Multivariate Power Series and Normal Form Calculation in *Mathematica*

Victor Edneral[1] and Raya Khanin[2]

[1] Institute for Nuclear Physics,
Moscow State University, Moscow, Russia
edneral@theory.sinp.msu.ru
[2] Department of Applied Mathematics and Theoretical Physics,
University of Cambridge
R.Khanin@damtp.cam.ac.uk

Abstract. This paper studies applications of the normal form method to dynamical systems using *Mathematica*. It describes the package for power series treatment, and the procedures for building a normal form and normalizing transformations. Examples of treatment of a crude one-dimension equation and a double pendulum system are given.

1 Introduction

The concept of using co-ordinate transformation to simplify ordinary differential equations has been widely used for a long time. The normal form method which has applications in many branches of engineering, physics, and applied mathematics is a powerful technique to simplify the governing ordinary differential equations. The basic idea of the method of normal form is the use of "local" coordinate transformation to simplify the equations that describe the dynamics of the system under consideration. In other words, with the method of normal forms, one seeks a near-identity coordinate transformation in which the dynamical system takes the simplest form or so-called normal form. The transformations are called local because they are generated in a neighbourhood of a known solution such as a fixed point or a periodic orbit of the system. For nonlinear dynamical systems the coordinate transformation will, in general, be nonlinear. The normalisation is usually carried out with respect to perturbation parameters. The normal forms of differential equations are also very useful in bifurcation analysis [1].

The method of normal forms dates back to the times of Euler, Poincaré, Dulac and Birkhoff. In this paper, algorithm based on the approach by Bruno [2] is employed. It is an algorithm for a resonant normal form creation. The important advantage of this approach is its general algorithmic frame, which allows one to investigate a wide class of autonomous systems including Hamilton and non- Hamilton cases. The algorithm provides a constructive way for obtaining the approximations of local families of periodic and conditional periodic solutions in the form of power (Fourier) series. Another advantage of the approach employed here is the simplicity of creation of the normal form and the corresponding transformations, which is implemented using recursive formula.

Examples of the lowest orders calculation of the normal forms in computer algebra necessary for bifurcation analysis has been done in [3]. A package for calculating coefficients of the normal form has been developed by one of the authors [4]. The package NORT is written in Standard LISP and contains about 2000 operators.

2 Preliminary Treatment of the System

Consider a system of autonomous ordinary differential equations:

$$\dot{\mathbf{x}} = \mathbf{\Phi}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector function of time, $\mathbf{\Phi}(\mathbf{x})$ is a vector function of $\mathbf{x}$ and other parameters. Such equations arise in numerous scientific, engineering and industrial applications. It is worth noting that a non-autonomous system of equations

$$\dot{\mathbf{x}} = \hat{\mathbf{\Phi}}(\mathbf{x}, t). \tag{2}$$

can be transformed into the autonomous system by introducing an additional variable $\theta = \theta(t)$ such that $\hat{\mathbf{\Phi}}(\mathbf{x}, \theta)$ becomes (explicitly) independent of t. $\theta(t)$ is defined by adding the equation $\dot{\theta} = \phi(\theta)$ to system (2). For example $\cos(\omega t)$ can be replaced by the sum $(\theta_1 + \theta_2)/2$ with an additional couple of equations: $\dot{\theta}_1 = i \cdot \omega \cdot \theta_1$ and $\dot{\theta}_2 = -i \cdot \omega \cdot \theta_2$.

The important condition on a system of ordinary differential equations (1) is that it is explicitly solved with respect to the first derivatives $\dot{\mathbf{x}}(\mathbf{t})$. Sometimes this can be done approximately.

Preanalysis. The second important condition on system (1) is that it is studied in the neighbourhood of stationary point $\mathbf{x_0}$, $\mathbf{\Phi}(\mathbf{x_0}) = 0$. It is convenient to shift variables by $-\mathbf{x_0}$, so that $\mathbf{\Phi}(\mathbf{0}) = 0$. Sometimes the original system has no stationary points or it is important to investigate it in the domain which is far from such points. In the neighbourhood of regular points other methods should be used (see, for example, [5]). Usually, the right-hand side of the system is approximated by polynomials. For analytical function, it is simply truncated power series of this function. Finally, the linear part of the right-hand side is transformed to Jordan's form so that the original system (1) is represented by

$$\dot{y}_i = \lambda_i y_i + \sigma_i y_{i-1} + \tilde{\Phi}_i(\mathbf{y}), \quad \sigma_1 = 0, \quad i = 1, \ldots n \tag{3}$$

Here $\Lambda = (\lambda_1, \ldots, \lambda_n)$ is the vector of eigenvalues of the matrix of the linear part of the system and $\tilde{\Phi}$ is the vector of polynomials obtained from approximating $\mathbf{\Phi}$ by series of polynomials. $\tilde{\Phi}$ does not contain constant or linear terms. Constant terms have been removed by shifting into the stationary point while linear terms are in the linear part of the system, which is written in Jordan form.

Normal form transformation. For the clarity of exposition let us consider pure diagonal systems. (The general normal form method can be applied to a full Jordan case as well.) Equations (3) can be written in the form:

$$\dot{y}_i = \lambda_i y_i + y_i \sum_{\mathbf{q} \in N_i} f_{i,\mathbf{q}} \mathbf{y}^{\mathbf{q}}, \quad i = 1, \ldots, n\,, \tag{4}$$

where we use the multi-index notation:

$$\mathbf{y}^{\mathbf{q}} = \prod_{j=1}^{n} {y_j}^{q_j},$$

with the power exponent vector $\mathbf{q} = (q_1, \ldots, q_n)$ and the sets:

$$N_i = \{\mathbf{q} \in \mathbb{Z}^n : q_i \geq -1 \text{ and } q_j \geq 0\,,\ \text{if } j \neq i\,, \quad j = 1, \ldots, n\}\,,$$

because the factor y_i has been moved out of the sum in (4).

The normalization is done with a near-identity transformation:

$$y_i = z_i + z_i \sum_{\mathbf{q} \in N_i} h_{i,\mathbf{q}} \mathbf{z}^{\mathbf{q}}, \quad i = 1, \ldots, n \tag{5}$$

and then equation (4) will have the form:

$$\dot{z}_i = \psi_i(\mathbf{z}) \stackrel{\text{def}}{=} \lambda_i z_i + z_i \sum_{\substack{\langle \mathbf{q}, \mathbf{\Lambda} \rangle = 0 \\ \mathbf{q} \in N_i}} g_{i,\mathbf{q}} \mathbf{z}^{\mathbf{q}}, \quad i = 1, \ldots, n\,. \tag{6}$$

The important difference between (4) and (6) is a restriction on the range of the summation which is defined by the equation:

$$\langle \mathbf{q}, \mathbf{\Lambda} \rangle \stackrel{\text{def}}{=} \sum_{j=1}^{n} q_j \lambda_j = 0\,. \tag{7}$$

The h and g coefficients in (5) and (6) are found by using the recurrence formula:

$$g_{i,\mathbf{q}} + \langle \mathbf{q}, \mathbf{\Lambda} \rangle \cdot h_{i,\mathbf{q}} = -\sum_{j=1}^{n} \sum_{\substack{\mathbf{p} + \mathbf{r} = \mathbf{q} \\ \mathbf{q} \in N_i}} (p_j + \delta_{ij}) \cdot h_{i,\mathbf{p}} \cdot g_{j,\mathbf{r}} + \tilde{\Phi}_{i,\mathbf{q}}\,, \tag{8}$$

where the second summation in the right-hand side is over all integer vectors satisfying the constraint $\mathbf{p}+\mathbf{r}=\mathbf{q}$, and $\tilde{\Phi}_{i,\mathbf{q}}$ is a coefficient of the factor $z_i\mathbf{z}^{\mathbf{q}}$ in the polynomial $\tilde{\Phi}_i$ in (3), arguments of which have been transformed by (5). Here $||\mathbf{p}||$ and $||\mathbf{r}|| < ||\mathbf{q}||$, where $||\mathbf{q}|| \stackrel{\mathrm{def}}{=} q_1+\ldots+q_n$.

The ambiguity in (8) is usually fixed by the conventions:

$$\begin{aligned} h_{i,\mathbf{q}} &= 0, \qquad \text{if} \quad \langle \mathbf{q}, \mathbf{\Lambda}\rangle = 0, \\ g_{i,\mathbf{q}} &= 0, \qquad \text{if} \quad \langle \mathbf{q}, \mathbf{\Lambda}\rangle \neq 0 \end{aligned} \tag{9}$$

and then the normalizing transformation is called a "basic".

3 Package for Working with Series

As one of the steps in the preliminary analysis involves approximating right-hand side by polynomials, one needs an effective tool for dealing with power series. Surprisingly, *Mathematica* lacks tools for working with multivariate power series. To fill this gap, a package *PolynomSeries* has been developed by the authors. The name of the package arises from an idea of a transformation between polynomial and specific power representation in *Mathematica*. Below is a brief description of the functionalities of the package. Consider a multivariate polynomial

```
p = Plus@@Flatten[Table[y1^i*y2^j*y3^k,
                 {i, 0, 1}, {j, 1, 2}, {k, 1, 3}]]
```

$$y2\ y3 + y1\ y2\ y3 + y2^2\ y3 + y1\ y2^2\ y3 + y2\ y3^2 + y1\ y2\ y3^2 + y2^2\ y3^2 + y1\ y2^2\ y3^2$$

Transforming the polynomial into a series with respect to variables y_1, y_2, y_3 yields:

```
R = Polynom2Series[p, {y1, y2, y3}]
```

$$\{\{2, y2\ y3\},\{3, y1\ y2\ y3 + y2^2\ y3 + y2\ y3^2\}, \{4, y1\ y2^2\ y3 + y1\ y2\ y3^2 + y2^2\ y3^2\}, \{5, y1\ y2^2\ y3^2\}\}$$

The above list, R, is the main object of the package. Each element of R consists of two entries: the first one is defined as "index". It is the common power $\alpha = \alpha_1 + \alpha_2 \ldots \alpha_n$ of the monomial $y_1^{\alpha_1} y_2^{\alpha_2} \ldots y_n^{\alpha_n}$. The monomial itself is the second entry. Here y_j are the elements of the second input parameter of the function. One can consider transformation of the polynomial into the series with respect to some (not all) variables (y_2, y_3 below):

```
Polynom2Series[p, {y1, y3}]
```

$$\{\{1, y2\ y3 + y2^2\ y3\}, \{2, y1\ y2\ y3 + y1\ y2^2\ y3 + y2\ y3^2 + y2^2\ y3^2\}, \{3, y1\ y2\ y3^2 + y1\ y2^2\ y3^2\}\}$$

Several arithmetic operations on series of type R are defined in the package, including addition of two series (**AddR[R1, R2]**), multiplying two series and retaining terms up to order (index) (**MultRR[R1, R2, n]**) or returning only terms of index n (**MultRRLayer[R, R1, n]**), ordering series by their indices (**OrderR[R]**)

It is often in the applications that one needs to substitute a new series for some (or all) variables in the given series. This can be done using **SubstR[R, RV, xvar, yvar, n]** command. Here **RV** is a vector of series with respect to new variables **yvar** which are substituted into the series **R** with respect to variables **xvar**. To substitute a simple series $(z1+z2, z2+z3, z1+z3)$ to every $(y1, y2, y3)$ in the series **R**, a vector of series substitutions is to be generated first:

```
subst = {z1 + z2, z2+z3, z1+z3};
RVy = MapThread[Polynom2Series[#1, #2]&,
{subst, {{z1, z2, z3}, {z1,z2,z3}, {z1,z2,z3}}}]
```

$$\{\{\{1, z1+z2\}\}, \{\{1, z2+z3\}\}, \{1, z1+z3\}\}\}$$

```
Rxy = SubstR[R, RVy, {y1, y2, y3}, {z1, z2, z3}, 3]
```

$$\begin{aligned} &\{\{2, z1z2 + z1z3 + z2z3 + z3^2\}, \{3, 2z1^2z2 + 2z1z2^2 + 2z1^2z3 + 6z1z2z3 \\ &+ 2z2^2z3 = 4z1z3^2 + 4z2z3^2 + 2z3^3\}\} \end{aligned} \tag{10}$$

Note that we only retain terms up to index 3. If the user is interested only in terms of index, say, 4, SubstRLayer[ ] is to be used:

```
Rxy = SubstRLayer[R1, RVy, {y1, y2, y3}, {z1, z2, z3}, 4]}
```

$$\{\{4, z1^3z2 + 3z1^2z2^2 + z1z2^3 + z1^3z3 + 7z1^2z2z3 + 7z1z2^2z3 + z2^3z3 + 4z1^2z3^2 + 10z1z2z3^2 + 4z2^2z3^2 + 4z1z3^3 + 4z2z3^3 + z3^4\}\} \tag{11}$$

There are separate commands for two commonly used functions, which are singular in zero:square root of series and inverse function. The correct usage of these commands can only be guaranteed if the series contains a constant term (i.e. term with index 0). To find square root with terms up to index 4 use

```
SqrtR[R, 4]
```

$$\{\{0, \sqrt{7}, \{1, x12\ \sqrt{7}\}, \{2, 27\ x1^256\ \sqrt{7}\}, \{3, 365\ x1^3784\ \sqrt{7}\}, \{4, 19763\ x1^443904\ \sqrt{7}\}\}$$

Inverse series $(1/R)$ is found by

```
InverseR[R, 4]
```

$$\{\{0, 17\}, \{1, -x149\}, \{2, -6\ x1^2343\}, \{3, -36\ x1^32401\}, \{4, -216\ x1^416807\}\}$$

4 *Mathematica* NormalForm Package

The procedures for building a normal form and a normalizing transformation till the fixed order are implemented in the package *NormalForm*. It includes procedures for a linear normalization, for a nonlinear normalization and for evaluating a periodicity condition. The input for investigation is the system of ODEs.

The package was tested by comparing the results for approximate solutions of Duffing's, van-der-Pol's, Henon-Heiles' systems with solutions obtained before by V.Edneral with the LISP based package NORT [4, 7]. The results are fully identical.

4.1 Example of One-Dimensional System

To demonstrate the main functions of the package, let us consider a trivial example of a one-dimensional crude system:

$$\dot{y} = -y + y^2. \tag{12}$$

This equation has an exact solution

$$y(t) = \frac{c \cdot \exp(-t)}{1 + c \cdot \exp(-t)}, \tag{13}$$

where c here is an integration constant.

Now let us calculate an approximation of this solution by the normal form method using the package *NormalForm*. The equation (12) is entered in standard form:

$$Eqs = \{\dot{y}[t] == -y[t] + y[t]^2\};$$

The first step of a linear normalization is performed by the procedure **LinearNormalization[Eqs_List, OVars_List, NVars_List]**. Here **Eqs** is a list of equations resolved with respect to the first derivatives of variables, **OVars** (original variables) and **NVars** is a list of the new variables:

```
{Atr, Btr, Jrd, NonLinV} = LinearNormalization[Eqs, {y[t]}, {u[t]}]
```

$$\{\{1\}, \{1\}, \{-1\}, \{u[t]^2\}\}$$

The result is four matrices: **Atr** and **Btr** are diagonalizing (right and inverse) matrices and Jrd is a matrix of the linear part in Jordan's form, i.e. **Jrd = Atr . LinearPart . Btr**, where **LinearPart** is a matrix of coefficients of the linear part of the original system. **NonLinV** is a vector of nonlinear parts in new variables

$$u[t] = Atr.\{y[t]\}.$$

The system is in a diagonal form already, so we have an identical transformation. Now we look at the linear part matrix. It is a 1 by 1 matrix and it has a single element $-y$, i.e. the eigenvalues vector is -1.

Because scalar production (7) can not be zero in this case, the right hand side of the normalized equation (normal form) will be linear:

$$\dot{z}[t] = -z[t]. \tag{14}$$

It is remarkable that we can write down the normal form without any calculations. It can be done for any eigenvalues for which equation (7) has finite numbers of solutions in natural integers. Such cases are called crude ones. Let us use the *NormalForm* procedure for the corresponding calculation:

```
{GRv, uN} = NormalForm[Jrd, NonLinV, u, z, 3]
```

$$\{\{-1\}, \{z1[t] - z1[t]^2 + z1[t]^3 - z1[t]^4\}\} \tag{15}$$

The result above (15) consists of a vector of the normalized right hand side **GRv** and a normalizing transformation **uN** (till the third order over identical transformation, i.e. $3 + 1 = 4$) such that the normalized equation has the form:

$$\dot{z}_i[t] = GRv_i \cdot z_i[t], \quad i = 1, \ldots, n \tag{16}$$

$$y_i[t] = \sum_{j=1,n} Atr_{i,j} u_j[t] = \sum_{j=1,n} Atr_{i,j} uN_j, \quad i = 1, \ldots, n \tag{17}$$

Here n is a dimension of the system, which is 1 in our case.

Solution of (14),(15), and (16) is $z(t) = c \cdot \exp(-t)$. For an approximation of a solution of (12) we need now to substitute this solution into the corresponding normalizing transformation of the type (5,15,17). After that we will have a truncated expansion of (13) in the series of constant c:

$$y[t] \approx c \cdot \exp(-t) - [c \cdot \exp(-t)]^2 + [c \cdot \exp(-t)]^3 - [c \cdot \exp(-t)]^4$$

4.2 Example of Four-Dimensional System. Double Pendulum

Now consider a fourth-order system which describes the equations of motion for the double pendulum:

$$\begin{aligned}
&(\frac{m_1}{3} + m_2)l_1^2 + m_2 l_1 l_2 \cos\theta_2 + \frac{m_2 l_2^2}{3})\ddot{\theta}_1 + (\frac{m_2 l_1 l_2}{2}\cos\theta_2 + \frac{m_2 l_2^2}{3})\ddot{\theta}_2 \\
&\quad = \frac{m_2 l_1 l_2}{2}(\dot{\theta}_1 + \dot{\theta}_2)^2 \sin\theta_2 - \left(\frac{m_1}{2} + m_2\right) g l_1 \sin\theta_1 \\
&\quad - \frac{m_2 l_1 l_2}{2}\dot{\theta}_1^2 \sin\theta_2 - \frac{m_2 g l_2}{2}\sin(\theta_1 + \theta_2).
\end{aligned}$$

$$(\frac{m_2 l_1 l_2}{2}\cos\theta_2 + \frac{m_2 l_2^2}{3})\ddot{\theta}_1 + \frac{m_2 l_2^2}{3}\ddot{\theta}_2 = -\frac{m_2 l_1 l_2}{2}\dot{\theta}_1^2 \sin\theta_2 - \frac{m_2 g l_2}{2}\sin(\theta_1 + \theta_2).$$

Equations in this form have been created by the automatic equation generator by D.Forehand *et al* [8]. This system is not initially solved with respect to derivatives. Assuming that ϕ and θ are small and approximating *Sin* up to the third order, a system in polynomials with respect to unknown variables is obtained (not shown). Solving the system with respect to higher derivatives and expanding the denominators in the right hand side in the power series of ϕ and θ till the third order yields the following right-hand sides for the system of the fourth order with respect to new variables $\phi1[t] = \phi[t]$, $\phi2[t] = \phi'[t]$, $\theta1[t] = \theta[t]$, $\theta2[t] = \theta'[t]$:

$$\begin{aligned}
&NewEqs \;=\; \Big\{ \qquad (18)\\
&\theta_2'[t] == -\frac{18\theta_1[t]}{7} + \frac{183\theta_1[t]^3}{49} - \frac{9}{7}\theta_1[t]\theta_2[t]^2 + \frac{9\phi_1[t]}{7} - \frac{873}{98}\theta_1[t]^2\phi_1[t] \\
&\quad + \frac{9}{7}\theta_2[t]^2\phi_1[t] + \frac{387}{49}\theta_1[t]\phi_1[t]^2 - \frac{123\phi_1[t]^3}{49} - \frac{6}{7}\theta_1[t]\phi_2[t]^2 + \frac{6}{7}\phi_1[t]\phi_2[t]^2 \\
&\phi_2'[t] == \frac{27\theta_1[t]}{7} - \frac{369\theta_1[t]^3}{49} + \frac{24}{7}\theta_1[t]\theta_2[t]^2 - \frac{24\phi_1[t]}{7} + \frac{891}{49}\theta_1[t]^2\phi_1[t] \\
&\quad - \frac{24}{7}\theta_2[t]^2\phi_1[t] - \frac{1539}{98}\theta_1[t]\phi_1[t]^2 + \frac{244\phi_1[t]^3}{49} + \frac{9}{7}\theta_1[t]\phi_2[t]^2 - \frac{9}{7}\phi_1[t]\phi2[t]^2 \\
&{\theta_1}'[t] == \theta_2[t], \\
&{\phi_1}'[t] == \phi_2[t]\Big\}
\end{aligned}$$

An assumption that the lengths and masses of two pendulums are equal $l21 = 1, m21 = 1$ has been made to simplify computations: The system is then recast in the Jordan form. Float point arithmetic was used for calculation of radicals in eigenvalues. The calculation of the normal form up to the 5th order yields a vector of the normalized right-hand sides:

$$\begin{aligned}\{\, & i + 0.13\, iz_1[t]z_2[t] - 0.08\, iz_1[t]^2 z_2[t]^2 - 0.08\, iz_3[t]z_4[t] \\ & + 0.21\, iz_1[t]z_2[t]z_3[t]z_4[t] - 3.65\, iz_3[t]^2 z_4[t]^2, \\ & 0.85\, i - 0.13\, iz_1[t]z_2[t] + 0.08\, iz_1[t]^2 z_2[t]^2 + 0.08\, iz_3[t]z_4[t] \\ & - 0.21\, iz_1[t]z_2[t]z_3[t]z_4[t] + 3.65\, iz_3[t]^2 z_4[t]^2, \\ & -2.29\, i - 0.30\, iz_1[t]z_2[t] + 1.52\, iz_1[t]^2 z_2[t]^2 + 7.87\, iz_3[t]z_4[t] \\ & + 6.22\, iz_1[t]z_2[t]z_3[t]z_4[t] + 71.86\, iz_3[t]^2 z_4[t]^2, \\ & 2.29\, i + 0.3\, iz_1[t]z_2[t] - 1.52\, iz_1[t]^2 z_2[t]^2 - 7.87\, iz_3[t]z_4[t] \\ & - 6.22\, iz_1[t]z_2[t]z_3[t]z_4[t] - 71.86\, iz_3[t]^2 z_4[t]^2 \,\}\end{aligned}$$

In new variables $z1[t], \ldots, z4[t]$ the system has the form (6). After the standard treatment of the normalized system [4] we get approximations for conditionally periodic families of solutions, where $a, b, t1, t2$ are integration constants below (the result is truncated at order 3 in small a, b):

$$\begin{aligned}\theta[t] = & \\ & 1.4a\cos[(t+t_1)\omega_1] + 0.09a^3\cos[(t+t_1)\omega_1] - 1.80ab^2\cos[(t+t_1)\omega_1] \\ & + 0.65a^3\cos[3(t+t_1)\omega_1] \\ & - 0.95b\cos[(t+t_2)\omega_2] + 0.07a^2b\cos[(t+t_2)\omega_2] - 1.65b^3\cos[(t+t_2)\omega_2] \\ & + 0.8b^3\cos[3(t+t_2)\omega_2] \\ & + 0.8ab^2\cos[t\omega_1 + t_1\omega_1 - 2t\omega_2 - 2t_2\omega_2] - 0.24a^2b\cos[2t\omega_1 + 2t_1\omega_1 - t\omega_2 - t_2\omega_2] \\ & + 0.61a^2b\cos[2t\omega_1 + 2t_1\omega_1 + t\omega_2 + t_2\omega_2] + 0.65ab^2\cos[t\omega_1 + t_1\omega_1 + 2t\omega_2 + 2t_2\omega_2]\end{aligned} \tag{19}$$

$$\begin{aligned}\phi[t] = & \\ & 2.a\cos[(t+t_1)\omega_1] + 0.18a^3\cos[(t+t_1)\omega_1] + 3.57ab^2\cos[(t+t_1)\omega_1] \\ & - 1.39a^3\cos[3(t+t_1)\omega_1] \\ & + 2.b\cos[(t+t_2)\omega_2] - 0.12a^2b\cos[(t+t_2)\omega_2] + 3.39b^3\cos[(t+t_2)\omega_2] \\ & - 1.59b^3\cos[3(t+t_2)\omega_2] \\ & - 2.16ab^2\cos[t\omega_1 + t_1\omega_1 - 2t\omega_2 - 2t_2\omega_2] + 3.28a^2b\cos[2t\omega_1 + 2t_1\omega_1 - t\omega_2 - t_2\omega_2] \\ & - 1.12a^2b\cos[2t\omega_1 + 2t_1\omega_1 + t\omega_2 + t_2\omega_2] - 0.97ab^2\cos[t\omega_1 + t_1\omega_1 + 2t\omega_2 + 2t_2\omega_2]\end{aligned} \tag{20}$$

where frequencies are given by the following formulas

$$\omega 1 = 0.85 + 0.08a^4 + 0.08b^2 + 3.65b^4 + a^2\big(-0.13 - 0.21b^2\big) \tag{21}$$

$$\omega 2 = 2.3 - 1.52a^4 - 7.87b^2 - 71.86b^4 + a^2\big(0.30 - 6.22b^2\big) \tag{22}$$

To generate the periodicity condition ([2], [4]) the procedure

ConditionForPeriodicity[GRv, Jrd, Var_List]

is employed. The meaning of **GRv** and **Jrd** arguments are explained in the previous section and **Var** is a list of all small variables. In this example it is $\{z1[t], z2[t], z3[t], z4[t]\}$. Approximations (19) are conditionally periodic and they formally satisfy equation (18), but the procedure **ConditionForPeriodicity** results in the condition for an orbital stability. A separate paper discussing this subject is in preparation. In the case of the double pendulum case, this procedure results in a condition:

$$\begin{gathered}-0.564279z1[t]z2[t] + 1.48528z1[t]^2z2[t]^2 + 0.579792z1[t]^3z2[t]^3 + \\ 6.91055z3[t]z4[t] + 4.84095z1[t]z2[t]z3[t]z4[t] + 58.0101z1[t]^2z2[t]^2z3[t]z4[t] + \\ 69.8654z3[t]^2z4[t]^2 + 268.305z1[t]z2[t]z3[t]^2z4[t]^2 + 746.679579792z3[t]^3z4[t]^3 = 0\end{gathered} \tag{23}$$

This condition can be solved in a, b constants giving:

$$a = b(3.49953 + 89.1055b^2 + 8868.38b^4).$$

5 Summary

The authors revisit the normal form method and discuss its implementation in *Mathematica*. The high order normal form for systems of ordinary differential equations can be calculated.

A separate package enabling one to work with truncated multivariate power series is discussed. The demo version of *PolynomialSeries* package can be accessed on *www.mathsource.com*. Authors are going to develop new features of the package. The power series package should use the $O(x^n)$ mechanism. Truncation of power series allows one to control the length of truncated series by a correct way as it is done in *Maple*. Therefore, an existing version is enough for a support of a normal form method. An important procedure for solving equations in power series is planned to be added to the package [6].

The method produces approximate formulas for the approximation of crude and periodic solutions of autonomous nonlinear ODEs. The comparison of *Mathematica* package with an earlier version of normal form package (NORT) written in LISP demonstrates that the calculations within the *Mathematica* system are more flexible and convenient but are considerably slower than under the LISP.

6 Acknowledgement

This research was supported by London Mathematical Society grant 02-5717 and the RFBR program for supporting leading scientific schools, grant 00-15-96-577. Raya Khanin is Dorothy Hodgkin Royal Society Research Fellow.

References

1. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields. Springer–Verlag, New York (1986)
2. Bruno, A.D.: Local Method in Nonlinear Differential Equations, Springer-Verlag, Berlin (1989)
3. Rand, R., Armbruster, D.: Perturbation Methods, Bifurcation Theory and Computer Algebra. Springer–Verlag, New York (1987)
4. Edneral, V.: *About Normal Form Method.* Proc. Second Workshop on Computer Algebra in Scientific Computing (CASC'99, Munich, Germany, 1999), V.G. Ganzha *et al.* (Eds.), Springer-Verlag, Berlin (1999) 51–66
5. Dupée, B.J., Davenport, J.H.: *An Automatic Symbolic-Numeric Taylor Series ODE Solver.* Proc. Second Workshop on Computer Algebra in Scientific Computing (CASC'99, Munich, Germany, 1999), V.G. Ganzha *et al.* (Eds.), Springer-Verlag, Berlin (1999) 37–50
6. Edneral, V.F., Vassiliev, N.N.: *Approach to solving equations in the ring of formal power series.* Proc. XIIth Workshop on High Energy Physics and Quantum Field Theory (Samara, Russia, 1997), B.B.Levtchenko (Ed.), Moscow (1999) 462–465
7. Edneral, V.F.: *A symbolic approximation of periodic solutions of the Henon–Heiles system by the normal form method.* Math. Comp. Simul. **45** (1998) 445–463
8. Forehand, D.I.M., Cartmell, M.P., and Khanin, R.: *Initial development towards an integrated symbolic-analytical multibody code* (submitted)