APPLICATIONS OF COMPUTER ALGEBRA

edited by

Richard Pavelle
Symbolics, Inc.

KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/Lancaster

15

NORMAL FORM AND CENTER MANIFOLD CALCULATIONS ON MACSYMA

R.H. RAND and W.L. KEITH

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

ABSTRACT

This paper describes the use of the symbolic manipulation system MACSYMA to facilitate normal form and center manifold computations, which arise in nonlinear dynamics problems. These computations represent a relatively new approach towards obtaining approximate solutions to systems of ordinary differential equations. A near-identity coordinate transformation in the form of a power series is used to locally transform a given system into a normal (i.e., simple or canonical) form. Center manifold calculations are a related technique for reducing the number of dimensions in a given system of ordinary differential equations by restricting attention to the flow on an invariant subspace. A MACSYMA program developed to perform such computations is presented and described in detail. The program is illustrated by applying it to a sample nonlinear dynamics problem.

NORMAL FORMS

There exist a variety of methods for obtaining approximate solutions to ordinary differential equation problems (e.g., perturbations, averaging, harmonic balance; see (1)). Most of these share a common approach in which the solution of the problem is written in the form of an infinite series, and the method involves a scheme for obtaining an approximation based on an n term truncation of the series.

In contrast to these methods, the method of normal forms does not involve expanding the solution in an infinite series. Rather, it is based on the idea of approximately transforming the differential equations themselves into a form which is easily solved. The method involves generating a transformation of dependent variables in the form of an infinite series, and computing the coefficients of the series so that the resulting transformed differential equations are in a normal (simple, canonical) form.

The method can be illustrated by considering a 2 dimensional system (although the same process applies to an n dimensional system):

(1) $x1' = f(x1,x2)$

(2) $x2' = g(x1,x2)$

We assume that the coordinates xi have been chosen such that the origin x1=x2=0 is an equilibrium point:

(3) $f(0,0) = g(0,0) = 0.$

We Taylor expand f and g about the origin,

(4) $x1' = a\ x1 + b\ x2 + F(x1,x2)$

(5) $x2' = c\ x1 + d\ x2 + G(x1,x2)$

where F and G are strictly nonlinear (i.e., contain terms of order 2 and higher).

We further assume that the coordinates xi have been chosen such that the linear system associated with (4),(5) is in canonical form (a problem in elementary linear algebra). In the case of real eigenvalues this will be diagonalized (or more generally, in Jordan canonical form), while for complex roots $u \pm iw$ we choose the real (but nondiagonal) form

(6) $$\begin{bmatrix} u & -w \\ w & u \end{bmatrix}$$

(In the case of an n dimensional system a corresponding form is chosen, built up from Jordan blocks, see e.g. (2).)

The linear transformation which puts the linearized system in canonical form can be thought of as the first step in a sequence of transformations, each of which simplifies the form of the terms of order i in (4),(5), for i=1,2,3,...

In the case in which F and G contain quadratic terms, we transform from x1,x2 to y1,y2 coordinates via a near-identity transformation with general quadratic terms:

$$x1 = y1 + a20\ y1^2 + a11\ y1\ y2 + a02\ y2^2 \quad (7)$$

$$x2 = y2 + b20\ y1^2 + b11\ y1\ y2 + b02\ y2^2 \quad (8)$$

where the aij and bij are to be determined.

We substitute (7),(8) into (4),(5) and neglect terms of order 3 (which do not influence the quadratic coefficients a20,...,b02). The resulting equations are linear in the derivatives yi'(t). We solve for the yi'(t) and again expand about y1=y2=0 and neglect terms of order 3. The result is of the form:

$$y1' = a\ y1 + b\ y2 + c120\ y1^2 + c111\ y1\ y2 + c102\ y2^2 \quad (9)$$

$$y2' = c\ y1 + d\ y2 + c220\ y1^2 + c211\ y1\ y2 + c202\ y2^2 \quad (10)$$

where cijk is the coefficient of $y1^j\ y2^k$ in the yi equation.

The cijk are known linear functions of the aij, bij. The linear terms in (9),(10) are identical to the linear terms in (4),(5) due to the near-identity nature of the transformation (7),(8).

The method consists of choosing the coefficients aij and bij so as to put equations (9) and (10) into a canonical form. The natural choice is to remove all the quadratic terms, c120=...=c202=0, although there are exceptional situations (involving resonances or repeated zero eigenvalues) in which this is not possible, see (3).

Once the coefficients in (7),(8) have been determined, we extend the transformation of xi to yi coordinates to include cubic terms. (Note that even if (4),(5) do not contain cubic terms, the transformation (7),(8) will generally introduce cubic terms.) Proceeding as before we compute the equations on the yi(t) neglecting terms of order 4, and choose the third order coefficients aij, bij in order to best simplify these equations. Note that the transformation up to and including cubic terms will not effect the already determined quadratic terms in the normal form.

Proceeding in this fashion we can (in principle) generate the desired transformation to any order of accuracy. Note, however, that the use of truncated power series can be expected to restrict the applicability of the method to a neighborhood of the origin.

CENTER MANIFOLDS

Center manifold theory is a related method which uses power series expansions of coordinates in order to reduce the dimension of a system of ordinary differential equations. The method involves restricting attention to an invariant subspace (called a center manifold) which contains all of the essential behavior of the system in the neighborhood of an equilibrium point, in the limit as time t approaches infinity.

This method is applicable to systems which, when linearized around an equilibrium point, have some eigenvalues which have zero real part, and others which have negative real part. (We assume that no eigenvalues have positive real part, since in such a case the center manifold is not attractive as t goes to infinity. Thus we assume that we are in the critical case of Lyapunov (see (4), p.150) in which the stability of the equilibrium point cannot be determined by the linearized equations.)

The idea of the method is that the components of the solution of the linearized equations which correspond to those eigenvalues with negative real part will decay as t goes to infinity, and hence the motion of the linearized equations will asymptotically approach the space S1 spanned by the eigenvectors corresponding to those eigenvalues with zero real part. The center manifold theorem (see (5)) assures us that this picture (which is so far based on the linearized equations) extends to the full nonlinear equations, as follows:

There exists a (generally curved) subspace S2 (the center manifold) which is tangent to the (flat) subspace S1 at the equilibrium point, and which is invariant under the flow given by the nonlinear equations. All solutions which start sufficiently close to the equilibrium point will tend asymptotically to the center manifold. In particular, the theorem states that the stability of the equilibrium point in the full nonlinear equations is the same as its stability in the flow on the center manifold (ref.(5), p.4). Moreover, any additional equilibrium points or periodic motions (limit cycles) which occur in a neighborhood of the given equilibrium point on the center manifold are guaranteed to exist in the full nonlinear equations (ref.(5), p.29).

In order to illustrate this method, we consider the following system of three differential equations:

(11) $x' = a x + b y + p(x,y,z)$

(12) $y' = c x + d y + q(x,y,z)$

(13) $z' = -z + r(x,y,z)$

where p,q and r are strictly nonlinear in x,y,z. Here the linearized x and y equations are uncoupled from the linearized z equation, and we assume that the coefficients a,b,c,d are such that the linearized x,y equations have eigenvalues with zero real part (i.e. either a pure imaginary complex conjugate pair or a double zero.)

In this problem the center manifold is a surface which is tangent to the xy plane at the origin. We may obtain an approximate expression for it by writing

(14) $z = h(x,y) = K20\ x^2 + K11\ x\ y + K02\ y^2 + \ldots$

where constant and linear terms have been omitted in order that z = h(x,y) be tangent to the xy plane.

The coefficients Kij in (14) are to be found by requiring z = h(x,y) to be invariant under the flow (11)-(13). This may be accomplished by differentiating (14):

(15) $z' = 2\ K20\ x\ x' + K11\ (x\ y'+x'\ y) + 2\ K02\ y\ y' + \ldots$

and substituting expressions for x',y',z' given by (11)-(13). The resulting expression, which depends on x,y and z, can be made to depend on x and y only, by using (14). Finally we may collect terms and set the coefficients of x^2, xy and y^2 to zero in order to obtain K20, K11 and K02. By including higher order terms in (14), this process may be extended to arbitrary accuracy.

Note that once z = h(x,y) is known, it may be substituted into (11),(12), thereby giving an abbreviated system of 2 differential equations representing the flow on the center manifold (or rather its projection onto the xy plane.)

We have just described the standard procedure for calculating center manifolds (5). In this work, however, we shall accomplish this computation in an equivalent but different manner. We shall consider the center manifold computation as a normal form problem, based on the following near-identity transformation from (x,y,z) to (u,v,w) coordinates:

(16) $x = u$

(17) $y = v$

(18) $z = w + h(u,v) = w + K20\ u^2 + K11\ u\ v + K02\ v^2$

where we have neglected terms of order 3.

When w=0 in (16)-(18), we obtain the previous expression (14) for the center manifold. Our procedure will be to substitute (16)-(18) into the differential equations (11)-(13), and to transform to new equations on u,v,w (as described in the previous section on normal forms). These will be of the form:

(19) $u' = a\,u + b\,v + Q1(u,v,w)$

(20) $v' = c\,u + d\,v + Q2(u,v,w)$

(21) $w' = -w + Q3(u,v,w)$

where Qi(u,v,w) stands for quadratic terms in u,v,w. We then set w=0 in the w equation, (21), and obtain the coefficients Kij by equating to zero the coefficients of the remaining quadratic terms u^2, uv, v^2.

We have adopted this scheme for performing center manifold calculations using normal form theory in order to use a single MACSYMA program to accomplish both kinds of problems. Although we have illustrated the procedure only for the case of a system of 3 differential equations with a 2 dimensional center manifold (11)-(13), the same scheme of embedding the center manifold calculation in a normal form problem will work for any size system.

EXAMPLE

We shall demonstrate our MACSYMA programs for accomplishing normal form and center manifold calculations by applying them to the following system of three differential equations:

(22) $x' = y$

(23) $y' = - x - x\,z$

(24) $z' = - z + x^2$

These equations represent a vibrating system with parametric feedback control, and have been discussed in (6).

Eqs.(22)-(24) have an equilibrium point at the origin. We shall be interested in the question of its stability. This system is of the form (11)-(13) previously discussed, and hence possesses a center manifold which is tangent to the xy plane at the origin.

In what follows we shall present the record of a MACSYMA session in which we first obtain an approximation to the center manifold, thereby reducing the dimension of the system from 3 to 2. Then we shall use normal forms to treat the resulting system, enabling us to determine the stability of the equilibrium point at the origin.

Before proceeding, we must consider the appropriate normal form for this problem. Takens (7) has shown that any system of the form

(25) $$x' = y + f(x,y)$$

(26) $$y' = -x + g(x,y)$$

where f and g are strictly nonlinear in x,y, can be put in the normal form

(27) $$r' = a1\ r^3 + a2\ r^5 + \ldots$$

(28) $$\theta' = -1 + b1\ r^2 + b2\ r^4 + \ldots$$

where r and θ are polar coordinates

(29) $$u = r \cos \theta, \qquad v = r \sin \theta$$

and where u,v are related to x,y by a near-identity transformation. In rectangular coordinates, (27),(28) become

(30) $$u' = v + a1\ (u^2 + v^2)\ u + b1\ (u^2 + v^2)\ v + O(5)$$

(31) $$v' = -u + a1\ (u^2 + v^2)\ v - b1\ (u^2 + v^2)\ u + O(5)$$

Here, then, is the MACSYMA session. The reader is referred to (8) for an introduction to MACSYMA.

We begin by loading a file called NORMFORM7.MAC, and then displaying the user instructions which have been saved in a variable named GO (see (8), Chapter 2):

```
(C1) LOADFILE(NORMFORM7,MAC)$

(C2) GO;
(D2) THIS FILE CONTAINS NF(), A NORMAL FORM FUNCTION.
WHEN ENTERING TRANSFORMATION, GEN(N) WILL GENERATE THE
GENERAL TERMS OF HOMOGENEOUS ORDER N IN 2 VARIABLES.
THE UTILITY DECOMPOSE() WILL ISOLATE THE COEFFS OF THE
NEW EQS., BUT WORKS ONLY FOR 2 EQS.
```

The MACSYMA program consists of the main function NF, and of the two auxiliary functions GEN and DECOMPOSE. The use of each of these functions will be illustrated in what follows. We start the center manifold computation by calling NF:

```
(C3) NF( )$
DO YOU WANT TO ENTER NEW VARIABLE NAMES (Y/N) ?
Y;
HOW MANY EQS
3;
SYMBOL FOR OLD X[ 1 ]
X;
SYMBOL FOR OLD X[ 2 ]
Y;
SYMBOL FOR OLD X[ 3 ]
Z;
SYMBOL FOR NEW X[ 1 ]
U;
SYMBOL FOR NEW X[ 2 ]
V;
SYMBOL FOR NEW X[ 3 ]
W;
DO YOU WANT TO ENTER NEW D.E.'S (Y/N) ?
Y;
ENTER RHS OF EQ. NO. 1 ,   D X /DT =
Y;
X  = Y
 T
ENTER RHS OF EQ. NO. 2 ,   D Y /DT =
-X-X*Z;
Y  = - X Z - X
 T
ENTER RHS OF EQ. NO. 3 ,   D Z /DT =
-Z+X**2;
       2
Z  = X   - Z
 T
```

Having entered the variable names and differential equations (22)-(24), we next enter the transformation of variables (16)-(18):

```
INPUT NEAR-IDENTITY TRANSFORMATION
(USE PREV FOR PREVIOUS TRANSFORMATION)
X = U + ?
0;
X = U
Y = V + ?
0;
Y = V
Z = W + ?
K20*U**2+K11*U*V+K02*V**2;
                   2                    2
Z = W + K02 V  + K11 U V + K20 U

ENTER TRUNCATION ORDER (HIGHEST ORDER TERMS TO BE KEPT)
2;
```

The program now computes the transformed equations to order 2. These correspond to eqs.(19)-(21) given previously:

```
U  + . . . = V + . . .
 T
V  + . . . = - U - W U + . . .
 T
                                             2
W  + . . . = - W + ((K11 - K20 + 1) U
 T
                                                         2
 + (- K11 - 2 K20 + 2 K02) V U + (- K11 - K02) V  ) + . . .
DO YOU WANT TO ENTER ANOTHER TRANSFORMATION (Y/N) ?
N;
```

In order to find the coefficients Kij which specify the center manifold, we set W=0 in the transformed equations. Note the standard MACSYMA use of % to refer to the previous result.

```
(C4) %,W=0;
(D4) [[U  = V, V  = - U, 0 =
        T       T

                  2
(- K11 - K02) V  + (- 2 K20 - K11 + 2 K02) U V

                           2
 + (- K20 + K11 + 1) U  ]]
```

We use the MACSYMA function PICKAPART to obtain the equations on the Kij:

```
(C5) PICKAPART(PART(%,1,3),3);

(E5)                          - K11 - K02

                                   2
(E6)                              V

(E7)                   - 2 K20 - K11 + 2 K02

(E8)                        - K20 + K11 + 1

                                   2
(E9)                              U

(D9)                 0 = E5 E6 + E7 U V + E8 E9

(C10) SOLVE([E5,E7,E8],[K20,K11,K02]);
                           3            2            2
(D10)             [[K20 = -, K11 = - -, K02 = -]]
                           5            5            5
```

Having solved for the Kij, we plug them into the transformation (called TRANS here), and substitute W=0 to obtain an approximate expression for the center manifold:

```
(C11) TRANS,%;
                                         2                  2
                                      2 V     2 U V    3 U
(D11)   [X = U, Y = V, Z = W + ---- - ----- + ----]
                                       5        5       5

(C12) PART(%,3),W=0,U=X,V=Y;
                          2                  2
                       2 Y     2 X Y    3 X
(D12)              Z = ---- - ----- + ----
                        5       5       5
```

Our next step is to substitute the expression D12 for the center manifold into the x and y equations, thereby obtaining an approximation for the flow on the center manifold. We once again call our function NF:

```
(C13) NF()$
DO YOU WANT TO ENTER NEW VARIABLE NAMES (Y/N) ?
Y;
HOW MANY EQS
2;
SYMBOL FOR OLD X[ 1 ]
X;
SYMBOL FOR OLD X[ 2 ]
Y;
```

```
SYMBOL FOR NEW X[ 1 ]
U;
SYMBOL FOR NEW X[ 2 ]
V;
DO YOU WANT TO ENTER NEW D.E.'S (Y/N) ?
Y;
ENTER RHS OF EQ. NO. 1 ,   D X /DT =
Y;
```

$$X_T = Y$$

```
ENTER RHS OF EQ. NO. 2 ,   D Y /DT =
EV(-X-X*Z,D12);
```

$$Y_T = - X \left(\frac{2 Y^2}{5} - \frac{2 X Y}{5} + \frac{3 X^2}{5}\right) - X$$

We wish to transform these equations via a near-identity transformation with general coefficients. Note that since all quadratic terms are absent from the differential equations, our near-identity transformation begins with cubic terms. To save typing, the function GEN is utilized:

```
INPUT NEAR-IDENTITY TRANSFORMATION
(USE PREV FOR PREVIOUS TRANSFORMATION)
X = U + ?
GEN(3);
```

$$X = A_{0, 3} V^3 + A_{1, 2} U V^2 + A_{2, 1} U^2 V + A_{3, 0} U^3 + U$$

```
Y = V + ?
GEN(3);
```

$$Y = B_{0, 3} V^3 + B_{1, 2} U V^2 + B_{2, 1} U^2 V + V + B_{3, 0} U^3$$

```
ENTER TRUNCATION ORDER (HIGHEST ORDER TERMS TO BE KEPT)
3;
```

The program now computes the transformed equations representing the flow on the center manifold to order 3:

$$U_T + . . . = V + ((A_{2, 1} + B_{3, 0}) U^3$$

$$+ (B_{2, 1} - 3 A_{3, 0} + 2 A_{1, 2}) V U^2$$

$$+ (- 2 A_{2, 1} + B_{1, 2} + 3 A_{0, 3}) V^2 U$$

$$+ (- A_{1,2} + B_{0,3}) V^3) + \ldots$$

$$V_T + \ldots = - U + ((5 B_{2,1} - 5 A_{3,0} - 3) U^3$$

$$+ (- 5 A_{2,1} - 15 B_{3,0} + 10 B_{1,2} + 2) V U^2$$

$$+ (- 10 B_{2,1} - 5 A_{1,2} + 15 B_{0,3} - 2) V^2 U$$

$$+ (- 5 B_{1,2} - 5 A_{0,3}) V^3)/5 + \ldots$$

```
DO YOU WANT TO ENTER ANOTHER TRANSFORMATION (Y/N) ?
N;
```

We now wish to isolate the coefficients of the terms in the transformed equations. As a convenient alternative to the MACSYMA function PICKAPART which we used previously (see C5), we use our function DECOMPOSE:

```
(C14) DECOMPOSE( )$
C[I,J,K] IS THE COEFF OF U **J V **K IN THE I-TH EQUATION
```

We aim for the Takens normal form of eqs.(30),(31). Using the coefficients C[I,J,K] assigned by DECOMPOSE, we produce the necessary equations:

(C15) C[1,3,0]=C[1,1,2];

(D15) $$B_{3,0} + A_{2,1} = - 2 A_{2,1} + B_{1,2} + 3 A_{0,3}$$

(C16) C[1,3,0]=C[2,2,1];

(D16) $$B_{3,0} + A_{2,1} = - 3 B_{3,0} - A_{2,1} + 2 B_{1,2} + \frac{2}{5}$$

(C17) C[1,3,0]=C[2,0,3];

(D17) $$B_{3,0} + A_{2,1} = - B_{1,2} - A_{0,3}$$

(C18) C[1,0,3]=C[1,2,1];

(D18) $$B_{0,3} - A_{1,2} = - 3 A_{3,0} + B_{2,1} + 2 A_{1,2}$$

```
(C19) C[1,0,3]=-C[2,3,0];
                                                3
(D19)         B      - A      = A      - B    + -
               0, 3     1, 2     3, 0     2, 1  5

(C20) C[1,0,3]=-C[2,1,2];
                                                          2
(D20)  B      - A      = 2 B      + A      - 3 B      + -
        0, 3     1, 2        2, 1     1, 2       0, 3   5
```

We now have 6 equations in 8 unknowns, A03,A12,A21,A30,B03, B12,B21,B30. We use the MACSYMA function SOLVE to automatically unravel this linear algebra problem. Expressions are returned for A03,A12,A21,A30,B03 and B12 in terms of B21 and B30 (called %R1 and %R2 respectively):

```
(C21) SOLVE([D15,D16,D17,D18,D19,D20],
[A[0,3],A[1,2],A[2,1],A[3,0],B[0,3],B[1,2],B[2,1],B[3,0]]);
                       10 %R1 - 1              20 %R2 - 7
(D21) [[A       = - ----------, A       = ----------,
          0, 3            10        1, 2       20

        20 %R1 - 1              40 %R2 - 13
A     = - ----------, A     = -----------,
 2, 1         20       3, 0        40

      40 %R2 - 3              20 %R1 - 3
B     = ----------, B     = ----------, B     = %R2,
 0, 3       40       1, 2       20       2, 1

B     = %R1]]
 3, 0
```

We check the previous calculation by rerunning it with the values of the coefficients D21 substituted into the previous general near-identity transformation. Note the use of the variable PREV to refer to the previous transformation:

```
(C22) NF()$
DO YOU WANT TO ENTER NEW VARIABLE NAMES (Y/N) ?
N;
DO YOU WANT TO ENTER NEW D.E.'S (Y/N) ?
N;
X  = Y
 T
                2              2
             2 Y    2 X Y   3 X
Y  = - X (---- - ----- + ----) - X
 T            5       5      5
```

```
INPUT NEAR-IDENTITY TRANSFORMATION

(USE PREV FOR PREVIOUS TRANSFORMATION)
X = U + ?
PREV,D21;
```

$$X = -\frac{(10\ \%R1 - 1)\ V^3}{10} + \frac{(20\ \%R2 - 7)\ U\ V^2}{20} - \frac{(20\ \%R1 - 1)\ U^2\ V}{20} + \frac{(40\ \%R2 - 13)\ U^3}{40} + U$$

```
Y = V + ?
PREV,D21;
```

$$Y = \frac{(40\ \%R2 - 3)\ V^3}{40} + \frac{(20\ \%R1 - 3)\ U\ V^2}{20} + \%R2\ U^2\ V + V + \%R1\ U^3$$

```
ENTER TRUNCATION ORDER (HIGHEST ORDER TERMS TO BE KEPT)
3;
```

Now the program once again computes the transformed equations. Note that while the transformation involves the arbitrary quantities %R1 and %R2, the resulting equations are unique:

$$U_T + . . . = V + \frac{2\ U^3 + 11\ V\ U^2 + 2\ V^2\ U + 11\ V^3}{40} + . . .$$

$$V_T + . . . = - U - \frac{11\ U^3 - 2\ V\ U^2 + 11\ V^2\ U - 2\ V^3}{40} + . . .$$

```
DO YOU WANT TO ENTER ANOTHER TRANSFORMATION (Y/N) ?
N;
```

Now we simplify the final equations, labeled D22, by transforming to polar coordinates:

```
(C23) DEPENDS([R,THETA],T);
(D23)                    [R(T), THETA(T)]

(C24) [U=R*COS(THETA),V=R*SIN(THETA)];
(D24)        [U = R COS(THETA), V = R SIN(THETA)]
```

```
(C25) D22,%,DIFF$
(C26) SOLVE(PART(%,1),[DIFF(R,T),DIFF(THETA,T)]);
                  3    2              3    2
                 R  SIN (THETA) + R  COS (THETA)
(D26) [[R   = -------------------------------,
         T                   20

                     2    2                   2    2
              11 R  SIN (THETA) + 11 R  COS (THETA) + 40
    THETA   = - ------------------------------------------]]
         T                        40

(C27) TRIGSIMP(%);
                      3                 2
                     R              11 R  + 40
(D27)        [[R   = --, THETA   = - ----------]]
                T    20         T        40
```

The result of the computation follows from the derived normal form D27:

$$r' = r^3 / 20 + O(5) \tag{32}$$

$$\theta' = -1 - 11\, r^2 /40 + O(4) \tag{33}$$

Eq.(32), used in conjunction with the center manifold theorem, implies that the equilibrium point r=0 is unstable.

THE MACSYMA PROGRAM

Before describing in detail the MACSYMA functions which we used in the previous example, we continue the previous run with FUNCTIONS and DISPFUN(ALL) commands, thereby displaying all user-defined functions:

```
(C28) FUNCTIONS;
(D28) [NF(), INP1(), INP2(), INP3(), INP4(), SETUP(),
STEP1(), STEP2(), STEP3(), GEN(NN), AUX(II, JJ, ROW),
DECOMPOSE()]

(C29) DISPFUN(ALL);
(E29) NF() := BLOCK(TEST :
READ("DO YOU WANT TO ENTER NEW VARIABLE NAMES (Y/N) ?"),
IF TEST = N THEN GO(JUMP), INP1(), SETUP(), JUMP, INP2(),
LOOP, INP3(), INP4(), STEP1(), STEP2(), STEP3(),
BRANCH : READ("DO YOU WANT TO ENTER ANOTHER TRANSFORMATION
(Y/N) ?"), IF BRANCH = Y THEN GO(LOOP), TEMP4)
```

```
(E30) INP1( ) := (N : READ("HOW MANY EQS"),
FOR I THRU N DO X  : READ("SYMBOL FOR OLD X[", I, "]"),
                 I

FOR I THRU N DO Y  : READ("SYMBOL FOR NEW X[", I, "]"))
                 I

(E31) INP2( ) :=
(PRINT("DO YOU WANT TO ENTER NEW D.E.'S (Y/N) ?"),
TEST : READ( ), FOR I THRU N DO (IF TEST = Y THEN
                                           I

RHS  : READ("ENTER RHS OF EQ. NO.", I, ",  D", X ,"/DT ="),

                                               I
EQ  : DIFF(X , T) = RHS , PRINT(EQ ),
  I         I          I         I

EQS : MAKELIST(EQ , I, 1, N)))
                 I

(E32) INP3( ) := (PRINT("INPUT NEAR-IDENTITY TRANSFORMATION
(USE PREV FOR PREVIOUS TRANSFORMATION)"),
FOR I THRU N DO (ROW : I, PREV : TR ,
                                   I

TR  : READ(X , "=", Y , "+ ?"), PRINT(X , "=", Y  + TR )),
  I         I        I                 I        I     I

TRANS : MAKELIST(X  = Y  + TR , I, 1, N))
                  I    I     I

(E33) INP4( ) := M : READ("ENTER TRUNCATION ORDER (HIGHEST
ORDER TERMS TO BE KEPT)")

(E34) SETUP( ) := FOR I THRU N DO DEPENDS([X , Y ], T)
                                           I   I

(E35) STEP1( ) := TEMP2 : TAYLOR(EV(EQS, TRANS, DIFF),
MAKELIST(Y , I, 1, N), 0, M)
          I

(E36) STEP2( ) := (FOR I THRU N
DO TEMP2 : SUBST(DUMMY , DIFF(Y , T), TEMP2),
                      I       I

TEMP3 : SOLVE(TEMP2, MAKELIST(DUMMY , I, 1, N)),
                                   I

FOR I THRU N DO TEMP3 : SUBST(DIFF(Y , T), DUMMY , TEMP3))
                                    I           I
```

```
(E37) STEP3( ) := (TEMP4 : TAYLOR(TEMP3,
MAKELIST(Y , I, 1, N), 0, M),
          I

FOR I THRU N DO PRINT(PART(TEMP4, 1, I)))

(E38) GEN(NN) := BLOCK(IF N > 2
 THEN (PRINT("GEN ONLY WORKS WITH 2 VARIABLES!"),
RETURN(0)) ELSE TEMPGEN : 0,
FOR II FROM 0 THRU NN DO (FOR JJ FROM 0 THRU NN
DO (IF II + JJ = NN THEN (AUX(II, JJ, ROW),

                                          II   JJ
TEMPGEN : TEMPGEN + Z                    Y    Y   ))), TEMPGEN)
                     II, JJ, ROW          1    2

(E39) AUX(II, JJ, ROW) := IF ROW = 1

    THEN Z             : A          ELSE Z             : B
          II, JJ, ROW     II, JJ          II, JJ, ROW     II, JJ

(E40) DECOMPOSE( ) := IF N = 2
THEN (FOR I THRU 2 DO NEWRHS  :
                            I

EXPAND(RHS(PART(TEMP4, 1, I))),
FOR I THRU 2 DO (FOR J FROM 0 THRU M
DO (FOR K FROM 0 THRU M DO (IF J + K = M

 THEN C         : COEFF(COEFF(NEWRHS , Y , J), Y , K)))),
       I, J, K                       I   1        2

PRINT("C[I,J,K] IS THE COEFF OF", Y , "**J", Y ,
                                  1          2

"**K IN THE I-TH EQUATION"), " ")

 ELSE PRINT("DECOMPOSE ONLY WORKS FOR 2 EQUATIONS")

(D40)                        DONE
```

The main function NF first asks the user if new variable names are to be entered. If the answer is affirmative, the function INP1 is called which reads the number of equations, N, and the variable names from the keyboard. Note that the old and new variables are respectively referred to as X[I] and Y[I] throughout the program, although they are displayed in the MACSYMA output by their user-defined symbols. Note also that in MACSYMA, X[I] is the same as X_I, the difference being that the former is the user-input version, while the latter is the MACSYMA-output version.

Next the function SETUP declares X[I] and Y[I] to be functions of T. The function INP2 then asks the user if new differential equations are to be entered. In the case of an affirmative reply, the right hand side of the ith equation, RHS[I], is read from the keyboard. The N differential equations are stored as a list called EQS. The transformation is then input via the function INP3 and is stored as a list called TRANS. Finally INP4 reads the truncation order M.

The input portion of the program having been completed, NF then calls STEP1, STEP2 and STEP3, which compute the transformed equations. STEP1 plugs the TRANSformation into the EQuationS, Taylor expands them, and calls the result TEMP2. STEP2 solves TEMP2 for the derivatives of the new variables, and stores the result as TEMP3. (In order to conveniently use the MACSYMA function SOLVE, the derivatives Y[I]' were replaced in TEMP2 by dummy variables.) STEP3 then Taylor expands TEMP3, stores the final result as TEMP4 and displays it. NF finishes up by asking the user if another transformation is to be entered, and if not, returns as its value the transformed equations TEMP4.

The auxiliary function GEN builds up a general homogeneous polynomial of arbitrary degree by use of nested FOR-DO loops. The auxiliary function DECOMPOSE is included as a convenience for isolating the coefficients of the transformed equations, TEMP4. As written, both GEN and DECOMPOSE only work for N=2 equations, but can obviously be extended to a larger number if desired.

CONCLUSION

Normal forms and center manifolds have received considerable attention lately in the applied mathematics literature (3,5,9). In addition to providing a method for investigating the stability of equilibrium (as illustrated in this work), these methods have also been used to investigate the bifurcation of equilibria and limit cycles, as well as to approximate rates of decay. Their use as a practical tool has been discouraged, however, by the great quantity of algebra involved. It is hoped that the availability of MACSYMA software to perform these calculations will increase their utility and popularity.

ACKNOWLEDGEMENT

This work was partially supported by Air Force grant # AFOSR-84-0051.

REFERENCES

1. Nayfeh, A., Perturbation Methods, John Wiley & Sons (1973)
2. Cesari, L., Asymptotic Behavior and Stability Problems in Ordinary Differential Equations, Third Edition, Springer-Verlag (1971)
3. Guckenheimer, J. and Holmes, P., Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer-Verlag (1983)
4. Minorsky, N., Nonlinear Oscillations, D. Van Nostrand Co. (1962)
5. Carr, J., Applications of Centre Manifold Theory, Springer-Verlag (1981)
6. Moon, F.C. and Rand, R.H., Parametric Stiffness Control of Flexible Structures, NASA Conference Proceedings: Workshop on Identification and Control of Flexible Space Structures, San Diego, May 1984
7. Takens, F., Singularities of Vector Fields, Publ. Math. Inst. Hautes Etudes Sci. 43, 47-100 (1974)
8. Rand, R.H., Computer Algebra in Applied Mathematics: An Introduction to MACSYMA, Pitman Publishing (1984)
9. Arnold, V.I., Geometrical Methods in the Theory of Ordinary Differential Equations, Springer-Verlag (1983)