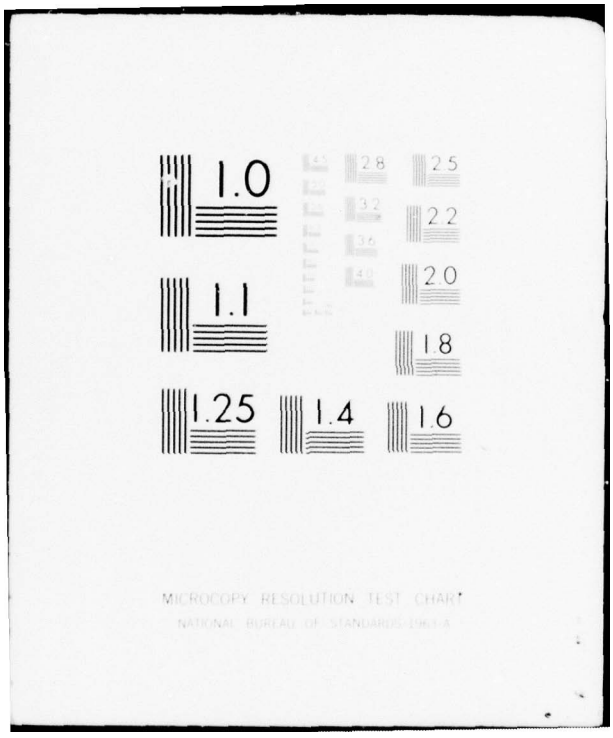
AD-A043 438 NAVAL RESEARCH LAB WASHINGTON D C F/G 9/2
SOLUTION OF THE PERTURBED HARMONIC OSCILLATOR BY COMPUTER.(U)
MAY 77 K T ALFRIEND , H M PICKARD

UNCLASSIFIED NRL-MR-3513 NL

| OF |
AD
A043438

END
DATE
FILMED
9 -77
DDC


1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

NRL Memorandum Report 3513

Solution of the Perturbed Harmonic Oscillator by Computer

K. T. ALFRIEND and H. M. PICKARD

Technical Staff
Space Systems Division

May 1977

NAVAL RESEARCH LABORATORY
Washington, D.C.

(9) Final rept.,

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 3513	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (14) NRL-MR-3513
4. TITLE (and Subtitle) (6) Solution of the Perturbed Harmonic Oscillator by Computer		5. TYPE OF REPORT & PERIOD COVERED Final Report on one phase of a continuing NRL problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) K.T. Alfriend and H.M. Pickard		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N RR014-02-41 79D03-06.101
11. CONTROLLING OFFICE NAME AND ADDRESS (16) RR01402 (17) RR014 02 41		(11) 12. REPORT DATE May 77 (12) 13. NUMBER OF PAGES 30 p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lindstedt-Poincare' Method

Strained Parameters

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A Fortran program has been developed for solving non-linear differential equations of the form $\ddot{X} + X + \epsilon f(X;\epsilon) = 0$ using Lindstedt's method. This method provides a uniformly valid series solution by introducing a transformation in the time variable via a scale factor which is a power series in the small parameter ϵ. Secular terms which cause non-uniformity may be then constrained to disappear. The details of the method are carried out using an algebraic manipulation package ~~written by R. Dasenbrock~~. The use of the computer program is discussed and examples are supplied.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 950 AB

CONTENTS

SOLUTION OF THE PERTURBED HARMONIC OSCILLATOR BY COMPUTER

I. INTRODUCTION

Many problems in applied mathematics reduce to a nonlinear oscillator. In fact, in celestial mechanics many times transformations are sought which will reduce a problem to a nonlinear oscillator. Some examples are:

1. The central force problem after the transformation $r = (1/u)$, $dt = r^2 d\tau$ where r is the radius and t is time;

2. Motion in the neighborhood of a stable equilibrium point and

3. The Vinti problem [1], motion about an oblate spheroid, which can be reduced to two uncoupled perturbed harmonic oscillators and a quadrature. The purpose of this report is the development of an analytic series solution of the perturbed harmonic oscillator

$$\ddot{X} + X + \varepsilon f(X;\varepsilon) = 0 \qquad (1)$$

where $f(X)$ is a polynomial in X and ε is a small parameter.

Note: Manuscript submitted April 28, 1977.

Low order solutions can be obtained "by hand" but the amount of algebraic manipulation required to obtain a high order solution makes carrying the solutions to higher order prohibitive. This report describes a computer program that develops an analytic series solution to the perturbed harmonic oscillator, Eq. (1), to the desired order in ϵ.

Lindstedt's method [2] is used to develop the solution and algebraic manipulation routines [3] are employed to carry out the algebraic manipulation.

II. TECHNICAL DISCUSSION

A. Motion Near an Equilibrium Point

As an example of a problem which reduces to a perturbed harmonic oscillator, consider the problem

$$\ddot{y} = F(y) \ , \ F(y_o) = 0 \ , \ \left.\frac{\partial F}{\partial y}\right|_{y=y_o} < 0 \qquad (2)$$

Thus $y = y_o$ is a stable equilibrium point. To analyze the motion in the neighborhood of the equilibrium point, expand $F(y)$ in a Taylor series about y_o, i.e.

$$F(y) = F(y_o) + F'(y_o)(y-y_o) + \tfrac{1}{2}F''(y_o)(y-y_o)^2 + \ldots \qquad (3)$$

and let

$$X = (y-y_o)/L \qquad (4)$$

where L is a representative scale length so that $X = O(1)$

at the maximum value of X. With

$$\varepsilon = 1/L$$
$$\sigma^2 = - F'(y_o) \tag{5}$$

Equation (2) reduces to

$$\ddot{X} + \sigma^2 X = - \varepsilon f(X;\varepsilon) \tag{6}$$

where $f(X;\varepsilon)$ is polynomial in X. A further time transformation of $s = \sigma t$ transforms the problem to one with a frequency of unity.

An exact solution of Eq. (1) can be obtained in terms of elliptic functions but sometimes elliptic function solutions are not particularly desirable and the reduction of the problem to obtain the elliptic function solution can be quite difficult. There are several methods for obtaining a series solution in power of ε but the best methods for this problem are Lindstedt's method [2] and Lie Series [4].

Lie Series involves the use of Hamiltonian mechanics and is easily automated on the computer. For studying properties of the solution, it is probably the best method, but to obtain solutions in terms of specific initial conditions, Lindstedt's method is the easiest to implement.

B. Lindstedt's Method

Consider the problem:

$$\ddot{X} + X = -\varepsilon f(X;\varepsilon),\ X(0) = A,\ \dot{X}(0) = B \tag{7}$$

To develop a series solution let

$$X = \sum_{n=o} \varepsilon^n X_n = X_o + \varepsilon X_1 + \varepsilon^2 X_2 + \ldots \tag{8}$$

Since the frequency is amplitude dependent in nonlinear oscillator problems, introduce a new time scale τ via

$$\tau = \omega t = (1 + \varepsilon\omega_1 + \varepsilon^2\omega_2 + \ldots)t \tag{9}$$

where the ω_n are undetermined at this point. Substituting Eqs. (8) and (9) into Eq. (7) and equating like powers of ε gives

$$\begin{aligned} X_o'' + X_o &= 0 \\ X_1'' + X_1 &= -2\omega_1 X_o'' - f(X_o;0) \\ X_n'' + X_n &= g_n(\omega_1,\omega_2,\ldots,\omega_{n-1}, X_o, X_1,\ldots, X_{n-1}) \end{aligned} \tag{10}$$

The initial conditions are satisfied by the zeroth order, i.e.

$$\begin{aligned} X_o(0) &= A,\ \ X_n(0) = 0,\ n = 1, 2, \ldots \\ X_o'(0) &= B \end{aligned} \tag{11}$$

$$\begin{aligned} X_1'(0) + \omega_1 X'_o(0) &= 0 \\ X_n'(0) + \omega_1 X'_{n-1}(0) + \ldots + \omega_{n-1} X_o'(0) &= 0 \end{aligned} \tag{12}$$

Eqs. (12) are a result of setting the zeroth order of $\frac{dX}{dt} = \omega\frac{dX}{d\tau}$ equal to B and the higher orders in ε equal to zero.

The solution of X_o is

$$X_o = A\cos\tau + B\sin\tau \tag{13}$$

Substitution of Eq. (13) into the X_1 equation results in resonance of the X_1 equation because the homogeneous portion of Eqs. (10) are identical. Since a uniformly valid solution is sought, no resonant or secular terms can be present in X_n, $n = 1, 2, \ldots$. Therefore, ω_1 is chosen to make the coefficient of the resonant term in X_1 zero. The rest of the ω_n terms are evaluated in like manner.

C. Example

As an example, consider a restricted form of the Duffing equation, i.e., the hard spring problem

$$\ddot{X} + X + \varepsilon X^3 = 0,\ X(0) = A,\ \dot{X}(0) = 0 \tag{14}$$

Application of Eqs. (8), (9) and (12) yields

$\underline{\varepsilon^o}$

$$X_o'' + X_o = 0,\quad X_o(0) = A,\ X_o(0) = 0 \tag{15}$$

$\underline{\varepsilon^1}$

$$X_1'' + X_1 = -2\omega_1 X_o'' - X_o^3 \tag{16}$$

$$X_1(0) = X_1'(0) = 0$$

$$\underline{\epsilon^2} \qquad X_2'' + X_2 = -(2\omega_2 + \omega_1^2)\, X_o'' - 2\omega_1 X_1'' - 3X_o^2\, X_1 \tag{17}$$

$$X_2(0) = X_2'(0) = 0$$

The solution to the zeroth order equation is

$$X_o = A\cos\tau \tag{18}$$

Substitution into Eq. (16) gives

$$X_1'' + X_1 = A(2\omega_1 - \frac{3}{4} A^2)\cos\tau - \frac{A^3}{4} \cos 3\tau$$

The $\cos\tau$ term results in resonant or secular terms in X_1 which cannot be present if a uniformly valid solution is to be obtained. To suppress this secular term set

$$\omega_1 = \frac{3}{8} A^2 \tag{19}$$

The corresponding solution of X_1 which satisfies the initial condition is

$$X_1 = \frac{A^3}{32} (\cos 3\tau - \cos\tau) \tag{20}$$

The solution of Eq. (14) is

$$X = A\cos\tau + \frac{1}{32} \epsilon A^3 (\cos 3\tau - \cos\tau) + O(\epsilon^2) \tag{21}$$

$$Y = \left[1 + \frac{3}{8} \epsilon A^2 + O(\epsilon^2)\right] t \tag{22}$$

III. COMPUTER IMPLEMENTATION

The computer program LNDSTD has been written to carry out the Lindstedt procedure. The program currently runs on the NRL DEC-System 10 computer and is available on request from Code 7904. Inputs to the program and other considerations are given below. Examples are given at the end.

A. Algebraic Expressions Written as Poisson Series

Before specifying the inputs to LNDSTD, it is necessary to describe the way in which algebraic expressions are coded in the algebraic routines used. LNDSTD uses the routines of R. Dasenbrock which are described in detail in ref. (3). In order to use these routines, an algebraic expression must be written as a Poisson series in which each term is of the form:

$$\frac{N}{D} \cdot \left[\prod_{i=1}^{12} v_i^{e_i} \right] \cdot \begin{pmatrix} \sin \\ \cos \end{pmatrix} \left[\sum_{i=1}^{6} \alpha_i \theta_i \right] \tag{23}$$

where:

N and D are decimal integers up to 11 digits;

v_i are polynomial variables;

e_i are the corresponding integer powers;

θ_i are trigonometric variables;

α_i are integer coefficients.

Each term is coded using six computer words. The first two words contain N and D respectively entered as decimal integers. The exponents e_1 - e_{12} corresponding to variables v_1 - v_{12} are packed into words three and four as two octal integers. (Note that an octal integer is indicated both here and on the DEC System 10 by using double quotation marks preceding the number). The fifth word is +1 if the cosine function is intended and -1 if the sine function is intended. The coefficients α_1 - α_6 are packed into the sixth word as an octal integer. The way in which a term is coded into these six words is best explained by the following example:

Code the expression $\frac{3}{4} v_1^3 v_4^{-2} v_5^9 v_8^{11} \cos(\theta_1 + 5\theta_6)$ into six computer words. The expression makes up a Poisson series of one term and may be written as

$$\frac{3}{4} v_1^3 v_2^0 v_3^0 v_4^{-2} v_5^9 v_6^0 v_7^0 v_8^{11} v_9^0 v_{10}^0 v_{11}^0 v_{12}^0$$

$$. x \cos\left[\theta_1 + 0\theta_2 + 0\theta_3 + 0\theta_4 + 0\theta_5 + 5\theta_6\right]$$

Word 1 is set equal to 3.

Word 2 is set equal to 4.

Each two octal digits of words 3, 4 and 6 are set equal to the octal value of the correspondong exponents or coefficients biased by the amount "40 for variables v_2 - v_6,

v_1 - v_{12}, θ_2 - θ_6 and by the amount "20 for variables v_1, v_7, θ_1.

Word 5 is set equal to +1.

The result is:

Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
N	D	$e_1e_2e_3e_4e_5e_6$	$e_7e_8e_9e_{10}e_{11}e_{12}$	sin cos	$\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5\alpha_6$
3	4	"234040365140	"205340404040	1	"214040404045

In what follows, any reference to a variable v_i or θ_i is an implicit reference to the position of the corresponding e_i or α_i within word 3, 4 or 6.

B. Inputs to LNDSTD

1. The first step in running LNDSTD is to specify whether input will be from the terminal or from disk and whether output will be to the terminal or to disk. In both cases, the response 'T' indicates the terminal, and any other response indicates disk. If disk is chosen in either case, a file name will be requested. Note that the extension .DAT is assumed for the file unless some other extension is specified or the file name is ended with a period.

The input file should be prepared following the formats of steps 2-7 below. All line sequence numbers must

be removed from the file.

2. The next input record specifies the desired order in ε of the solution. The order may not exceed 4.

3. The next input record specifies the number of terms in $f(x;\varepsilon)$, (NTERM).

4. The next 10 input records specify names for ε, x, initial value of x, initial value of $\dot{x}$, τ, and 5 other variables available to the user for defining $f(x;\varepsilon)$. Carriage returns may be entered for variables which will not be used. Names may be up to four characters long.

5. The next NTERM records define $f(x;\varepsilon)$. Note that the $f(x;\varepsilon)$ used must correspond to the differential equation written in the form

$$\ddot{x} + x + \varepsilon f(x;\varepsilon) = 0 \tag{24}$$

Each term is entered as one record. The function $f(x;\varepsilon)$ must be a polynomial in x and may not contain any trigonometric variables, i.e., each term is of the form

$$\frac{N}{D} \cdot \left[\prod_{i=1}^{12} v_i^{e_i} \right]. \tag{25}$$

Thus, each term is composed of a numerator, a denominator, and two octal words corresponding to variables 1 to 12. If ε appears in a term, variable v_1 must be used for it. Variables v_2 - v_6 must not be used; i.e., they should be

entered as "40's. Variable v_{12} must be used for x. All other variables should be entered as v_7 - v_{11}. The format for each term is (2I10, 3x, 012, 3x, 012).

6. The next record specifies the form of the zero order solution. In general, the form is

$$x_I \cos \theta + \dot{x}_I \sin \theta$$

where x_I and $\dot{x}_I$ are the initial values of x and $\dot{x}$ respectively. Enter 'C' if only the cosine solution is desired ($\dot{x}_I = 0$); 'S' if only the sine solution is desired ($x_I = 0$); or 'SC' if a sine and cosine solution is desired ($x_I \neq 0$ and $\dot{x}_I \neq 0$).

7. The last input record specifies options available to the user. Six options are available, and are input in the format (6I). Enter the value 1 if the option is desired and zero otherwise. A carriage return is equivalent to choosing no options. The options available are:

1) Skip the check at the end: If this option is not chosen, the program will substitute the final solution into the differential equation as a check. If the solution is large, the final check may involve a very long execution time and may possibly require more storage than is available.

2) Do not type final x.

3) Do not type final $\dot{x}$.

4) Do not type final t.

5) Do not type intermediate results.

6) Decide at each step whether to type intermediate results. If this option is chosen, LNDSTD will type out at each output step the number of terms in the expression corresponding to that step and will then type "PRINT(Y/N)?" The user should type "Y" to have the expression typed out. Any other response will suppress typing, and processing will continue.

If the first number typed is other than a 0 or a 1, this list of option descriptions will be typed.

C. Output from LNDSTD

LNDSTD first types out the information input by the user so that it may be verified. (The user is asked whether or not to proceed. A negative response causes LNDSTD to start over from the beginning.) If none of options 2-6 were taken, the program proceeds to type all intermediate and final results. Intermediate results consist of the solutions at each order i of w_i and x_i. Final results consist of the solutions of x, $\dot{x}$, and t to

the order in ε specified in step 2 of the input. If any of the options 2-6 are chose, output of results to the terminal is changed accordingly. If option 1 was not chosen, LNDSTD will carry out a check on the solution by evaluating the expression $\ddot{x} + x + \varepsilon f(x)$. If the check is successful, the series corresponding to this expression will have no terms in it, i.e., the result is zero.

Regardless of which options are chosen, the final solution of x, $\dot{x}$, and $\dot{f}(=w)$ are output to disk files named FOR21.DAT, FOR22.DAT, and FOR23.DAT. This output is in coded form, i.e., for each term in the Poisson series corresponding to the solution, one line containing six words as described earlier is written to the file. (Preceding the six words in each line is a number which is of significance only to the algebraic routine.) The variables appearing in the series and their names are shown at the beginning of each file. Aside from this list of variables and names, the file carries no other information which identifies the series. It is the responsibility of the user to supply this identification and to rename the files. The contents of these files may be transformed into a more readable form using the utility program TYPSER, which is illustrated in an example below.

D. Warnings and Special Considerations

1. While LNDSTD does not limit the number of terms in f(x), the maximum degree in x of f(x) is 32/i where i is the desired order. Practical considerations further restrict the length and degree of f(x). For example, a sine solution corresponding to $f(x) = x-2x^3$ is 35 terms long to 4th order in ϵ and required about one minute of CPU time. A sine and cosine solution corresponding to f(x) $f(x) = ax + 3x^2 + bx^3$, where a and b are constants, is about 700 terms long to 4th order and requires ten minutes of CPU time. In the latter case, not only is the execution time long, but the storage requirements approach the limits of the DEC System 10 computer. It is strongly recommended that the solution for any f(x) be first run only to 1st or 2nd order to get some idea of the size of the solution. For the second case above, a 2nd order solution is less than 100 terms long.

2. The problem most likely to be encountered by the user is that of insufficient core storage for the job. If this occurs, LNDSTD will type out a message stating that available storage has been exceeded, and execution will terminate. At this point, disk files FOR21.DAT, FOR22.DAT, and FOR23.DAT will contain the solution to the highest

order which was complete when execution terminated. In order to get a solution to higher order than this, the dimensions in LNDSTD must be increased and the job reloaded.

3. If the 0th order solution of x is chosen to be of the form $x_I \cos \theta$ then the initial value of $\dot{x}$ is zero and does not appear in the solution. In this case, variable 6 is available to the user, thus increasing the number of free variables to 6, i.e., variables 6-11. In any case, the name used for variable 6 will be that entered for initial $\dot{x}$ at Step 3.

4. In the present version of LNDSTD, all numeric constants must be approximated by a rational fraction N/D. Thus the largest constant possible is $\pm$ 34359738367 and the smallest possible is $\pm$ 1/34359738367. If during the procedure a number exceeds these bounds, execution will terminate and the message "LOSING ACCURACY IN REFACT" will be typed.

5. LNDSTD does not check to insure that the exponents e_i and trigonometric coefficients α_i do not exceed their bounds (-32 to +31 or -16 to +15). It is the responsibility of the user to insure that this does not occur. If it does occur, the result is that the exponent or coefficient overflows into the position of an adjacent

variable, and the end result will be incorrect. It will be evident in the final check if this has occurred. If the check is suppressed by choosing option 1, then the fact than an error has occurred will be evident only by the appearance of undesired variables or in the drastic effects on the variable which overflowed.

E. Example Runs

Use of Programs LNDSTD and TYPSER are illustrated in the three examples below. The first example solves the equation $\ddot{u} + u + \varepsilon(au + 3u^2 + bu^3) = 0$ to 1st order in ε. In the second example, the disk file EXAMPL is created for input to LNDSTD to solve the differential equation $\ddot{\eta} + \eta + \varepsilon(\eta - 2\eta^3) = 0$. In this case, no options are chosen and output is to disk file EXAMPL.LP, which is later queued to the line printer. (Note that in both examples, unused variables are given the name, "????". Since the appearance of these variables in any of the results indicates an error, this will serve to flag such an occurrence.)

Example 3 illustrates the use of TYPSER. The program is self-prompting and needs little explanation. If variable names appear at the beginning of the input file, TYPSER will use them. If the user wishes to use different names, all lines containing alphabetic characters must be

removed from the input file. TYPSER will then ask for new names and corresponding variable numbers, first for the polynomial variables, and then for the trigonometric variables. A carriage return indicates that no more names are to be entered. TYPSER will also truncate a series to any order in any polynomial variable and will reverse the print order if desired. (Series are normally printed in order of decreasing powers of variable 1). The series printed in example 3 are taken from examples 1 and 2. The variable names in the last series are changed as illustrated.

Example 1:

```
.RUN DSKD:LNDSTD[363,1Ø56]

LNDSTD:VERSION6......REVISED 2/16/77


START OF EXECUTION


 1.IS DATA TO BE ENTERED FROM A TERMINAL(T) OR FROM DISK(D)? T
   IS PRINTED OUTPUT TO BE TO TERMINAL(T) OR TO DISK(D)? T

 2.ENTER MAXIMUM ORDER IN E OF FINAL SOLUTION: 1

 3.ENTER NUMBER OF TERMS IN F(X): 3

 4.ENTER DESIRED NAMES FOR VARIABLES AND PARAMETERS:
                E= EPS
                X=   U
          INITIAL X=  UI
       INITIAL XDOT= UDI
              TAU= TAU
            PARAM1=????
            PARAM2=   A
            PARAM3=   B
            PARAM4=????
            PARAM5=????

 5.ENTER F(X) ONE TERM AT A TIME. THE ELEMENTS OF EACH TERM ARE
   INUM, IDNOM, IEXP1, IEXP2 AND THE FORMAT IS (2I1Ø,2(3XO12)):
NNNNNNNNNNDDDDDDDDDD   OOOOOOOOOOOO   OOOOOOOOOOOO
         1         1   2Ø4Ø4Ø4Ø4Ø4Ø   2Ø414Ø4Ø4Ø41
         3         1   2Ø4Ø4Ø4Ø4Ø4Ø   2Ø4Ø4Ø4Ø4Ø42
         1         1   2Ø4Ø4Ø4Ø4Ø4Ø   2Ø4Ø41Ø4Ø43

 6.SINE, COSINE, OR SINE & COSINE SOLUTION(S,C, OR SC)? SC

 7.ENTER OPTIONS (TYPE "2" FOR LIST); FORMAT(61): 2
                OPT1=1 MEANS SKIP CHECK AT END
                OPT2=1 MEANS DO NOT PRINT FINAL X
                OPT3=1 MEANS DO NOT PRINT FINAL XDOT
                OPT4=1 MEANS DO NOT PRINT FINAL TAUDOT
                OPT5=1 MEANS DO NOT PRINT INTERMEDIATE RESULTS
                OPT6=1 MEANS DECIDE PRINTING ON INDIVIDUAL BASIS
   ENTER OPTIONS: Ø Ø Ø Ø Ø 1
```

****

```
   THE PERTURBING FUNCTION IS:

F(X) =
 +A*U
 +B*U**3
 +3*U**2

   THE SOLUTION WILL BE A SINE & COSINE SOLUTION TO ORDER 1 IN EPS.

   THE OPTIONS CHOSEN ARE:

                OPT6:  DECIDE PRINTING ON INDIVIDUAL BASIS

   PROCEED(Y/N)? Y

   SOLUTIONS FOR    U,    UDOT AND  TAUDOT WILL BE OUTPUT IN PACKED
   FORMAT TO FILES NAMED FOR21.DAT, FOR22.DAT, AND FOR23.DAT.



*************************************************************************
*************************    ZEROTH ORDER    ************************
*************************************************************************


   UØ = (    2 TERMS)
PRINT(Y/N)?Y
 +UI*COS(TAU)
 +UDI*SIN(TAU)



*************************************************************************
*************************    FIRST ORDER    *************************
*************************************************************************

 OMEGA1 = (    3 TERMS)
PRINT(Y/N)?Y
 +3/8*UDI**2*B
 +1/2*A
 +3/8*B*UI**2


    U1 = (    17 TERMS)
PRINT(Y/N)?N

*************************************************************************
*************************    FINAL RESULT    ************************
*************************************************************************
```

```
   U = (   19 TERMS)
PRINT(Y/N)?Y
 -3/32*EPS*UDI**2*B*UI*COS(3TAU)
 +1/32*EPS*B*UI**3*COS(3TAU)
 -1/32*EPS*UDI**3*B*SIN(3TAU)
 +3/32*EPS*UDI*B*UI**2*SIN(3TAU)
 -1/2*EPS*UDI**2*COS(2TAU)
 +1/2*EPS*UI**2*COS(2TAU)
 +EPS*UDI*UI*SIN(2TAU)
 +3/32*EPS*UDI**2*B*UI*COS(TAU)
 +2*EPS*UDI**2*COS(TAU)
 -1/32*EPS*B*UI**3*COS(TAU)
 +EPS*UI**2*COS(TAU)
 +UI*COS(TAU)
 -9/32*EPS*UDI**3*B*SIN(TAU)
 -1/2*EPS*UDI*A*SIN(TAU)
 -21/32*EPS*UDI*B*UI**2*SIN(TAU)
 -2*EPS*UDI*UI*SIN(TAU)
 +UDI*SIN(TAU)
 -3/2*EPS*UDI**2
 -3/2*EPS*UI**2

   UDOT = (   17 TERMS
PRINT(Y/N)?N


 TAU DOT = (   4 TERMS)
PRINT(Y/N)?Y
 +3/8*EPS*UDI**2*B
 +1/2*EPS*A
 +3/8*EPS*B*UI**2
 +1



**************************************************************************
*************************      CHECK         *************************
**************************************************************************

D2(X)/DT2 + X + E*F(X;E) = (   Ø TERMS)
STOP

END OF EXECUTION
CPU TIME: 17.Ø6 ELAPSED TIME: 5:16.1Ø
EXIT

.RENAME U=FOR21.DAT, UDOT=FOR22.DAT, TAUDOT=FOR23.DAT
Files renamed:
FOR21.DAT
FOR22.DAT
FOR23.DAT
```

Example 2:

```
.SOS EXAMPL
Input: EXAMPL
00100   4
00200   2
00300    EPS
00400      4H
00500   ????
00600    HDI
00700    TAU
00800   ????
00900   ????
01000   ????
01100   ????
01200   ????
01300             1          1   204040404040   204040404041
01400            -2          1   204040404040   204040404043
01500   S
01600   0$
*ES

[DSKD:EXAMPL]

.RUN DSKD:LNDSTD[363.1056]

LNDSTD:VERSION 6......REVISED 2/16/77


START OF EXECUTION



1.IS DATA TO BE ENTERED FROM A TERMINAL(T) OR FROM DISK(D)? D
  IS PRINTED OUTPUT TO BE TO TERMINAL(T) OR TO DISK(D)? D
  ENTER INPUT FILE NAME: EXAMPL.
  ENTER OUTPUT FILE NAME: EXAMPL.LP



**************************************************************************
```

```
   THE PERTURBING FUNCTION IS:

F(X)=
 -2*H**3
 +H

   THE SOLUTION WILL BE A      SINE       SOLUTION TO ORDER 4 IN EPS.

   NO OPTIONS WERE CHOSEN.

   PROCEED(Y/N)? Y

   SOLUTIONS FOR    H,    HDOT AND  TAUDOT WILL BE OUTPUT IN PACKED
   FORMAT TO FILES NAMED FOR21.DAT, FOR22.DAT, AND FOR23.DAT.



**************************************************************************
*************************     ZEROTH ORDER     *************************
**************************************************************************

   HØ = (     1 TERMS)



**************************************************************************
*************************      FIRST ORDER     *************************
**************************************************************************

OMEGA1 = (     2 TERMS)

  H1 =       3 TERMS)



**************************************************************************
*************************     SECOND ORDER     *************************
**************************************************************************

 OMEGA2 = (     3 TERMS)

   H2  (     6 TERMS)=
```

```
**************************************************************************
************************      THIRD ORDER     ************************
**************************************************************************

 OMEGA3= (     4 TERMS)

    H3 = (     1Ø TERMS)



**************************************************************************
************************      FOURTH ORDER     ************************
**************************************************************************

OMEGA4 = (     5 TERMS)

   H4 = (     15 TERMS)



**************************************************************************
************************      FINAL RESULT     ************************
**************************************************************************

    H = (     35 TERMS)

    HDOT = (     31 TERMS)

 TAU DOT = (     15 TERMS)


**************************************************************************
*****************************      CHECK     *****************************
**************************************************************************

D2(X)/DT2 + X + E*F(X;E) = (     Ø TERMS)

   ** ZERO **

STOP
```

```
END OF EXECUTION
CPU TIME: 1:9.22          ELAPSED TIME: 4:59.00
EXIT

.RENAME H=FOR21.DAT, HDOT=FOR22.DAT, TAUHDT=FOR23.DAT
Files renamed:
FOR21.DAT
FOR22.DAT
FOR23.DAT

.PRINT EXAMPL.LP
Total of 11 blocks in 1 file in LPT request
```

Example 3

```
.SOS V=U
Edit: U
*FEND$
ØØ8ØØ   END
*PØ:8ØØ
ØØ1ØØ   THE VARIABLES USED ARE:
ØØ2ØØ     POLYNOMIAL VARIABLE  1 =  EPS
ØØ3ØØ     POLYNOMIAL VARIABLE  6 =  UDI
ØØ4ØØ     POLYNOMIAL VARIABLE  8 =    A
ØØ5ØØ     POLYNOMIAL VARIABLE  9 =    B
ØØ6ØØ     POLYNOMIAL VARIABLE 12 =   UI
ØØ7ØØ          TRIG. VARIABLE  6 =  TAU
ØØ8ØØ   END
*DØ:8ØØ
8 Lines (ØØ1ØØ/1:ØØ8ØØ) deleted
*ES
```

```
[DSKD:V]

.RUN DSKD:TYPSER[363,1Ø56]

START OF EXECUTION



ENTER INPUT FILE NAME: UDOT.
ENTER OUTPUT FILE NAME: UDOT.TTY
ENTER SERIES NAME: U TO 1ST ORDER
TRUNCATION?(Y/N): N
REVERSE PRINT ORDER?(Y/N): Y
CONTINUE OR QUIT?(C/Q) C
ENTER INPUT FILE NAME: H.
ENTER OUTPUT FILE NAME: H2.TTY
ENTER SERIES NAME: H TO 2ND ORDER
TRUNCATION?(Y/N): Y
VARIABLE NUMBER AND ORDER: 1 2
REVERSE PRINT ORDER?(Y/N): N
CONTINUE OR QUIT?(C/Q) C
ENTER INPUT FILE NAME: V.
ENTER OUTPUT FILE NAME: V.TTY
ENTER SERIES NAME: V TO 1ST ORDER
INPUT VARIABLE NAMES:
POLYNOMIAL VARIABLE: 1 EPSV
POLYNOMIAL VARIABLE: 6 VDI
POLYNOMIAL VARIABLE: 8 ALFA
POLYNOMIAL VARIABLE: 9 BETA
POLYNOMIAL VARIABLE: 12 VI
POLYNOMIAL VARIABLE:
TRIG. VARIABLE: 6 TAUV
TRIG. VARIABLE:
TRUNCATION?(Y/N): N
REVERSE PRINT ORDER?(Y/N): N
CONTINUE OR QUIT?(C/Q) Q
STOP

END OF EXECUTION
CPU TIME: 13.73 ELAPSED TIME: 11:24.23
EXIT

.TYPE UDOT.TTY, H2.TTY, V.TTY
SERIES NAME: U TO 1ST ORDER
SERIES LENGTH:    17
```

```
  -UI*SIN(F)
  -EPS*UI**2*SIN(F)
  -11/32*EPS*B*UI**3*SIN(F)
  -1/2*EPS*A*UI*SIN(F)
  -2*EPS*UDI**2*SIN(F)
  -15/32*EPS*UDI**2*B*UI*SIN(F)
  +UDI*COS(F)
  -2*EPS*UDI*UI*COS(F)
  -9/32*EPS*UDI*B*UI**2*COS(F)
  +3/32*EPS*UDI**3*B*COS(F)
  -EPS*UI**2*SIN(2F)
  +EPS*UDI**2*SIN(2F)
  +2*EPS*UDI*UI*COS(2F)
  -3/32*EPS*B*UI**3*SIN(3F)
  +9/32*EPS*UDI**2*B*UI*SIN(3F)
  +9/32*EPS*UDI*B*UI**2*COS(3F)
  -3/32*EPS*UDI**3*B*COS(3F)
SERIES NAME: H TO 2ND ORDER
SERIES LENGTH:    10

  +1/256*EPS**2*HDI**5*SIN(5F)
  +3/16*EPS**2*HDI**5*SIN(3F)
  -5/32*EPS**2*HDI**3*SIN(3F)
  +1/16*EPS*HDI**3*SIN(3F)
  +271/256*EPS**2*HDI**5*SIN(F)
  -45/32*EPS**2*HDI**3*SIN(F)
  +3/8*EPS**2*HDI*SIN(F)
  +9/16*EPS*HDI**3*SIN(F)
  -1/2*EPS*HDI*SIN(F)
  +HDI*SIN(F)
SERIES NAME: V TO 1ST ORDER
SERIES LENGTH:    19

  -3/32*EPSV*VDI**2*BETA*VI*COS(3TAUV)
  +1/32*EPSV*BETA*VI**3*COS(3TAUV)
  -1/32*EPSV*VDI**3*BETA*SIN(3TAUV)
  +3/32*EPSV*VDI*BETA*VI**2*SIN(3TAUV)
  -1/2*EPSV*VDI**2*COS(2TAUV)
  +1/2*EPSV*VI**2*COS(2TAUV)
  +EPSV*VDI*VI*SIN(2TAUV)
  +3/32*EPSV*VDI**2*BETA*VI*COS(TAUV)
  +2*EPSV*VDI**2*COS(TAUV)
  -1/32*EPSV*BETA*VI**3*COS(TAUV)
  +EPSV*VI**2*COS(TAUV)
  +VI*COS(TAUV)
  -9/32*EPSV*VDI**3*BETA*SIN(TAUV)
  -1/2*EPSV*VDI*ALFA*SIN(TAUV)
  -21/32*EPSV*VDI*BETA*VI**2*SIN(TAUV)
  -2*EPSV*VDI*VI*SIN(TAUV)
  +VDI*SIN(TAUV)
  -3/2*EPSV*VDI**2
  -3/2*EPSV*VI**2
```

REFERENCES

1. Alfriend, K.T., Deprit, A., Dasenbrock, R., and Pickard, H.M., "The Extended Phase Space Formulation of the Vinti Problem," Paper No. 76-827, presented at the 1976 AIAA/AAS Astrodynamics Conference, San Diego, Calif., August 1976.

2. Nayfeh, A.H., Perturbation Methods, John Wiley, 1972.

3. Dasenbrock, R., Algebraic Manipulation by Computer, NRL Report 7564, Naval Research Laboratory, June 1973.

4. Deprit, A.A., "Canonical Transformations Depending on a Small Parameter," Celestial Mechanics J., Vol. 1, No. 1, 1969, pp 12-30.