A Mathematica package to derive
weight-homogeneous decompositions of vector fields

Radoslaw Antoni Kycia
University of Warsaw
Faculty of Mathematics, Informatics and Mechanics
Banacha 2, Warsaw, 02-097, Poland
E-mail: rkycia@mimuw.edu.pl

Abstract. Description of a Mathematica package that performs weight-homogeneous
decompositions of polynomial vector fields will be presented. The package has also functions
that compute the Kovalevskaya matrix and exponents. Some details of implementation and
examples will be also outlined. It can be used in the testing of integrability of dynamical
systems described by vector fields as well as in the Painlevé analysis.

1. Introduction
Weight-homogeneous decomposition of a polynomial vector field is base of many methods that
allows to obtain some results on the nature of solutions of ordinary differential equations (ODEs),
e.g., it is an ingredient of the Painlevé test formulated in [3]. The decomposition can also help
to deduce the structure of first integrals. It is a purely combinatorial task. For vector fields
that contain small number of monomials in small number of dimensions it can be done by hand,
however, when a space dimension is large and a vector field contains many terms it becomes a
task in which Computer Algebra Systems (CAS) like Mathematica [6] should be used.
There exist very general packages for Mathematica, Maple or Maesyma CAS to perform
the Painlevé analysis (see, [3], [4], [1], [7] and the references therein) which do the first step of
weight-homogeneous decomposition, i.e., finding the scale-invariant terms, as a part of algorithm,
however, they do not do full decomposition. In this approach the vector structure of the field
is preserved which is additional advantage. As far as the author know there is no algorithms
which are based on this approach. Therefore, it is important to provide implementation of a
method of the decomposition which can serve as a building block for future use.
This paper describes the set of functions in Mathematica CAS [6] language that assist user
in weight-homogeneous decomposition. It is organized as follows. In the next section the theory
and simple use of the package will be presented. Then details of implementation will be outlined.
In Conclusions we provide possible applications and plans for future development.

2. Theory and simple example
The presentation of the theory will be based on [3]. Assume that the space dimension is n,
i.e., the vector of coordinates is of the form \( x = [x_1, \ldots, x_n] \). Take a vector of integer numbers,
the weights \( w \in \mathbb{Z}^n \). The following convention about taking powers with respect to the vector
simplifies notation \( x^w := [x_1^{w_1}, \ldots, x_n^{w_n}] \) for the vector \( x \) and \( t^w := [t_1^{w_1}, \ldots, t_n^{w_n}] \) for a scalar \( t \). A function \( f(x) \) has a weight degree \( d \) with respect to the weight vector \( w \) if

\[
f(x_1 t_1^{w_1}, \ldots, x_n t_n^{w_n}) = t^d f(x_1, \ldots, x_n).
\]

The weight degree is usually denoted by \( \text{deg}(f, w) := d \). If \( \forall i \in \{1, \ldots, n\} \ w_i = 1 \) then the classical definition of the homogeneous function is restored. The definition naturally extends to vector fields, i.e., a field has the weight degree \( d \) with respect to the weight vector \( w \) if every component of this field vector has this property.

Starting point for a weight-homogeneous decomposition is a scale invariant vector field. It is a weight-homogeneous vector field \( f = [f_1, \ldots, f_n] \) which components have weights that fulfills

\[
\forall i \in \{1, \ldots, n\} \quad \text{deg}(f_i, w) = w_i - 1.
\]

For polynomial vector fields there can exist many scale-invariant terms. By fixing one of the possibilities the vector field can be expanded in the following form

\[
f = f^{(0)} + f^{(1)} + \ldots + f^{(p)},
\]

where the terms are defined as follows

\[
f_i^{(j)}(t^w x) = t^{w_i - q^{(j)} - 1} f_i^{(j)}(x), \quad i \in \{1, \ldots, n\}; \quad \forall t \in \mathbb{C},
\]

\[
0 < q^{(j)} < q^{(i)} \quad \forall i < j,
\]

where \( f^{(0)} \) is a scale-invariant term that determines the decomposition. Every scale-invariant term generates one of the weight-homogeneous decomposition. Determination of a scale-invariant term and selection of one of the corresponding dominant balances defines the Kovalevskaya matrix

\[
K := D f^{(0)}(\alpha) - \text{diag}(w),
\]

which is defined as the Jacobian of the scale-invariant term derived on the scale-invariant solution minus diagonal matrix with weights on its diagonal. If \( K \) is semi-simple, i.e., diagonalizable then
its eigenvalues are called the Kovalevskaya exponents. It is a fact [3] that one of the exponents is always $-1$. It corresponds to arbitrariness of choice of initial 'time'. The values of the Kovalevskaya exponents have important meaning in the Painlevé test and in determining the class of possible first integrals [3].

All above quantities can be calculated using the package. The use of it will be presented below on a simple example. The package can be downloaded from [5].

Consider the planar Lotka-Volterra system of ODEs [3]

\[
\begin{align*}
\dot{x} &= x(ay - x), \\
\dot{y} &= y(bx - y),
\end{align*}
\]  

(8)

where $a, b$ are real parameters. In order to find all possible scale invariant terms we use \texttt{AnalyzeScaleInvariantTerms[equations, variables]} function. It requires the list of RHS of equations and corresponding variables in order in which the system is arranged. It is useful to define new variables in Mathematica that keep these values, i.e.,

\[
\text{equation} = \{x (a*y - x), y*(b*x - y)\}; \\
\text{variables} = \{x, y\}; \\
\text{sol} = \text{AnalyzeScaleInvariantTerms[equation, variables]};
\]

PrintScaleInvariantTerms[sol];

The variable \texttt{equation} is substituted by the list of RHS of equations, the variable \texttt{variables} contains the list of variables in the order indicated by the order of LHS of the equations and \texttt{sol} keeps the result of analysis of scale-invariant terms. It can be printed in a user-friendly form by \texttt{PrintScaleInvariantTerms[solution]}. As the result three terms will be obtained with the weights

\[
\begin{align*}
\text{f}^{(0)} &= \begin{bmatrix} -x^2 \\ bxy \end{bmatrix}, \\
w &= \begin{bmatrix} w_1 \rightarrow -1 \\ w_2 \rightarrow \text{anything} \end{bmatrix}; \\
\text{f}^{(0)} &= \begin{bmatrix} -x^2 + axy \\ bxy - y^2 \end{bmatrix}, \\
w &= \begin{bmatrix} w_1 \rightarrow -1 \\ w_2 \rightarrow -1 \end{bmatrix}; \\
\text{f}^{(0)} &= \begin{bmatrix} axy \\ -y^2 \end{bmatrix}, \\
w &= \begin{bmatrix} w_1 \rightarrow \text{anything} \\ w_2 \rightarrow -1 \end{bmatrix};
\end{align*}
\]  

(9)

where \texttt{anything} means arbitrary value. One can also note that the second solution is a special case of the two remaining ones where arbitrary value of the exponents $w_i$ were set to $-1$. The problem of arbitrariness of some values of the exponents will be addressed below.

The coefficients $\alpha$ for scale-invariant solutions (3) of (9) can be computed with the help of \texttt{CalculateScaleInvariantSolution[solution, variables]}. Continuing the example, if we execute

\[
\text{sol} = \text{CalculateScaleInvariantSolution[sol, variables]};
\]

PrintScaleInvariantSolutions[sol];

we obtain the list of all values of $\alpha$ that are solutions of (4). Some of scale-invariant solutions do not describe singular solutions and they can be removed by using \texttt{RemoveNonsingular[sol, variables]}. If one is interested in terms which do not contain arbitrary values of weights then these terms can be selected by \texttt{RemoveArbitrary[sol, variables]}.

In the next step the Kovalevskaya exponents can be calculated and printed using the following set of instructions

\[
\text{km} = \text{KovalevskayaMatrix[sol, variables]};
\]

PrintScaleInvariantKovalevskaya[km];
The final task is to compute remaining terms of the decomposition using \( \text{WeightHomogenousDecompose} [\text{solution}, \text{system}, \text{variables}, \text{solNumber}] \), where \( \text{solution} \) is previously obtained the set of scale-invariant terms (\( \text{sol} \) variable in the above example), \( \text{system} \) and \( \text{variables} \) are RHS of the system and corresponding variables and \( \text{solNumber} \) is the index of weight and scale-invariant term which will be treated as the weight and the initial term of the expansion, e.g., the command

\[
\text{WeightHomogenousDecompose} [\text{sol, equation, variables, 1}];
\]

produces the remaining terms of the decomposition and corresponding \( q \) values for the first scale-invariant term of (9).

The function that performs all the above steps is \( \text{WeightHomogenousAnalysis} [\text{system, variables}] \). For complicated vector fields the output will be long and the step by step method with elimination of unwanted terms may be more preferred by the user.

Application-oriented reader can skip the next section in which details of implementation will be discussed.

3. Details of implementation

This section may be interesting for those who want to alter or adjust the package to their needs.

Firstly, the algorithm for the searching of scale-invariant terms will be discussed. It consists of a few steps:

(i) Extract monomials with respect to the variables in a form of the lists for every components of the vector field and add 0 to every monomial list.

(ii) Create all possible combinations of monomials from different components/lists. Exclude zero vector from the list of combinations.

(iii) Substitute \( x \to t^w \) for variables.

(iv) Form equations (4).

(v) Solve the equations.

(vi) Remove terms which have no solutions for exponents.

(vii) Simplify solutions for exponents with arbitrary parameters by back substitution and join the terms which have the same weights.

(viii) Create new terms which are terms with contains arbitrary exponents that were substituted by the solutions from other terms - create special cases of a solution with arbitrary parameters. Then join the terms which have the same weights.

(ix) Join terms that one is a special case of the other.

(x) Repeat vii to join the terms introduced in the previous two steps.

The step vii is a necessary because Mathematica \( \text{Solve}[] \) function used for solving the system for exponents produce sometimes output of the form

\[
\begin{bmatrix}
  w_1 & \to & 1 \\
  w_2 & \to & 1 - w_1
\end{bmatrix},
\]

which can be obviously simplified by substituting \( w_1 \) form the first rule into the RHS of the second one. In order to explain the step viii consider two terms

\[
\begin{bmatrix}
  w_1 & \to & -\frac{2}{3} \\
  w_2 & \to & -\frac{4}{3} \\
  w_3 & \to & -\frac{5}{3} \\
  w_4 & \to & -\frac{1}{3}
\end{bmatrix},
\quad
\begin{bmatrix}
  w_1 & \to & \text{anything} \\
  w_2 & \to & -\frac{2}{3} \\
  w_3 & \to & -1 + w_1 \\
  w_4 & \to & -\frac{5}{3}
\end{bmatrix},
\]

\[(11)\]
It is obvious that the second vector of rules is a special case \( (w_1 = -\frac{2}{3}) \) of the first one and the polynomial associated with this general term can be added to the polynomial associated with the specialized term. It is performed by substitution of weights of special cases to the general ones, which create new terms, and then join distinct components of the vector fields with the same weights. In the step ix the algorithm joins the terms which have weights with the value \( \text{anything} \) and are that are 'orthogonal', which is illustrated by the following two terms

\[
\begin{bmatrix}
w_1 \rightarrow \text{anything} \\
w_2 \rightarrow 1
\end{bmatrix},
\begin{bmatrix}
w_1 \rightarrow 2 \\
w_2 \rightarrow \text{anything}
\end{bmatrix}.
\]

They produce more specific term with \( w_1 = 2, w_2 = 1 \), which generates the polynomial vector which is the sum of the polynomial vectors associated with both more general weights.

All these tasks are performed also when finding the scale-invariant terms by hand. The result of the function is a list that every entity is a list that contains a scale-invariant term as a first element and a vector of weights as a second component. The procedure that calculates the scale invariant solutions uses \((4)\) for every scale invariant term. Likewise, the algorithm for finding the Kovalevskaya exponents is also simple use of \((7)\). It creates a list in which to every scale invariant term the lists that contain the Kovalevskaya matrices and their eigenvalues is appended.

Now let us focus on the algorithm that for a given scale-invariant term produces all higher terms with respect to its weight. The outline of the algorithm is as follows

(i) For each term in every component of a vector field exclude terms from the scale-invariant term and calculate their weights \( q^{(i)} \) from \((6)\).

(ii) Add the terms with the same weight for every component.

It is simpler comparing to the previous algorithm as now the weights can be calculated independently for each term in every component.

This section should give the reader general idea of implementation of the algorithms in Mathematica. For more detailed description we refer the reader to the code \([5]\). The notebook contains a section that provides many examples of analysis of dynamical systems from \([3]\). It also serves as a benchmark. The next section gives general discussion on possible use of the package.

4. Conclusions

The package presented in the paper can be used as a base to implement more complex packages that automatize the Painlevé test or those that check integrability of dynamical systems. It can be also used to check if a dynamical system has scale invariant solutions. The future development of the package will go in these directions. As it was mentioned in Introduction there is a lot of the Painlevé test packages, however to our knowledge none of them is using the weight-homogeneous decomposition. Large parts of the code are written using functional style of programming, which is very effective approach in symbolic computations.

Acknowledgment

RK is supported by the Warsaw Center of Mathematics and Computer Science from the funds of the Polish Leading National Research Centre (KNOW). RK is grateful to Galina Filipuk(MIM UW, PL) and Rod Halburd(UCL, UK) for enlightening discussions.

References

[1] D. Baldwin and W. Hereman 2006 Symbolic software for the Painlevé test of nonlinear differential ordinary and partial equations, Journal of Nonlinear Mathematical Physics, 13 (1), 90–110
[2] Conte, R. (editor) 1999, The Painlevé property. One century later CRM Series in Mathematical Physics, Springer
[3] Goriely, A. 2001 Integrability and nonintegrability of dynamical systems Advanced Series on Nonlinear Dynamics, World Scientific
[4] Goriely, A. 1992 Investigation of Painlevé property under time singularities transformations J. Math. Phys. 33, 2728;
[5] Kycia, R. A. Web page: http://www.mimuw.edu.pl/~erkycia/
[6] Wolfram Mathematica Web page: http://www.wolfram.com/mathematica/
[7] Gui-qiong Xu, Zhi-bin Li 2004 Symbolic computation of the Painlevé test for nonlinear partial differential equations using Maple Comput. Phys. Commun. 161 65–75