{15} \cos(5t/3) + K_{17} \cos(7t/3) + K_{19} \cos(3t),$$

$$y_1(t) = K_{21} \cos(t/3) + K_{23} \cos(t) + K_{25} \cos(5t/3) + K_{27} \cos(7t/3) + K_{29} \cos(3t), \quad (47)$$

where:

$$K_{11} = \frac{-9Ax_0^3}{32\omega_1} - \frac{21x_0^2y_0B}{32\omega_1} - \frac{9x_0y_0^2C}{64\omega_1} - \frac{279y_0^3D}{320\omega_1} + \frac{9p_1}{8\omega_1},$$

$$K_{13} = \frac{9Ax_0^3}{32\omega_1} + \frac{9x_0^2y_0B}{16\omega_1} + \frac{27y_0^3D}{32\omega_1} - \frac{9p_1}{8\omega_1},$$

$$K_{15} = \frac{3x_0^2y_0B}{32\omega_1} + \frac{3x_0y_0^2C}{32\omega_1}, \quad K_{17} = \frac{3x_0y_0^2C}{64\omega_1}, \quad K_{19} = \frac{9y_0^3D}{320\omega_1},$$

and:

$$K_{21} = \frac{-27x_0^3E}{32\omega_1} - \frac{9x_0^2y_0F}{32\omega_1} - \frac{9x_0y_0^2G}{16\omega_1},$$

$$K_{23} = y_{10} + \frac{27x_0^3E}{32\omega_1} + \frac{9x_0^2y_0F}{64\omega_1} + \frac{117x_0y_0^2G}{320\omega_1} - \frac{y_0^3H}{32\omega_1},$$

$$K_{25} = \frac{9x_0^2y_0F}{64\omega_1} + \frac{9x_0y_0^2G}{64\omega_1}, \quad K_{27} = \frac{9x_0y_0^2G}{160\omega_1}, \quad K_{29} = \frac{y_0^3H}{32\omega_1}. \quad (49)$$

Even in this case the results coincide with the Chen and Cheung expressions, apart from the coefficient of $\cos(t)$ in the first equation.

Equations (28, 29) express the quadratic part of the expansion in a form which is suitable for the extraction of the secular terms, according to the lines

$$\text{coef2coseq1} = \text{Coefficient}[\text{Numerator}[\text{ss2eq1}], \text{Cos}[t/3]], \quad (50)$$

$$\text{coef2coseq2} = \text{Coefficient}[\text{Numerator}[\text{ss2eq2}], \text{Cos}[t]]. \quad (51)$$

They are

$$\begin{aligned} \text{coef1coseq2} = & K_{11}(81Ax_{00}^2 + 18x_{00}y_{00}B + 18y_{00}^2C - 4\omega_1 + 36\sigma_1) \\ & + K_{13}(27Ax_{00}^2 + 36x_{00}y_{00}B) + K_{15}(18x_{00}y_{00}B + 9y_{00}^2C) \\ & + K_{17}9y_{00}^2C + K_{21}(27x_{00}^2B + 18x_{00}y_{00}C + 54y_{00}^2D) \\ & + K_{23}(9x_{00}^2B + 36x_{00}y_{00}C) + K_{25}(18x_{00}y_{00}C + 27y_{00}^2D) \\ & + K_{27}27y_{00}^2D - 4x_{00}\delta_2\omega_1. \end{aligned} \quad (52)$$

and

$$\begin{aligned} \text{coef2coseq2} = & K_{11}(3x_{00}^2E + 4x_{00}y_{00}F) + K_{13}(6x_{00}^2E + 3y_{00}^2G) \\ & + K_{15}(3x_{00}^2E + 2x_{00}y_{00}F) + K_{17}2x_{00}y_{00}F + K_{19}y_{00}^2G \\ & + K_{21}(x_{00}^2F + 4x_{00}y_{00}G) + K_{23}(2x_{00}^2F - 4\omega_1) \\ & + K_{25}(x_{00}^2F + 2x_{00}y_{00}G) + 2K_{27}x_{00}y_{00}G + 3K_{29}y_{00}^2H - 4y_{00}\delta_2\omega_1. \end{aligned} \quad (53)$$

4. NUMERICAL RESULTS

In order to perform a comparison between our results and the applications reported in reference [1], one has to adopt the following values for the coefficients:

$$\begin{aligned} \omega_{10} = 1, \quad \omega_{20} = 3, \quad A = 0.2788, \quad B = -0.3111, \quad C = 1.116, \quad D = -0.3864, \\ E = B/3, \quad F = C, \quad G = 3D, \quad H = 3.8703. \end{aligned} \quad (54)$$

Moreover, as already said, the results given in reference [1] can be obtained by putting $K_{11} = 0$. A first comparison can be performed for the free vibration analysis, $p_1 = p_2 = 0$, and in Table 1 the frequency of the response is given as a function of the initial amplitude x_{00} . As can be immediately seen, the discrepancies are very small, at least for small amplitudes. A forced vibration analysis is also performed, for $p_1 = 1$, $p_2 = 0$ and $\omega_0 = \omega_{10}$, as reported in Table 2. If the initial amplitude is reasonably small, the agreement can be

TABLE 1

Free frequency response of the system for various initial amplitude x_{00} values, as given by the authors (first column) and by [1] (second column).

A_{10}	ω		A_{10}	ω	
0.1	1.00105	1.00105	3	1.66428	1.69641
0.5	1.02588	1.02599	3.5	1.84159	1.88499
1	1.09866	1.10011	4	2.02634	2.08137
1.5	1.20776	1.21323	4.5	2.21677	2.28355
2	1.34301	1.35543	5	2.41159	2.49012
2.5	1.49691	1.51849			

TABLE 2

Forced frequency response of the system ($p_1 = 1$, $p_2 = 0$) for various initial amplitude x_{00} values, as given by the authors (first column) and by reference [1] (second column)

A_{10}	ω		A_{10}	ω	
-8	3.65296	3.79746	0.5	1.82131	1.82151
-7.5	3.44763	3.58062	1	1.57164	1.63133
-7	3.24405	3.36539	1.5	1.34271	1.36704
-6.5	3.04260	3.15215	2	1.14269	1.16579
-6	2.84378	2.94138	2.5	1.34746	1.38134
-5.5	2.64827	2.73376	3	1.56984	1.59519
-5	2.45695	2.53018	3.5	1.75505	1.80745
-4.5	2.27106	2.33192	4	1.95741	2.02018
-4	2.09232	2.14078	4.5	2.16064	2.23410
-3.5	1.9232	1.95945	5	2.16064	2.44937
-3	1.76738	1.79198	5.5	2.57069	2.66596
-2.5	1.63059	1.64477	6	2.7775	2.88373
-2	1.52246	1.52840	6.5	2.98535	3.10255
-1.5	1.46099	1.46188	7	3.19414	3.32230
-1	1.48707	1.4863	7.5	3.40376	3.54285
-0.5	1.74746	1.74708	8	3.61412	3.76409

considered satisfactory. Nevertheless, if x_{00} is large enough, the difference can become noticeable (up to 4%).

All the numerical values have been generated by means of the simple *Mathematica* notebook reported in the Appendix. Case (b) can be treated with some minor changes.

5. CONCLUSIONS

A recently proposed modified Lindstedt–Poincaré perturbation method has been proved to be particularly suited to automatic symbolic analysis. A two-degree-of-freedom oscillator has been used as an example, in the presence of internal resonances and general cubic non-linearities. The numerical examples show quite a good agreement with some results from the literature [1].

REFERENCES

1. S. H. CHEN and Y. K. CHEUNG 1996 *Journal of Sound and Vibration* **193**, 751–762. A modified Lindstedt–Poincaré method for a strongly non-linear two-degree-of-freedom system.
2. G. GREEN 1837 *Transactions of the Cambridge Philosophical Society* **6**, 457–462. On the motion of waves in a variable canal of small depth and width.
3. J. LIOUVILLE 1837 *Journal de Mathématiques Pures et Appliquées* **2**, 16–35. Sur le développement des fonctions ou parties de fonctions en séries.
4. J. KEVORKIAN and J. D. COLE 1981 *Perturbation Methods in Applied Mathematics*. Springer.
5. E. J. HINCH 1997 *Perturbation Methods*. Cambridge University Press.
6. M. H. HOLMES 1995 *Introduction to Perturbation Methods*. Springer.
7. P. G. DRAZIN 1994 *Nonlinear Systems*. Cambridge University Press.
8. P. J. MELVIN 1977 *SIAM Journal of Applied Mathematics* **33**, 161–194. On the construction of Poincaré–Lindstedt solutions: the nonlinear oscillator equation.

9. A. H. NAYFEH and D. T. MOOK 1979 *Nonlinear Oscillations*. J. Wiley.
10. N. N. BOGOLIUBOV and Y. A. MITROPOLSKY 1961 *Asymptotic Methods in the Theory of Nonlinear Oscillations*. Gordon and Breach.
11. P. YU, Y. M. DESAI, N. POPPLEWELL and A. H. SHAH 1996 *Journal of Sound and Vibration* **192**, 413–438. The Krylov–Bogoliubov and Galerkin methods for non-linear oscillations.
12. A. H. NAYFEH and B. BALACHANDRAN 1989 *Applied Mechanics Review* **42**, 175–201. Modal interactions in dynamical and structural systems.
13. J. F. GEER and C. M. ANDERSEN 1990 *SIAM Journal of Applied Mathematics* **50**, 1474–1495. A hybrid perturbation-galerkin technique that combines multiple expansions.
14. S. E. JONES 1978 *International Journal Non-linear of Mechanics* **13**, 125–141. Remarks on the perturbation process for certain conservative systems.
15. T. D. BURTON 1984 *International Journal of Non-linear Mechanics* **19**, 397–407. A perturbation method for certain non-linear oscillators.
16. T. D. BURTON and Z. RAHMAN 1986, **21(2)**, 135–146. On the multi-scale analysis of strongly non-linear forced oscillators.
17. Y. K. CHEUNG, S. H. CHEN and S. L. LAU 1991 *International Journal of Non-linear Mechanics* **26**, 367–378. A modified Linstedt–Poincaré method for certain strongly non-linear oscillators.
18. P. HAGEDORN 1982 *Nonlinear Oscillations*. Clarendon Press.
19. R. H. RAND and D. ARMBRUSTER 1987 *Perturbation Methods, Bifurcation Theory and Computer Algebra*. Springer.
20. A. HECK 1993 *Introduction to Maple*. Springer.
21. S. WOLFRAM 1991 *Mathematica: A System for Doing Mathematics by Computer*. Addison–Wesley.

APPENDIX: THE MATHEMATICA NOTEBOOK

```

Off[General::spell, General::spell1 ]
<<Algebra 'Trigonom'
sintras = Sin [t_] -> Sqrt[1 - Cos [t] Cos [t]];
x[t] = x0[t] + x1[t]*e + x2[t]*e^2;
y[t] = y0[t] + y1[t]*e + y2[t]*e^2;
eq1 = (1 + delta2*e^2)*
  D[x[t], {t, 2}] + (1 - e)*x[t] + e/omega1*(AA*x[t]^3 +
  BB*x[t]^2*y[t] + CC*x[t]*y[t]^2 + DD*y[t]^3) -
  e/omega1*p1*Cos[t];
eq2 = (1 + delta2*e^2)*
  D[y[t], {t, 2}] + 9*(1 - e)*y[t] + e/omega1*ss2*y[t] +
  e/omega1*(EE*x[t]^3 + FF*x[t]^2*y[t] + GG*x[t]*
  y[t]^2 + HH*y[t]^3) - e/omega1*p2*Cos[t];
sseq1[t] = Series[eq1, {e, 0, 2}];
sseq2[t] = Series[eq2, {e, 0, 2}];
ss0eq1[t] = Simplify[Coefficient[sseq1[t], e, 0]];
ss0eq2[t] = Simplify[Coefficient[sseq2[t], e, 0]];
ss1eq1[t] = Simplify[Coefficient[sseq1[t], e, 1]];
ss1eq2[t] = Simplify[Coefficient[sseq2[t], e, 1]];
ss2eq1[t] = Simplify[Coefficient[sseq1[t], e, 2]];
ss2eq2[t] = Simplify[Coefficient[sseq2[t], e, 2]];

lin = DSolve[{ss0eq1[t] == 0, ss0eq2[t] == 0,
  x0[0] == x00, x0'[0] == 0, y0[0] == y00, y0'[0] == 0},
  {x0[t], y0[t]}, t];
x0[t] = x0[t]/. lin; y0[t] = y0[t]/. lin;
x0''[t] = D[x0[t], {t, 2}]; y0''[t] = D[y0[t], {t, 2}];

```

```

ss1eq1[t] = Simplify[ComplexToTrig[Expand[
    ss1eq1[t], Trig→True]];
ss1eq2[t] = Simplify[ComplexToTrig[Expand[
    ss1eq2[t], Trig→True]];
coefcoseq1 = Coefficient[Numerator[ss1eq1[t]], Cos[t]];
coefcoseq2 = Coefficient[Numerator[ss1eq2[t]], Cos[3*t]];

Solve[coefcoseq1 == 0, omega1];
omega1 = Simplify[omega1/.%];
eqA20 = Simplify[coefcoseq2];
ss1eq1[t] = Simplify[ss1eq1[t]];
ss1eq2[t] = Simplify[ss1eq2[t]];
ddd = Simplify[4*omega1];
rhs = Coefficient[Numerator[ss1eq1[t]], Cos [3 t]]*
    Cos [3 t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [5 t]]*
    Cos [5 t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [7 t]]*
    Cos [7 t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [9 t]]*
    Cos [9 t];
DSolve[{x1''[t] + x1[t] == cc1*rhs[[1, 1]],
    x1[0] == 0, x1'[0] == 0}, x1[t], t];
quad1 = Simplify[%, Trig→True];
rhs = Coefficient[Numerator[ss1eq2[t]], Cos [t]]*
    Cos [t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [5 t]]*
    Cos [5 t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [7 t]]*
    Cos [7 t] +
    Coefficient[Numerator[ss1eq1[t]], Cos [9 t]]*
    Cos [9 t];
DSolve[{y1''[t] + 9*y1[t] == cc1*rhs[[1, 1]],
    y1[0] == y10, y1'[0] == 0}, y1[t], t];
quad2 = Simplify[%, Trig→True];
x1[t] = x1[t] /. quad1; y1[t] = y1[t] /. quad2;
x1'[t] = D[x1[t], t]; x1''[t] = D[x1[t], {t, 2}];
y1'[t] = D[y1[t], t]; y1''[t] = D[y1[t], {t, 2}];
x1[t] /. sintras; y1[t] /. sintras;
x1[t] = Expand[x1[t], Trig→True] /. cc1 → -1/ddd;
y1[t] = Expand[y1[t], Trig→True] /. cc1 → -1/ddd;
ss2eq1[t] = Simplify[ComplexToTrig[Expand[
    ss2eq1[t], Trig→True]];
ss2eq2[t] = Simplify[ComplexToTrig[Expand[
    ss2eq2[t], Trig→True]];
coef2coseq1 = Collect[Coefficient[
    Numerator[ss2eq1[t]], Cos [t]], delta2]
coef2coseq2 = Coefficient[
    Numerator[ss2eq2[t]], Cos[3*t]]
AA = .2788; BB = -0.3111; CC = 1.116; DD = -0.3864; HH = 3.8703;
GG = 3*DD; FF = CC; EE = BB/3; p1 = 1; p2 = 0; ss2 = 0;

```

```

Do[
tt1 = NSolve[eqA20 == 0, y00, 30]/. x00→Amp10;
Amp20 = y00/.tt1[[1]];
ss1 = N[omega1,20]/. {x00→Amp10, y00→Amp20};
tt2 = NSolve[{coef2coseq1 == 0, coef2coseq2 == 0}, {delta2, y10},
30]/. {x00→Amp10, y00→Amp20};
alpha = omega1/(1 + omega1)/. {x00→Amp10, y00→Amp20};
omega2 = 1/(1 - alpha)*(1 + delta2*alpha^2)/.tt2;
Print[StringForm["Initial amplitude = \"", Amp10]];
Print[StringForm["Second mode amplitude = \"", Amp20]];
Print[StringForm["Omega1 = \"", ss1]];
Print[StringForm["Frequency = \"", Sqrt[omega2]]];
Print[StringForm["delta2 = \"", tt2[[1, 2]]]];
Print[StringForm["Amplitude y10 = \"", tt2[[1, 1]]]];
Print[StringForm["Amplitude y00 = \"", tt1]];
Print[""],
{Amp10, -8, 8, .1}]

```