Reprinted from "Computer Aided Proofs in Analysis", eds. K.R.Meyer and D.S.Schmidt, pp.71-89, Springer-Verlag, 1991

# MACSYMA PROGRAM TO IMPLEMENT AVERAGING USING ELLIPTIC FUNCTIONS

VINCENT T. COPPOLA AND RICHARD H. RAND*

**Abstract.** The method of averaging is applied to the system:

$$x'' + \alpha(\tau)\, x + \beta(\tau)\, x^3 + \epsilon\, g(x, x', \tau) = 0$$

where $\tau = \epsilon t$ is slow time, and where $\epsilon << 1$. This involves the laborous manipulation of Jacobian elliptic functions, a process which is most easily and accurately accomplished using computer algebra. We present the listing of a MACSYMA program which implements the method to $0(\epsilon)$, as well as the results of a run for which $g(x, x', \tau)$ is taken as a general cubic polynomial in $x$ and $x'$.

**Introduction.** In this paper we treat a class of problems which involve perturbing off of the Jacobian elliptic function solution of the strongly nonlinear oscillator

$$x'' + \alpha\, x + \beta\, x^3 = 0 \tag{1}$$

In particular, we investigate the following nonautonomous perturbation of eq. (1):

$$x'' + \alpha(\tau)\, x + \beta(\tau)\, x^3 + \epsilon\, g(x, x', \tau) = 0 \tag{2}$$

where $\tau$ represents "slow time",

$$\tau = \epsilon\, t, \quad \epsilon << 1 \tag{3}$$

and where primes represent differentiation with respect to t. The functions $\alpha(\tau)$, $\beta(\tau)$, and $g(x, x', \tau)$ must be specified for a particular problem but are otherwise arbitrary.

We shall use the method of averaging implemented on the computer algebra system MACSYMA to obtain approximate equations governing the solutions to eq. (2).

Although the method of averaging has been treated in numerous references (e.g. [13-15, 17-19, 21-23]), most treatments deal almost exclusively with perturbations off of the simple harmonic oscillator. A few authors have considered perturbations

*Department of Theoretical and Applied Mechanics, Cornell University, Ithaca, NY 14853

off of nonlinear systems using elliptic functions. Kuzmak [16] looks for periodic solutions in eq. (1) using a multiple scale method, where $\alpha$ and $\beta$ are slowly varying parameters. Garcia-Margallo and Bejarano [9] find limit cycles in a generalized van der Pol oscillator using generalized harmonic balance. Davis [8] investigates second order ordinary differential equations using elliptic functions. Cap [2] applies the method of averaging to perturbations of the mathematical pendulum. Pocobelli [20] studies the slowly varying pendulum. Greenspan and Holmes [12] and Guckenheimer and Holmes [13] apply the Melnikov method to perturbations of eq. (1) where $\alpha < 0$. Nayfeh [18], Kevorkian & Cole [15] and Sanders & Verhulst [22] also treat such problems.

Perturbations of eq. (1) with $\alpha = 0$ (a purely nonlinear oscillator) have appeared in the literature. Chirikov [3] studies resonance overlap under multiple harmonic excitations. Garcia-Margallo and Bejarano [10] employ generalized harmonic balance in order to approximate limit cycles. Yuste and Bejarano [24] use first order averaging as a means to find transitory behavior as the motion is attracted to a limit cycle.

In most of these references the authors have obtained general expressions for approximate equations of motion in terms of integrals. Although the problem is thus reduced to the evaluation of these integrals, little use has been made of these treatments because the evaluations require complicated algebraic manipulations of Jacobian elliptic functions. By using the computer algebra system MACSYMA, we have been able to efficiently evaluate the associated integrals.

**Variation of Parameters.** In preparation for the method of averaging, we use the method of variation of parameters to express the influence of the slowly varying and order $\epsilon$ terms in eq. (2) on the solution of the unperturbed eq. (1). In contrast to the method of averaging, the computations presented in this section are exact. However, the results are intractable and unenlightening. The method of averaging (introduced in the next section) replaces the results obtained in this section by more useful equations, which are, however, approximate (valid in the small $\epsilon$ limit.)

The unperturbed solution (the general solution to eq. (1) for $\tau$ fixed and $\epsilon = 0$) is expressed in terms of Jacobian elliptic functions. A brief summary of elliptic functions is given in Appendix A, which also gives the notation we use for elliptic functions and elliptic integrals. We are interested in systems (2) in which the unperturbed system (1) is oscillatory. It can be shown [4] that the unperturbed solution for such systems can be expressed as:

$$x = \mu\, r\, cn(u,k) \tag{3.1}$$

$$x' = -\mu\, r\, a\, sn(u,k)\, dn(u,k) \tag{3.2}$$

where

$$k^2 = \frac{\beta\, r^2}{2\, a^2},\; a^2 = \beta\, r^2 + \alpha,\; u = a\, t + u_o$$

$$r \geq 0,\; a \geq 0,\; \mu = \pm 1$$

In eqs. (3), $r$ and $u_o$ are arbitrary constants of integration. As usual in the method of variation of parameters, we look for a solution to eq. (2) in the form of eqs. (3), where the two arbitrary constants $r$ and $u_o$ are allowed to vary in time. This results in first order differential equations on $r(t)$ and $u_o(t)$. Unfortunately the resulting expression for $du_o(t)/dt$ is not periodic, and is thus unsuitable for averaging. This may be remedied by replacing $u_o$ by $\varphi$, the angle variable of action-angle variables [11], where

$$u = 4\, K(k)\, \varphi \tag{4}$$

The proportionality factor $4K(k)$ in eq. (4) takes into account the dependence of frequency on amplitude in eq. (1), thus eliminating "phase shear". The resulting equations on $r$ and $\varphi$ then become:

$$\frac{dr}{dt} = -\epsilon\, g \frac{sn\; dn}{\sqrt{\alpha + \beta\, r^2}} - \epsilon \frac{d\alpha}{d\tau} \frac{r\,(1 - cn^2)}{2\,(\alpha + \beta r^2)} - \epsilon \frac{d\beta}{d\tau} \frac{r^3\,(1 - cn^4)}{4\,(\alpha + \beta\, r^2)} \tag{5.1}$$

$$\frac{d\varphi}{dt} = \frac{\sqrt{\alpha + \beta\, r^2}}{4\, K} \tag{5.2}$$

$$+\,\epsilon\, g \frac{2\,\alpha\,(\alpha + \beta\, r^2)\,(Z\; sn\; dn + cn) + \alpha\,\beta\, r^2\, cn^3 + \beta^2\, r^4\, cn}{4\, K\, r\,(\alpha + \beta\, r^2)^{3/2}\,(2\,\alpha + \beta\, r^2)}$$

$$+\,\epsilon \frac{d\alpha}{d\tau} \frac{\alpha\, Z\, cn^2 - (2\,\alpha + \beta\, r^2)\, Z - \alpha\; cn\; sn\; dn}{4\, K\,(\alpha + \beta\, r^2)\,(2\,\alpha + \beta\, r^2)}$$

$$+\,\epsilon \frac{d\beta}{d\tau} \frac{\left[\alpha\,\beta\, r^2 (Z\, cn^4 + Z - cn\; sn\; dn\,(2 + cn^2)) + 2\,\alpha^2 Z - \beta^2 r^4 cn\; sn\; dn\right]}{8\, K\,\beta\,(\alpha + \beta\, r^2)\,(2\alpha + \beta\, r^2)}$$

where $cn = cn(4\, K\, \varphi, k)$, $sn = sn(4\, K\, \varphi, k)$, $dn = dn\,(4\, K\, \varphi, k)$, and $Z = Z(4\, K\, \varphi, k)$. Eqs. (5) are periodic in $\varphi$ and are thus in the proper form for averaging.

**The Method of Averaging.** In order to explain the method of averaging, we write eqs. (5) in the abbreviated form:

(6.1) $$r' = \epsilon\, F_1(r,\varphi)$$

(6.2) $$\varphi' = \Omega(r) + \epsilon\, F_2(r,\varphi)$$

The method is based on positing a near-identity transformation from $(r,\varphi)$ to $(\bar{r},\bar{\varphi})$:

(7.1) $$r = \bar{r} + \epsilon w_1(\bar{r},\varphi) + \epsilon^2\, v_1(\bar{r},\varphi) + O(\epsilon^3)$$

(7.2) $$\varphi = \bar{\varphi} + \epsilon\, w_2(\bar{r},\varphi) + O(\epsilon^2)$$

where the generating functions $w_1, v_1,$ and $w_2$ are to be chosen so as to simplify the resulting equations. Substituting eqs. (7) into eqs. (6) and co llecting terms gives equations of the form:

(8.1) $$\bar{r}' = \epsilon\left[F_1(\bar{r},\varphi) - \Omega(\bar{r})\,\frac{\partial w_1}{\partial\varphi}\right] + \epsilon^2\left[H_1(\bar{r},\varphi) - \Omega(\bar{r})\,\frac{\partial v_1}{\partial\varphi}\right] + O(\epsilon^3)$$

(8.2) $$\bar{\varphi}' = \Omega(\bar{r}) + \epsilon\left[F_2(\bar{r},\varphi) + \frac{d\Omega}{dr}(\bar{r}) - \Omega(\bar{r})\,\frac{\partial w_2}{\partial\varphi}\right] + O(\epsilon^2)$$

The transformation functions $w_1, w_2,$ and $v_1$ are chosen so that eqs. (8) are in averaged form, i.e.,

(9.1) $$\bar{r}' = \epsilon\,\bar{F}_1(\bar{r}) + \epsilon^2\,\bar{H}_1(\bar{r}) + O(\epsilon^3)$$

(9.2) $$\bar{\varphi}' = \Omega(\bar{r}) + \epsilon\,\bar{F}_2(\bar{r}) + O(\epsilon^2)$$

where $\bar{F}_1, \bar{F}_2$, and $\bar{H}_1$ are the means of $F_1, F_2$, and $H_1$ taken over one period in the periodic variable $\varphi$.

**Computer Algebra.** We have written a computer algebra (MACSYMA) program that implements the averaging procedure (6)-(9) for eqs. (5). The user first inputs expressions for $\alpha, \beta$ and $g$, which may contain symbolic parameters. The computer then generates $F_1, F_2$, and $\Omega$ from eqs. (5) (where eqs. (3) are substituted for $x$ and $x'$ in $g$). Using an elliptic function integration subroutine that we developed, the program finds $\bar{F}_1, \bar{F}_2, w_1$, and $w_2$ (which completes the first order averaging).

The listing for this first order averaging program is given in Appendix B, and a sample run is given in Appendix C.

This program was then extended to include second order averaging of the $r$ equation. We found it essential to proceed in several steps in order to prevent excessive intermediate expression swell. First, $H_1$ is computed and its terms are divided up among several pre-identified categories. These categories group together terms whose means are computed in like manners. The mean of $H_1$ is then computed category by category. After this second order averaging is completed, the program outputs the averaged system and the first order transformation.

The second order averaging program consists of 460 lines of code. Typical runs on a Symbolic 3670 computer take from one to six hours. For $\alpha = \beta = 1$ and $g$ consisting of three ter ms ($x', x\,x'^2,\ x'^3$), there are 497 second order terms to be averaged. For more information on the program, see [4 – 7] which contains many applications. Electronic copies of both programs are available from the authors via BITNET. Our current BITNET addresses are:

$DUGY@CRNLVAX5$ (for VTC), $RHRY@CRNLVAX5$ (for RHR)

**Results.**

In order to illustrate the use of the program, we present the results of applying it to eq. (2) when $g(x, x', \tau)$ takes the general form:

$$(10)\quad \begin{aligned} g(x,x',t) \;=\; & a_{00}(\tau) + a_{10}(\tau)\,x + a_{01}(\tau)\,x' + a_{20}(\tau)\,x^2 + a_{11}(\tau)\,x\,x' \\ & + a_{02}(\tau)\,x'^2 + a_{30}(\tau)\,x^3 + a_{21}(\tau)\,x^2\,x' + a_{12}(\tau)\,x\,x'^2 \\ & + a_{03}(\tau)\,x'^3 \end{aligned}$$

The averaged eqs. (9) become, to lowest order in $\epsilon$ (dropping the bars on $r$ and $\varphi$ for convenience):

$$(11.1)\qquad r' = \epsilon\,\frac{d\alpha}{d\tau}\,\frac{1}{\beta\,r}\,(\frac{E}{K} - 1) - \epsilon\,\frac{d\beta}{d\tau}\,\frac{1}{6\,\beta^2\,r}\left[\beta\,r^2 + 4\,\alpha\,(\frac{E}{K} - 1)\right]$$

$$+ \epsilon\, a_{01}(\tau)\, \frac{1}{3\,\beta\, r} \left[ 2\,\alpha\, (\frac{E}{K} - 1) - \beta\, r^2 \right]$$

$$-\epsilon\, a_{11}(\tau)\, \frac{\sqrt{2}\,\pi\, H(k^2-1)}{8\,K\,r} \left[ \frac{\alpha + \beta\, r^2}{\beta} \right]^{3/2}$$

$$+\epsilon\, a_{21}(\tau) \left[ \frac{8\,a^2}{15\,\beta^2\,r}\, (1 - \frac{E}{K}) + \frac{2\,\alpha\, r}{15\,\beta}\, (5 - 6\,\frac{E}{K}) + \frac{r^3}{5}\, (1 - 2\,\frac{E}{K}) \right]$$

$$+\epsilon\, a_{03}(\tau) \left[ \frac{8\,\alpha^3}{35\,\beta^2\,r}\, (\frac{E}{K} - 1) + \frac{2}{35}\,\frac{\alpha^2}{\beta}\, r\, (16\,\frac{E}{K} - 15) + \frac{\alpha\, r^3}{35}\, (16\,\frac{E}{K} - 23) - \frac{\beta}{7}\, r^5 \right]$$

(11.2) $$\varphi' = \frac{\sqrt{\alpha + \beta\, r^2}}{4\,K} + O(\epsilon)$$

where $H(k^2-1)$ is the Heaviside step function with argument $k^2 - 1$, i.e., $H(k^2 - 1) = 1$ when the system point has $k^2 > 1$ and $H(k^2-1) = 0$ otherwise. In (11.1), K and E respectively represent complete elliptic integral s of the first and second kinds.

The generality of the foregoing example may obviate the need for the pro gram in many cases. Consider, for example, the system

(12) $$x'' + x + x^3 + 0.035\, x' - 0.6\, x^2 x' + 0.1\, x'^3 = 0$$

This system may be cast in the form (2) by choosing $\epsilon = 0.1$ and

(13) $$\alpha(\tau) = 1,\ \beta(\tau) = 1,\ g(x, x', \tau) = x'\, (0.35 - 6\, x^2 + x'^2)$$

Then we find from eq. (11.1)

(14) $$r' = \epsilon \left[ \frac{769}{210}\, \frac{1}{r}\, (\frac{E}{K} - 1) + \frac{r}{420}\, (2400\, \frac{E}{K} - 2089) + \frac{r^3}{7}\, (20\, \frac{E}{K} - 13) - \frac{r^5}{7} \right]$$

Numerical evaluation of the right hand side of eq. (14) shows that it has two zeros at about $r = r_1 \simeq 1.12675$ and $r = r_2 \simeq 0.83984$, and thus that the original system (12) is predicted to have two limit cycles of approxim ate amplitudes $r_1$ and $r_2$. This result is in agreement with numerical integration of eq. (12).

It is interesting to note that the usual approach to averaging (based on perturbations off of the simple harmonic oscillator) fails to give correct predictions for this examp le. In order to use such an approach, we first write the example (12) in the form:

(15) $$x'' + x + \epsilon\,(\lambda\, x^3 + \delta\, x' + \rho\, x^2 x' + \eta\, x'^3) = 0$$

such that

(16) $$\epsilon\lambda = 1,\ \epsilon\delta = 0.035,\ \epsilon\rho = -06,\ \epsilon\eta = 0.1$$

Eq. (15) may be averaged using previously published computer algebra p rograms in [21]. This involves assuming a solution of the form

(17) $$x = r(t)\ cos\ [t + \theta(t)]$$

and results in the following slow flow for $r(t)$, valid to $0(\epsilon^2)$:

(18) $$r' = -\,\epsilon\,\frac{r}{8}\left[4\delta + (\rho + 3\eta)r^2\right] + \epsilon^2\,\frac{r^5}{32}\left[\lambda\,(\rho - 3\eta)\right] + O(\epsilon^3)$$

If we keep only $0(\epsilon)$ terms in (18), and use (16), we obtain

(19) $$r' = -\,\frac{r}{8}\,(0.14 - 0.3\,r^2)$$

which results in the incorrect prediction of only one limit cycle at $r \simeq 0.68$. If, on the other hand, we keep both $0(\epsilon)$ and $0(\epsilon^2)$ terms in (18), we obtain

(20) $$r' = -\,\frac{r}{8}\,(0.14 - 0.3\,r^2 + 0.225\,r^4)$$

the right hand side of which has no real roots. Thus both $0(\epsilon)$ and $0(\epsilon^2)$ approximations based on trigonometric averaging fail to give qualitatively correct limit cycle results for example (12).

## Appendix A: Jacobian Elliptic Functions

Jacobian elliptic functions involve a collection of identities which are similar to those for trigonometric functions but are more complicated algebraically. The use of computer algebra makes manipulation of these identities easier, permitting investigations to proceed on problems which were previously avoided because of the large quantities of algebra involved. All formulas and conventions concerning Jacobian elliptic functions in this paper are taken from Byrd and Friedman's Handbook of Elliptic Integrals for Engineers and Physicists [1].

For the reader's convenience, we offer a brief comparison of elliptic functions with the more familiar trigonometric functions. Corresponding to sin(u) and cos(u) are three fundamental elliptic functions sn(u,k), cn(u,k), and dn(u,k). Each of the elliptic functions depends on the modulus k as well as the argument u. These reduce to sin(u), cos(u), and 1 respectively, when k = 0. The sn and sin functions share common properties as do cn and cos. These are summarized in Table A. The dn function has no trigonometric counterpart. Note that the elliptic functions sn and cn may be thought of as generalizations of sin and cos where their period depends on the modulus k.

The argument u is identified as the incomplete elliptic integral of the first kind which is usually denoted $F(\theta, k)$. This identification shows that u also depends on k. The value of k normally ranges from 0 to 1, but we allow $k^2 \epsilon [-\infty, \infty]$. For the interpretation of the elliptic functions on this range, see [4] or [1].

Table A. Properties of Jacobi elliptic functions

| | Function f | | | | |
|---|---|---|---|---|---|
| Property | sn(u,k) | sin(u) | cn(u,k) | cos(u) | dn(u,k) |
| Max. value | 1 | 1 | 1 | 1 | 1 |
| Min. value | -1 | -1 | -1 | -1 | $(1-k^2)^{1/2}$ |
| Period | 4K(k) | $2\pi$ | 4 K(k) | $2\pi$ | 2 K(k) |
| Odd/Even | Odd | Odd | Even | Even | Even |
| df/du | cn dn | cos | -sn dn | -sin | $-k^2$ sn cn |
| $f\|k=0$ | sin | sin | cos | cos | 1 |
| $f\|k=1$ | tanh | — | sech | — | sech |

$$\text{K(k)} = \text{complete elliptic integral of the first kind}$$

$$\text{K}(0) = \pi/2, \ \text{K}(1) = +\infty$$

The elliptic functions also satisfy the following identities which correspond to $sin^2 + cos^2 = 1$:

$$sn^2 + cn^2 = 1 \tag{A1}$$

$$k^2\, sn^2 + dn^2 = 1 \tag{A2}$$

$$1 - k^2 + k^2\, cn^2 = dn^2 \tag{A3}$$

In addition to the sn, cn, and dn functions, there are three other frequently encountered elliptic functions. First, there is the amplitude function am(u,k) $(= \theta)$ which is the inverse of $F(\theta, k)$ $(=$ u). This function maps the elliptic argument u onto a trignometric argument $\theta$ so that the period 4 K(k) in u equals the period $2\pi$ in $\theta$.

Second, there is $E(\theta, k)$, the incomplete elliptic integral of the second kind. It is often written in abbreviated notation as E(u) since $\theta$ depends on u (via the am function) and the dependence on k is understood. Both E(u) and u are not periodic in u. The complete elliptic integral of the second kind is denoted E(k).

Finally, there is the Jacobi Zeta function $Z(\theta, k)$, a linear combination of E(u) and u:

$$Z(\theta, k) = E(\theta, k) - \frac{E(k)}{K(k)}\, F(\theta, k) \tag{A4}$$

This function is periodic in u with period 2 K(k) and has zero mean. It is often written in abbreviated notation as Z(u) in the same manner as E(u).

Appendix B: MACSYMA Program Listing

```
/* Averaging using Elliptic Functions */

/* x'' + alpha(tau) x + beta(tau) x^3 + e g(x,x',tau) = 0 */

/* where tau = eps t and g is polynomial in x and x' */

/* Variable names and their meanings */
/* */
/* X = variable in differential equation */
/* Y = dx/dt */
/* T = time */
/* TAU = eps*t */
/* G = g(x,x',tau) = g(x,y,tau) is a perturbation */
/* AL = alpha(tau) */
/* BE = beta(tau) */
/* RR = amplitude */
/* AA = instantaneous time frequency */
/* U0 = phase angle constant */
/* K = modulus of the elliptic functions */
/* U = argument of elliptic functions */
/* CN(U,K) = a Jacobi elliptic function */
/* CN'(U,K) = - sn(u,k) dn(u,k) = derivative of cn(u,k) w.r.t.  argument u */
/* ZZ = jacobi zeta function */
/* KC = complete elliptic integral of the first kind */
/* EC = complete elliptic integral of the second kind */
/* PHI = angle variable [where u = 4 kc phi] */
/* MU = +1 or -1 depending on whether the system point is within left(-1) */
/*     or right(+1) separtrix loop when k^2 > 1 */
/* WITHINSEP = H(k^2-1) where H is the Heaviside step function */
/*     = a flag telling whether an orbit is within the double */
/*     homoclinic loop separtrix (+1) or not (0) */

/* The 2nd order differential equation is written as 2 1st order equations:  */
/* */
/* x' = y */
/* y' = - eps g(x,y,tau) - alpha(tau) x - beta(tau) x^3 */

/* For alpha and beta fixed */
/* [and proper interpretation of elliptic variables */
/* for k between -infinity and infinity] */
/* the differential equation is solved exactly by:  */
/* */
/* x = mu rr cn(u,k) y = mu rr aa cn'(u,k) u = aa t + u0 */
/* aa^2 = al + be rr^2 k^2 = be rr^2/2/aa^2 */
/* */
/* where initial conditions determine the values of rr and u0 */

/* But, we must use slow flow equation for phi [rather than u0 or u] */
```

```
/* where initial conditions determine the values of rr and u0 */

/* But, we must use slow flow equation for phi [rather than u0 or u] */

/* The Variational equations to be averaged are:  */
/* */
/* diff(rr,t) = eps F[1] + O(eps^2) */
/* diff(phi,t) = aa/4/kc + eps F[2] + O(eps^2) */

/* Averaging uses a near-identity transformation as follows:  */
/* */
/* rr=rbar+eps*W[1](rbar,phi) */
/* phi=phibar+eps*W[2](rbar,phi) */

/* Symbols used in the computation */

/* XX = cn function */
/* YY = cn' function */
/* ZZ = Jacobian zeta function */
/* SS = arcsin(k*sn(u,k)) = arcsin(k*sqrt(1-xx^2)) */
/* S0 = integral of SS with respect to u */
/* S2 = integral of SS*XX^2 with respect to u */
/* UU = argument */
/* TH = LN(THETA(U)/THETA(0)) */

/* Symbols used on output */
/* */
/* CNF = cn elliptic function */
/* CNP = cn' elliptic function = - sn dn */
/* SNF = sn elliptic function */
/* ZETA = jacobi zeta function */
/* THETA = an elliptic theta function */
/* S0 = see above */
/* S2 = see above */

AVERAGE():=BLOCK([X,Y,XX,YY,F,FI,W,DPHIW,H,V,FBAR,W1MEAN,W1INT,W1C,K,AL,BE,
    RR,XX,YY,ZZ,SS,UU,TH,TAU,KC,EC,EPS,RBAR,KBAR,PHI,CNF,SNF,CNP,
    THETA,ZETA,ARCSIN],

/* Input problem */

PRINT("AVERAGING OF X'' + ALPHA(TAU) X + BETA(TAU) X^3 + EPS G(X,X',TAU) = 0")
PRINT(" WHERE TAU = EPS T AND G IS POLYNOMIAL IN X AND X'"),
PRINT(" "),ALVAL:READ("ENTER ALPHA(TAU):"),
PRINT(" "),BEVAL:READ("ENTER BETA(TAU): "),
PRINT(" "),PRINT("ENTER G(X,X',TAU) USING Y=X':"),
G:READ(),
PRINT(" "),
PRINT("UNPERTURBED SOLUTION IS: X = RR CN(U,K) AND X' = Y = RR AA CN'(U,K)"),
PRINT("WHERE RR = AMPLITUDE AND U = 4 KC PHI = PHASE"),
PRINT(" "),PRINT("AVERAGING WILL USE A NEAR-IDENTITY TRANSFORMATION FROM"),
PRINT("(RR,PHI) TO (RBAR,PHIBAR) AS FOLLOWS:"),
```

```
PRINT(" "),
PRINT("RR=RBAR+EPS*W[1](RBAR,PHI)"),
PRINT("PHI=PHIBAR+EPS*W[2](RBAR,PHI)"),

/* Kill variables used by the routine */

KILL(K,AL,BE,RR,XX,YY,ZZ,SS,UU,TH,TAU,KC,EC,EPS,RBAR,KBAR,PHI,
    CNF,SNF,CNP,THETA,ZETA,ARCSIN,WITHINSEP,FF),

/* Set AL and BE dependency and check for numeric k value */

DEPENDS([AL,BE],TAU),
IF ALVAL = 0 THEN K:SQRT(1/2),
IF BEVAL = 0 THEN K:0,

/* Set WITHINSEP flag to zero if no double homoclinic loops exist */
/* in unperturbed system */

IF SCALARP(ALVAL) AND ALVAL>=0 THEN WITHINSEP:0,

/* Create REDUC routine that reduces an expression involving YY (cn') */
/* into its even and odd components and apply the identity */
/* (cn')^2 = (1-cn^2)*(1-k^2+k^2*cn^2) */

REDUC(EXPR):=BLOCK([EVEN,ODD,VAL],
    EVEN:EXPAND((EXPR+EV(EXPR,YY=-YY))/2),
    ODD:EXPAND((EXPR-EVEN)/YY),
    ODD:YY*EXPAND(EV(ODD,YY=SQRT((1-XX^2)*(1-K^2+K^2*XX^2)))),
    EVEN:EXPAND(EV(EVEN,YY=SQRT((1-XX^2)*(1-K^2+K^2*XX^2)))),
    VAL:EVEN+ODD
),

/* Substitute x = r cn and y = r a cn' into g */

G:EV(G,X=RR*XX,Y=RR*SQRT(AL+BE*RR^2)*YY),

/* Compute F[1] and F[2] */

F[1]:-REDUC(YY*G)/SQRT(BE*RR^2+AL),
IF DIFF(ALVAL,TAU)#0 THEN
    F[1]:F[1]+DIFF(AL,TAU,1)*RR*(XX-1)*(XX+1)/(2*(BE*RR^2+AL)),
IF DIFF(BEVAL,TAU)#0 THEN
    F[1]:F[1]+DIFF(BE,TAU,1)*RR^3*(XX-1)*(XX+1)*(XX^2+1)/(4*(BE*RR^2+AL)),

F[2]:-REDUC(G*(2*AL*BE*RR^2*YY*ZZ+2*AL^2*YY*ZZ-AL*BE*RR^2*XX^3
    -BE^2*RR^4*XX-2*AL*BE*RR^2*XX-2*AL^2*XX))
    /(4*RR*(BE*RR^2+AL)^(3/2)*(BE*RR^2+2*AL)*KC),
IF DIFF(ALVAL,TAU)#0 THEN
    F[2]:F[2]+DIFF(AL,TAU,1)*(AL*XX^2*ZZ-BE*RR^2*ZZ-2*AL*ZZ+AL*XX*YY)
    /(4*(BE*RR^2+AL)*(BE*RR^2+2*AL)*KC),
IF DIFF(BEVAL,TAU)#0 THEN
    F[2]:F[2]+DIFF(BE,TAU,1)*(AL*BE*RR^2*XX^4*ZZ+AL*BE*RR^2*ZZ+2*AL^2*ZZ
    +AL*BE*RR^2*XX^3*YY+BE^2*RR^4*XX*YY+2*AL*BE*RR^2*XX*YY)
    /(8*BE*(BE*RR^2+AL)*(BE*RR^2+2*AL)*KC),
```

```
/* If k=0 then simplify above by setting be=0 and zz=0 */

IF SCALARP(K) AND K = 0 THEN (
    F[1]:EV(F[1],BE=0),
    F[2]:EV(F[2],ZZ=0,BE=0)
),

/* Integrate F[ii] w.r.t.  phi (GENINT) and find mean FBAR[ii] */

FOR II:1 THRU 2 DO (
    FI[II]:GENINT(F[II],K),
    FF[II]:EV(FI[II],XX=0,YY=0,ZZ=0,TH=0),
    FBAR[II]:RATCOEF(FF[II],UU)+WITHINSEP*RATCOEF(FF[II],SS)*%PI/2/KC
),

/* Find transformation or not */

KILL(W,W1MEAN,W1INT),PRINT(" "),
Q1:READ("DO YOU WISH TO FIND THE TRANSFORMATION? (Y/N)"),

/* If q1<>n then find transformation by computing w[1] and w[2] */

IF Q1#N THEN (

    /* Compute W[1] and integrate w.r.t.  phi (GENINT). Find mean of W[1] */

    W[1]:1/SQRT(AL+BE*RR^2)*( EV(FI[1],UU=0)
    -WITHINSEP*RATCOEF(FF[1],SS)*%PI/2/KC*UU) ),
    W1INT:GENINT(W[1],K),
    W1MEAN:RATCOEF(W1INT,UU,1)+WITHINSEP*RATCOEF(W1INT,SS)*%PI/2/KC,

    /* When withinsep=1 then ss-%pi/2/kc*uu has zero mean.*/
    /* Make mean of W[1]=0 by adding a constant */

    W[1]:W[1]-W1MEAN,

    /* Find diff(w[1],phi) */

    DPHIW[1]:4*KC/SQRT(AL+BE*RR^2)*(F[1]-FBAR[1]),

    /* Find W1C=diff(aa/4/kc,rr) and simplify if possible */

    W1C:SQRT(BE*RR^2+AL)*(BE*RR^2*KC+2*AL*KC-2*AL*EC)
    /(4*RR*(BE*RR^2+2*AL)*KC^2),
    IF ALVAL=0 THEN (W1C:EV(W1C,AL=0),W1C:EV(W1C,ABS(RR)=RR)),
    IF BEVAL=0 THEN W1C:0,

    /* Find diff(w[2],phi) */

    DPHIW[2]:4*KC/SQRT(AL+BE*RR^2)*(F[2]-W[1]*W1C-FBAR[2]),

    /* Find W[2] */

    W[2]:1/SQRT(AL+BE*RR^2)*( EV(FI[2],UU=0)
     -WITHINSEP*RATCOEF(FF[2],SS)*%PI/2/KC*UU
     -W1C*(W1INT-W1MEAN*UU)

    /* W[2] has not been set to have zero mean */
),
```

```
/* Create a list of substitution rules for output */

IF SCALARP(K) AND K = 0 THEN
    PF:[XX=COS(2*%PI*PHI),YY=-SIN(2*%PI*PHI),UU=2*%PI*PHI,ABS(RR)=RBAR,
    RR=RBAR,KC=%PI/2,AL=ALVAL,BE=BEVAL]
ELSE
PF:[XX=CNF(4*KC*PHI),YY=CNP(4*KC*PHI),ZZ=ZETA(4*KC*PHI),
    TH=LOG(THETA(4*KC*PHI)/THETA(0)),SS=ARCSIN(KBAR*SNF(4*KC*PHI)),
    UU=4*KC*PHI,ABS(RR)=RBAR,RR=RBAR,AL=ALVAL,BE=BEVAL],
IF NOT SCALARP(K) THEN PFK:[K=SQRT(BEVAL*RBAR^2/2/(BEVAL*RBAR^2+ALVAL))]
    ELSE PFK:[],

/* Change results to output form */

FOR II:1 THRU 2 DO FBAR[II]:RATSIMP(EV(FBAR[II],PF,PFK,DIFF)),

/* Save averaged system into Rflow and Phiflow */

RFLOW:EPS*FACTOR(FBAR[1]),
PFLOW:EV(1/4/KC*SQRT(AL+BE*RR^2),PF)+EPS*FACTOR(FBAR[2]),

/* Print avg.  eqns, kbar^2, kc, ec (VAL contains this list of output) */

DERIVABBREV:TRUE,
VAL:[DIFF(RBAR(T),T)=RFLOW,DIFF(PHIBAR(T),T)=PFLOW,
    'KBAR^2=FACTOR(BEVAL*RBAR^2/2/(ALVAL+BEVAL*RBAR^2))],
IF K=0 THEN VALCOMP:[KC=%PI/2,EC=%PI/2] ELSE VALCOMP:[KC=KC(KBAR),EC=EC(KBAR)]
VAL:APPEND(VAL,VALCOMP),

PRINT("THE AVERAGED EQUATIONS ARE"),PRINT(" "),PRINT(VAL),PRINT(" "),

/* If q1<>n then simplify transformation */

IF Q1#N THEN (

    /* Change to output form */

    PRINT("SIMPLIFYING TRANSFORMATION"),
    FOR II:1 THRU 2 DO W[II]:EXPAND(EV(W[II],PF,PFK,DIFF)),

    /* Save transformations into Rtrans and Phitrans (TRANSF contains both) */

    RTRANS:RR=RBAR+EPS*MAP(FACTOR,W[1]),
    PTRANS:PHI=PHIBAR+EPS*MAP(FACTOR,W[2]),
    TRANSF:[CTRANS,PTRANS],

    /* Input to see transformation */

    Q2:READ("DO YOU WISH TO SEE THE TRANSFORMATION? (Y/N)"),
    IF Q2#N THEN (

    /* Print transformation */

    PRINT("THE TRANSFORMATION IS :"),PRINT(" "),PRINT(TRANSF)
    )
)
)$
```

```
/* F1,F2,W1,W2 integrator */

/* Routine to integrate integrands of the form :  */

/* (a) xx^m (b) xx^m yy (c) zz xx^m (d) zz xx^m yy */
/* (e) ss xx^m (f) ss xx^m yy */

/* Symbols */

/* XX = cn function */
/* YY = cn' function (derivative of cn w.r.t.  argument) */
/* ZZ = Jacobian zeta function */
/* SS = arcsin(k*sn(u,k)) = arcsin(k*sqrt(1-xx^2)) */
/* S0 = integral of SS with respect to u */
/* S2 = integral of SS*XX^2 with respect to u */
/* UU = argument and hence 1st elliptic integral */
/* TH = ln(theta(u)/theta(0)) */
/* KC,EC = complete elliptic integrls of 1st,2nd kinds */
/* K = modulus */

/* V contains the expression to be integrated.  */
/* Expressions are integrated w.r.t u.  */
/* For integrations w.r.t.  phi, multiply by 1/4/kc */

GENINT(V,K):=BLOCK([TEMP,STERMS,XTERMS,ZTERMS,XYT,XT,SYT,ST,ZYT,ZT,
    VALX,VALZ,VALS,VAL],
TEMP:EXPAND(V),

/* V is assumed to be in REDUC form, ie., V is linear in YY */

/* Separate V into categories:  */
/* */
/* XT contains terms in V of the form (a) */
/* XYT contains terms in V of the form (b) */
/* ZT contains terms in V of the form (c) */
/* ZYT contains terms in V of the form (d) */
/* ST contains terms in V of the form (e) */
/* SYT contains terms in V of the form (f) */

STERMS:EXPAND(DIFF(TEMP,SS)),
ZTERMS:EXPAND(DIFF(TEMP,ZZ)),
XTERMS:EXPAND(TEMP-SS*STERMS-ZZ*ZTERMS),
XYT:EXPAND(DIFF(XTERMS,YY)),
XT:EXPAND(XTERMS-YY*XYT),
SYT:EXPAND(DIFF(STERMS,YY)),
ST:EXPAND((STERMS-YY*SYT)),
ZYT:EXPAND(DIFF(ZTERMS,YY)),
ZT:EXPAND(ZTERMS-YY*ZYT),

/* Create XYINT function to integrate form (b) */

XYINT(VV):=BLOCK(VV:EXPAND(VV),EXPAND(INTEGRATE(VV,XX))),

/* Integrate forms (a) [using CNINT routine] and (b) */
```

```
VALX:CNINT(XT,K)+XYINT(XYT),

/* Integrate form (d) by integration by parts */

ARG:XYINT(ZYT),
VALZ:ZZ*ARG-CNINT(ARG*(1-K^2-EC/KC+K^2*XX^2),K)

Integrate form (f) by integration by parts */

ARG:XYINT(SYT),
VALS:SS*ARG-CNINT(ARG*K*XX,K),

/* Create a general Integration By Parts routine for forms (c) and (e) */

INTBYPARTS(VV,K,TYPE):=BLOCK([ARG,UUT,ZZT,SST,YYT,XXT,VALP],

    /* VV just contains the XX terms of forms (c) and (e) */
    /* TYPE indicates either form (c) or (e) */

    /* Find DERIV, the derivative of TYPE w.r.t.  u */

    IF TYPE = ZZ THEN DERIV:1-K^2-EC/KC+K^2*XX^2,
    IF TYPE = SS THEN DERIV:K*XX,

    /* Set ARG = integral of VV w.r.t.  u [using CNINT routine] */

    ARG:CNINT(VV,K),

    /* Separate ARG into categories:  */
    /* */
    /* UUT contains UU terms in ARG */
    /* ZZT contains ZZ terms in ARG */
    /* SST contains SS terms in ARG */
    /* YYT contains terms in ARG of the form (b) */
    /* XXT contains terms in ARG of the form (a) */

    UUT:DIFF(ARG,UU),
    ZZT:DIFF(ARG,ZZ),
    SST:DIFF(ARG,SS),
    YYT:DIFF(ARG,YY),
    XXT:EXPAND(ARG-UU*UUT-ZZ*ZZT-SS*SST-YY*YYT),

    /* Perform integration by parts */

    IF TYPE = ZZ THEN
    VALP:EXPAND(UUT*(UU*ZZ-TH)+ZZT*(ZZ^2/2)+EV(SST*SO*DERIV,XX=SQRT(S2/SO))
    +XYINT(YYT*DERIV)+CNINT(XXT*DERIV,K)),
    IF TYPE = SS THEN
    VALP:EXPAND(UUT*(UU*SS-SO)+ZZT*(ZZ*SS-K^2*S2-(1-K^2-EC/KC)*SO)
    +SST*(SS^2/2)+XYINT(YYT*DERIV)+CNINT(XXT*DERIV,K)),
    VALP:EXPAND(TYPE*ARG-VALP)
),

/* Integrate forms (c) and (e) using INTBYPARTS */

VALZ:VALZ+INTBYPARTS(ZT,K,ZZ),
VALS:VALS+INTBYPARTS(ST,K,SS),
```

```
/* Add together */

VAL:EXPAND(VALX+VALZ+VALS)
)$

/* CN function integrator */

/* Routine finds the integral of g(xx) where g is polynomial in XX */
/* and XX stands for the cn function */

/* Symbols */

/* XX = cn function */
/* YY = cn' function (derivative of cn w.r.t.  argument) */
/* ZZ = Jacobian zeta function */
/* SS = arcsin(k*sn(u,k)) = arcsin(k*sqrt(1-xx^2)) */
/* UU = argument and hence 1st elliptic integral */
/* KC,EC = complete elliptic integrls of 1st, 2nd kinds */
/* K = modulus */

CNINT(V,K):=BLOCK([TEMP,HI,IC,VAL],

/* Find highest power of cn in V and kill integration function IC */

TEMP:EXPAND(V),
HI:HIPOW(TEMP,XX),
KILL(IC),

/* IC[II] = integration function array that defines the integral of xx^ii */
/* It is defined recursively */

/* If k=0, then cn=cos so set the IC to use cosine routine */

IF SCALARP(K) AND EV(K) = 0 THEN (

    IC[0]:UU,
    IC[1]:-YY,
    IC[II]:=RATSIMP((II-1)/II*IC[II-2]-1/II*XX^(II-1)*YY)

) ELSE (

    IC[0]:UU,
    IC[1]:SS/K,
    IC[2]:1/K^2*(ZZ+(EC/KC-(1-K^2))*UU),
    IC[3]:1/2/K^3*((2*K^2-1)*SS-K*YY),
    IC[II]:=RATSIMP((II-2)*(2*K^2-1)*IC[II-2]+(II-3)*(1-K^2)*IC[II-4]
    -XX^(II-3)*YY)/K^2/(II-1)
),

/* Set VALue of the integral to zero */

VAL:0,

/* For each xx^ii expression found in V, substitute its integral IC[ii] */

FOR II:0 THRU HI DO VAL:VAL+RATCOEF(TEMP,XX,II)*IC[II],
VAL:EXPAND(VAL)
)$
```

Appendix C: Sample Run of MACSYMA Program

```
AVERAGE()$
AVERAGING OF X'' + ALPHA(TAU) X + BETA(TAU) X^3 + EPS G(X,X',TAU) = 0
   WHERE TAU = EPS T AND G IS POLYNOMIAL IN X AND X'

ENTER ALPHA(TAU):
A(TAU);

ENTER BETA(TAU):
1;

ENTER G(X,X',TAU) USING Y=X':
DEL*Y;

UNPERTURBED SOLUTION IS: X = RR CN(U,K) AND X' = Y = RR AA CN'(U,K)
WHERE RR = AMPLITUDE AND U = 4 KC PHI = PHASE

AVERAGING WILL USE A NEAR-IDENTITY TRANSFORMATION FROM
(RR,PHI) TO (RBAR,PHIBAR) AS FOLLOWS:

RR=RBAR+EPS*W[1](RBAR,PHI)
PHI=PHIBAR+EPS*W[2](RBAR,PHI)

DO YOU WISH TO FIND THE TRANSFORMATION? (Y/N)
N;

THE AVERAGED EQUATIONS ARE

 D                                                                   2
[-- (RBAR(T)) = - EPS (2 KC A(TAU) DEL - 2 EC A(TAU) DEL + KC RBAR DEL
 DT

        D                      D
+ 3 KC (---- (A(TAU))) - 3 EC (---- (A(TAU))))/(3 KC RBAR),
        DTAU                   DTAU

                                      2                       2
D                 SQRT(A(TAU) + RBAR )      2           RBAR
-- (PHIBAR(T)) = ---------------------, KBAR = ------------------,
DT                        4 KC                                  2
                                                2 (A(TAU) + RBAR )

KC = KC(KBAR), EC = EC(KBAR)]

[SYMBOLICS 3670 time = 211 seconds]
```

REFERENCES


[1] Byrd, P. and Friedman, M., *Handbook of Elliptic Integrals for Engineer and Scientists*, Springer-Verlag, Berlin, 1954.

[2] Cap, F. F., *Averaging Methods for the Solution of Non-linear Differential Equations with Periodic Non-harmonic Solutions*, International Journal of Non-Linear Mechanics, vol. 9, 1973, pp.441–450.

[3] Chirikov, B. V., *A Universal Instability of Many Dimensional Oscillator Systems*, Physics Reports, vol. 52, 1979, pp.263–379.

[4] Coppola, V.T., *Averaging of Strongly Nonlinear Oscillators Using Elliptic Functions*, Ph.D. dissertation, Cornell University, 1989.

[5] Coppola, V.T. and Rand, R.H., *Symbolic Computation and Perturbation Methods Using Elliptic Functions*, Transactions of the Sixth Army Conference on Applied Mathematics and Computing (1989), pp. 639–676.

[6] Coppola, V.T. and Rand, R.H., *Averaging Using Elliptic Functions: Approximation of Limit Cycles*, Acta Mechanica (1989) (to appear).

[7] Coppola, V.T. and Rand, R.H., *Chaos in a System with Periodically Disappearing Separatrix*, in preparation.

[8] Davis, H.T., *Introduction to Nonlinear Differential and Integral Equations*, Dover, NY, 1962.

[9] Garcia-Margallo, J. and Bejarano J., *Stability of Limit Cycles and Bifurcations of Generalized van der Pol Oscillators*, preprint.

[10] Garcia-Margallo, J. and Bejarano, J., *A Generalization of the Method of Harmonic Balance*, Journal of Sound and Vibration, vol. 116(3), 1987, pp. 591–595.

[11] Goldstein, H., *Classical Mechanics*, second edition, Addison-Wesley, Reading, Mass., 1980.

[12] Greenspan, B. and Holmes, P., *Repeated Resonance and Homoclinic Bifurcation in a Periodically Forced Family of Oscillators*, SIAM Journal of Mathematical Analysis, vol. 15, no.1, 1983, pp 69–97.

[13] Guckenheimer, J. and Holmes, P., *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Applied Mathematical Sciences, vol. 42, Springer-Verlag, NY, 1983.

[14] Hagedorn, P., *Non-Linear Oscillations*, Oxford University Press, NY, 1982.

[15] Kervokian, J. and Cole, J.D., *Perturbation Methods in Applied Mathematics*, Applied Mathematical Sciences, vol. 34, Springer-Verlag, NY, 1981.

[16] Kuzmak, G.E., *Asymptotic Solutions of Nonlinear Second Order Differential Equations with Variable Coefficients*, P.M.M. (English translation), vol. 23, no. 3, 1959, pp. 515–526.

[17] Minorsky, N., *Nonlinear Oscillations*, Van Nostrand, NY, 1962.

[18] Nayfeh, A., *Perturbation Methods*, Wiley-Interscience, NY, 1973.

[19] Nayfeh, A. and Mook, D.T., *Nonlinear Oscillations*, John Wiley and Sons, NY, 1979.

[20] Pocobelli, G., *Electron Motion in a Slowly Varying Wave*, Phys. Fluids, vol. 24(12), 1981, pp. 2173–2176.

[21] Rand, R.H. and Armbruster, D., *Perturbation Methods, Bifurcation Theory and Computer Algebra*, Applied Mathematical Sciences, vol. 65, Springer-Verlag, NY, 1987.

[22] Sanders, J.A. and Verhulst, F., *Averaging Methods in Nonlinear Dynamical Systems*, Applied Mathematical Sciences, vol. 59, Springer-Verlag, NY, 1985.

[23] Stoker, J.J., *Nonlinear Vibrations*, Wiley, NY, 1950.

[24] Yuste, S. and Bejarano, J., *Construction of Approximate Analytical Solutions to a New Class of Nonlinear Oscillator Equations*, Journal of Sound and Vibration, vol. 110(2), 1986, pp. 347-350.