Journal Title
XX(X):1–13

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Extending the Functionality of a Symbolic Computational Dynamic Solver by Using a Novel Term-tracking Method

Niloufar Motazedi [1], Matthew P Cartmell[2] and Jem Rongong [1]

Abstract

Symbolic Computational Dynamic (SCD) solvers are currently under development in order to provide new and powerful tools for modelling nonlinear dynamical systems. Such solvers consist of two parts; the core solver, which comprises an approximate analytical method based on perturbation, averaging, or harmonic balance, and a specialised term-tracker. A term-tracking approach has been introduced to provide a powerful new feature into computational approximate analytical solutions by highlighting the many mathematical connections that exist, but which are invariably lost through processing, between the physical model of the system, the solution procedure itself, and the final result which is usually expressed in equation form. This is achieved by a highly robust process of term-tracking, recording, and identification of all the symbolic mathematical information within the problem. In this paper the novel Source and Evolution Encoding Method is introduced for the first time and an implementation in *Mathematica* is described through the development of a specialised algorithm.

Keywords

Symbolic Computational Dynamics, Term-tracking methods, SEEM, Multiple Scales method, Approximate analytical methods, Nonlinear dynamics, *Mathematica*

Introduction

There are various advanced techniques for designing and predicting the responses of dynamical systems. Linear models and solutions are well established and frequently used for systems operating over a limited performance range, but in reality all mechanisms and systems are nonlinear. It is therefore important to develop an in-depth understanding of the effect of local and global nonlinearities on the systems response. Numerical methods constitute one of the most powerful approaches for estimating the solution of nonlinear systems and they can create tractable solutions for full scale problems. However, such solutions are invariably dependent on the initial boundary conditions, and the presence of nonlinearity in a system tends to make any interpolation and extrapolation of results invalid. It is also important to appreciate that linear analytical modelling techniques and numerical solution procedures both lend themselves to application to large systems, for which the number of degrees of freedom can be in the hundreds or thousands. This is not the case with approximate analytical methods which can be applied to nonlinear models, and in these cases the general problem scale is very small because of the limitations due to rapidly increasing algebraic complexity, and are usually of the order of a few degrees of freedom at most. This has led to the term *reduced order modelling* within the nonlinear dynamics community, and for the time being this general restriction applies to the approaches to be described in this paper. It should be noted that despite this limitation in scale reduced order models, if applied with care and ingenuity, can frequently offer excellent phenomenological representations of the dynamics of a wide range of nonlinear systems.

A natural path for development here is to consider the use of symbolic computation for the formulation of a computational algorithmic methodology for solution, based on some accepted mathematical technique such as perturbation, averaging, or harmonic balancing. To this end a Symbolic Computational Dynamic (SCD) solver concept was introduced, and an early version, with term-tracking, was proposed by Cartmell *et al.* [1]. SCD solvers can generally be considered to be constructed from two parts; a core solver and a term-tracker. The solver part is based on a computational form of an approximate analytical solution method and the term-tracker adds an extra dimension to the solver by generating a range of internal mathematical information during and after the solution process. This internal mathematical information comprises links and connections between quantities, functions, parameters, and coordinates and also between all the stages which are needed to generate the emerging solution. Normally this information is lost through the conventional processes of mathematics and also through the many algebraic and numerical simplifications that take place, and so it is

[1]Mechanical Engineering Department, University of Sheffield, Sheffield, S1 3JD, UK. [2]Aerospace Centre of Excellence, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, G1 1XJ, UK

Corresponding author:
Niloufar Motazedi, Mechanical Engineering Department,University of Sheffield, Mappin Building, Sheffield, S1 3JD, UK
Email: n.motazedi@sheffield.ac.uk

never seen. In the case of the SCD approach everything is tracked, identified through encoding, and recorded, for final visualisation, as an aid to the normal functionality obtained from a time or frequency domain response generated numerically from an approximate analytical solution.

The first of this generation of symbolic solvers was programmed using the *Mathematica* interface. The perturbation method of Multiple Scales [2, 3] was computerised [4, 5, 6], and applied in several case studies [7, 8]. However, this solver structure was not compatible with more recent ideas for term-tracking and so Forehand *et al.* [9] modified the solver of Khanin *et al.* [4] and implemented an early term-tracker based on what the authors then termed Source Encoding Method (SEM), again using the *Mathematica* programming interface. This modification changed the structure of the solver in such a way that users now had access to all stages of the analysis, as they would if using a pen and paper.

Investigation of the results generated with the SEM method have clarified the potential of the term-tracking methods to be used for extracting more information within the solution procedure. The SEM is designed to be used to create a tree-shaped history for each term within the solution procedure. However, it is beneficial to extract the history of each individual quantity within the terms as well. Furthermore, the ability to extract the perturbation order or codify the changing significance of fundamental quantities as the analysis proceeds can also be of great interest to the user.

The current study introduces the far more powerful Source and Evolution Encoding Method (SEEM) term-tracking method, and discusses its implementation within a modified multiple scales solver by the development of an adaptable and powerful encoding algorithm which has been designed for accurate and automated term-tracking within the procedures of symbolic computational dynamics solvers.

The Source and Evolution Encoding Method

Nonlinearity is an inherent part of all physical systems.There are several numerical and analytical methods capable of modelling and solving nonlinear systems. The equation of motion of a nonlinear system can sometimes be solved analytically, but frequently this degenerates into a difficult mathematical problem, and very often one which is based on assumptions which disregard aspects of the real physical system. As a case in point most vibrational problems, for example, are fundamentally very complicated, and finding the exact analytical solution is sometimes impossible. Therefore, some approximations must be made in the system modelling and also within the actual solution procedure [10]. As figure (1) shows, the SCD process creates valid connections between the *physical model*, the *solution method*, and the *final result*. This can provide an essential understanding of the problem and, importantly, highlight how these approximations can affect the results, by using appropriate *term-tracking* methods such as those described in this paper.

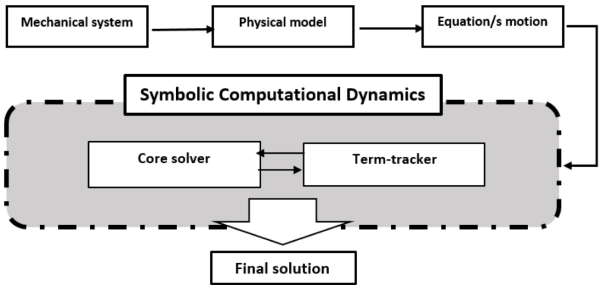

Figure 1. An overview of Symbolic Computational Dynamic solvers

A parametrically excited pendulum

In this paper the SEEM encoding method is demonstrated by using the same case study as explored in [9] for the simpler term-tracking method.
Figure (2) shows a pendulum with a harmonic oscillating support, the finite rotation of the pendulum is shown by $\theta(t)$. The external excitation is shown by $W(t)$, which is equal to $qlcos(\Omega t)$, and the damping torque ϕ is set to be equal to $cl^2\dot{\theta}$.

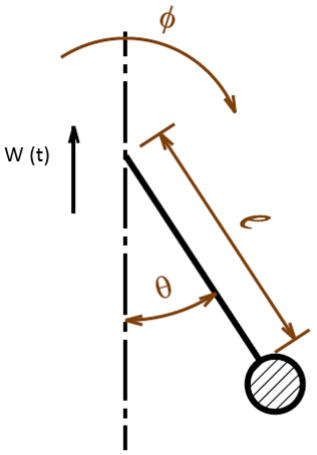

Figure 2. Parametric excitation of a pendulum based on [11].

Equation (1) shows the SEEM method applied to the appropriate equation of motion for parametric excitation of a pendulum [11]. The information generated by the SEEM is denoted by an under-bracket below each term, unlike in the SEM notation the position of the encoding vectors (either above or below the line) is optional and does not have any particular meaning.
The **first level** of the SEEM is unique for all the quantities, the equation number and the order of the small parameter (epsilon). The first digit for all encoding vectors is set to 1, as all the quantities are introduced in equation (1). The second digit for all the quantities is set to 1 (i.e. the epsilon order), except for ω which is placed inside the argument of the cosine function. It should be noted that the SEEM avoids the application of encoding for the time and the dependent variables before the introduction of the perturbation expansion. As a result (θ) and t have not been encoded.

$$\theta'' + \underbrace{2\epsilon\beta}_{(1,1)}\,\theta' - \underbrace{q\epsilon\omega^2}_{(1,1)}\,\theta\cos(\underbrace{\omega}_{(1,0)}\,t) + \underbrace{\epsilon\gamma}_{(1,1)}\,\theta^3 + \theta = 0 \quad (1)$$

In this case study the Method of Multiple Scales [3]is used because of its adaptability and transparent structure. In this method the approximate solution is a function of multiple independent time-scales, and the number of independent time-scales is limited by the order at which the expression is truncated. All variables or coordinates must be expressed by a uniformly valid equations to create a set of perturbation expansion. Consequently, the terminology *HOT* is used as an abbreviation for higher order terms. The final solution is built up gradually by utilising the solution to each level of perturbation (i.e. θ_0, θ_1,... and θ_i) and then combining them appropriately.

$$\theta(\epsilon,t)=\sum_{i=0}^{n-1}\epsilon^i\theta_i\left(\tau_0,\tau_1,...,\tau_n\right)$$

where

$$\tau_i=\epsilon^i t, i=1,2,...$$

In this case study the perturbation equations up to and including the first order correction are used. Equation (2) shows the SEEM applied to the perturbed solution form.

The **second level** encoding information for θ_0 is given as $(2,0)$, which highlights the origin and the order of the quantity, respectively. Whilst due to the presence of ϵ, the second term is encoded as $(2,1)$.

$$\theta=\underbrace{\theta_0}_{(2,0)}+\underbrace{\epsilon\theta_1}_{(2,1)}+HOT \quad (2)$$

Equation (3) and (4) are the encoded versions of the perturbed time derivatives, noting that these are constructed from the partial derivatives of the independent time-scales ($D_j=\dfrac{\partial}{\partial\tau_i}$).

$$\frac{d}{dt}=\underbrace{D_0}_{(3,0)}+\underbrace{\epsilon D_1}_{(3,1)}+HOT \quad (3)$$

$$\frac{d^2}{dt^2}=\underbrace{D_0^2}_{(9,0)}+\underbrace{2\epsilon D_0D_1}_{(4,1)}+HOT \quad (4)$$

When considering the equation of motion, it can be seen that the cubic form of the dependent variable (θ^3) can be written in an explicit form by using the perturbation expansion. The second level of the SEEM is applied to equation (5), the explicit equation number is added as the third encoding element to all the encoding vectors. Clearly the origin of this variable is from equation (2), while the order of the term is zero and it has became explicit in equation (5).

$$\theta^3=\underbrace{\theta_0^3}_{(2,0,5)}+HOT \quad (5)$$

Then equations (2), (3), (4), and (5) are substituted into equation (1), resulting in equation (6). In the early stages of the analysis the encoding information for all the quantities within an equation is quite similar. However, the SEEM information becomes more complex as the analysis progresses to the final stage.

$$\begin{aligned}&\underbrace{\theta_0}_{(2,0)}-\underbrace{q\epsilon\omega^2}_{(1,1)}\theta\cos(\underbrace{\omega\tau_0}_{(1,0)})+\underbrace{\epsilon\gamma}_{(1,1)}\underbrace{\theta_0^3}_{(2,0,5)}+\underbrace{\epsilon\theta_1}_{(2,1)}+\\&\underbrace{2\epsilon\beta}_{(1,1)}\underbrace{D_0}_{(3,0)}\underbrace{\theta_0}_{(2,0)}+\underbrace{2\epsilon D_0D_1}_{(4,1)}\underbrace{\theta_0}_{(2,0)}+\underbrace{D_0^2}_{(9,0)}\underbrace{\theta_0}_{(2,0)}\\&+\underbrace{\epsilon}_{(2,1)}\underbrace{D_0^2}_{(9,0)}\underbrace{\theta_1}_{(2,1)}=0\end{aligned} \quad (6)$$

The next step is to construct the zeroth and first order perturbation equations from equation (6). The coefficients of the zeroth (ϵ^0) and first (ϵ^1) order are separated and set to zero, leading to Equations (7) and (8), respectively.

$$\underbrace{\theta_0}_{(2,0)}+\underbrace{D_0^2}_{(9,0)}\underbrace{\theta_0}_{(2,0)}=0 \quad (7)$$

$$\begin{aligned}&\underbrace{\theta_1}_{(2,1)}+\underbrace{D_0^2}_{(9,0)}\underbrace{\theta_1}_{(2,1)}=\\&-\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{q\omega^2}_{(1,1)}\theta\cos(\underbrace{\omega\tau_0}_{(1,0)})+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{\gamma}_{(1,1)}\underbrace{\theta_0^3}_{(2,0,5)}\\&+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{2\beta}_{(1,1)}\underbrace{D_0}_{(3,0)}\underbrace{\theta_0}_{(2,0)}+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(4,1)}}\underbrace{2D_0D_1}_{(4,1)}\underbrace{\theta_0}_{(2,0)}\end{aligned} \quad (8)$$

Looking at Equation (8) one can see that normally in a standard multiple scales method, the ϵs from both sides would cancel out. However, in the SEEM the cancellation of parameters within the solution procedure is avoided; unless both the quantity and two first encoding digits are similar. $\epsilon\rightarrow(1,1)$ and $\epsilon\rightarrow(2,1)$ are both small parameters but not necessary numerically equal, as $\epsilon\rightarrow(1,1)$ is introduced at the stage where the choice is made to specify whether the system is lightly damped or softly nonlinear, while correspondingly the $\epsilon\rightarrow(2,1)$ is introduced by the choice of the perturbation expansion and adding the correction term.This helps the analyst to identify the stages of analysis that are affected by the initial assumptions during the modelling.

The solution of the zeroth order perturbation equation (7) has been given in equation (9). As θ_0 has appeared explicitly, the third level of the SEEM has been applied to the quantities with the first level of the SEEM, and new the first level of encoding is introduced for the un-encoded quantities. In this equation A and $\bar{A}$ are unknown complex functions of the slow time scaled τ_1 coefficient and its complex conjugate.

$$\theta_0=\underbrace{A}_{(2,0,9)}\underbrace{exp}_{(9,0)}(\overbrace{i}^{(9,0)}\tau_0)+\underbrace{\bar{A}}_{(2,0,9)}\underbrace{exp}_{(9,0)}(\overbrace{-i}^{(9,0)}\tau_0) \quad (9)$$

Then the solution of the zeroth-order perturbation equation is substituted into the cubic form of the dependent variable θ_0^3, resulting in equation (10). The encoding of A and $\bar{A}$ are defined as (2,0,9,10), which clearly shows the origin of this zeroth order term is equation (2), and that it has appeared in an explicit equation structure in Equation (9). Furthermore,

its structure has been modified in Equation (10).

$$\begin{aligned}\underbrace{\theta_0^3}_{(1,0,5)} &= \underbrace{A^3}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \overbrace{3}^{(1,0,10)}\ \tau_0)}_{(9,0)} \\ &+ \underbrace{3}_{(10,0)}\ \underbrace{A^2}_{(2,0,9,10)}\ \underbrace{\bar{A}}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{i}^{(10,0)}\ \tau_0)}_{(9,0)} \\ &+ \underbrace{3}_{(10,0)}\ \underbrace{\bar{A}^2}_{(2,0,9,10)}\ \underbrace{A}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{-i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &+ \underbrace{\bar{A}^3}_{(2,0,9,10)}\ \underbrace{-i}_{(9,0,10)}\ \underbrace{exp\ (\overbrace{i}^{(9,0)}\ \overbrace{3}^{(1,0,10)}\ \tau_0)}_{(9,0,10)}\end{aligned} \tag{10}$$

Equation (10) is an example of the complex logic behind the SEEM. For example, it can be noted that there are two types of number 3, with the same numerical value but dissimilar encoding sources. The first one is $3 \rightarrow (1, 0, 10)$ within the index of the exponential function, and this is referred to the first natural frequency of the system. The second number 3 is $(3 \rightarrow (10, 0))$, which is defined because of the algebraic procedure in this equation.

Then the exponential form of the harmonic function within the external excitation term in equation (1) is given below:

$$\cos \underbrace{\omega}_{(1,0)} \tau_0 = \underbrace{\frac{1}{2} exp(-i\omega\tau_0)}_{(1,0,11)} + \underbrace{\frac{1}{2} exp(i\omega\tau_0)}_{(1,0,11)} \tag{11}$$

The solution of the zeroth order perturbation equation (9) and the cubic form of the dependent variable (10) are substituted into the first order perturbation equation (8), and the result is shown in equation (18).

$$\begin{aligned}&\underbrace{D_0{}^2}_{(9,0)}\ \underbrace{\theta_1}_{(2,1)} + \underbrace{\theta_1}_{(2,1)} = \\ &\underbrace{-2}_{(4,1)}\ \underbrace{\frac{\overbrace{\epsilon}^{(4,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{D_1}_{(4,1)}\ \underbrace{A}_{(2,0,9)}\ \underbrace{i}_{(9,0)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \tau_0)}_{(9,0)} - \\ &\underbrace{2}_{(4,1)}\ \underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{D_1}_{(4,1)}\ \underbrace{\bar{A}}_{(2,0,9)}\ \underbrace{-i}_{(9,0)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &\underbrace{-2}_{(4,1)}\ \underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\beta}_{(1,1)}\ \underbrace{A}_{(2,0,9)}\ \underbrace{i}_{(9,0)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &\underbrace{-2}_{(4,1)}\ \underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\beta}_{(1,1)}\ \underbrace{A}_{(2,0,9)}\ \underbrace{i}_{(9,0)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &-\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\gamma}_{(1,1)}\ \underbrace{A^3}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \overbrace{3}^{(10,0)}\ \tau_0)}_{(9,0)} \\ &-\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\gamma}_{(1,1)}\ \underbrace{3}_{(10,0)}\ \underbrace{A^2\bar{A}}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &-\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\gamma}_{(1,1)}\ \underbrace{3}_{(10,0)}\ \underbrace{\bar{A}^2A}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{-i}^{(9,0)}\ \tau_0)}_{(9,0)} \\ &-\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{\gamma}_{(1,1)}\ \underbrace{\bar{A}^3}_{(2,0,9,10)}\ \underbrace{exp(\overbrace{-i}^{(9,0)}\ \overbrace{3}^{(10,0)}\ \tau_0)}_{(9,0)} \\ &+\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{q\omega^2}_{(1,1)}\ \underbrace{\frac{\overbrace{1}^{(1,0,11)}}{\underbrace{2}_{(1,0,11)}}}_{(1,0,11)}\ \underbrace{A}_{(2,0,9)}\ \underbrace{exp(\underbrace{i}_{(*,12)}\ \tau_0(\underbrace{\omega}_{(1,0,11)} + \underbrace{1}_{(9,0)}))}_{(*,12)} \\ &+\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{q\omega^2}_{(1,1)}\ \underbrace{\frac{\overbrace{1}^{(1,0,11)}}{\underbrace{2}_{(1,0,11)}}}_{(1,0,11)}\ \underbrace{\bar{A}}_{(2,0,9)}\ \underbrace{exp(\underbrace{-i}_{(*,12)}\ \tau_0(\underbrace{\omega}_{(1,0,11)} + \underbrace{1}_{(9,0)}))}_{(*,12)} \\ &+\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{q\omega^2}_{(1,1)}\ \underbrace{\frac{\overbrace{1}^{(1,0,11)}}{\underbrace{2}_{(1,0,11)}}}_{(1,0,11)}\ \underbrace{A}_{(2,0,9)}\ \underbrace{exp(\underbrace{-i}_{(*,12)}\ \tau_0(\underbrace{\omega}_{(1,0,11)} + \underbrace{-1}_{(9,0)}))}_{(*,12)} \\ &+\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}}_{(4,1)}\ \underbrace{q\omega^2}_{(1,1)}\ \underbrace{\frac{\overbrace{1}^{(1,0,11)}}{\underbrace{2}_{(1,0,11)}}}_{(1,0,11)}\ \underbrace{\bar{A}}_{(2,0,9)}\ \underbrace{exp(\underbrace{i}_{(*,12)}\ \tau_0(\underbrace{\omega}_{(1,0,11)} + \underbrace{-1}_{(9,0)}))}_{(*,12)}\end{aligned} \tag{12}$$

The information added by implementation of the SEEM has created a large amount of information for each term. It is possible to track each quantity to its origin and identify the stages at which the term has been modified.

The SEEM signifies the distinctions between the $2 \rightarrow (1,1)$, $2 \rightarrow (4,1)$, and $2 \rightarrow (1,0,11)$ which relate to the damping term, the second derivative definition after introducing the perturbation expansion, and the exponential form of the external excitation term, respectively. Moreover, the application of the **compound encoding level** of the SEEM is used in this equation for the first time. For example, $\underbrace{exp}_{(*,12)}(\underbrace{i}_{(*,12)}\tau_0(\underbrace{\omega}_{(1,0,11)}+\underbrace{1}_{(9,0)}))$ means $exp(i\tau_0) \rightarrow (9,0)$ is subsumed into $exp(-i\omega\tau_0) \rightarrow (1,0,11)$ at equation (12) for the first time. As a result, applying the compound level of the SEEM has helped to simplify the structure of the equation, without losing any fundamental encoding information.

The perturbation expansion in the multiple scales method must be uniformly valid, therefore, θ_1 must always be smaller than θ_0 . Terms that are causing a particular perturbation solution contribution to grow too large too quickly are defined as *secular*. Terms containing the natural frequency of the homogeneous system are resonant and their sum must be zero in order to guarantee a valid perturbation expansion. In this case, the natural frequency of the homogeneous system is equal to one, therefore, terms containing $exp(i\tau_0)$ are taken out and set to zero. There are some terms in equation (12) that can be resonant depending on the value of ω, the external excitation frequency. If ω is set to the numerical value of 2, then $\underbrace{exp}_{(*,12)}(\underbrace{i}_{(*,12)}\tau_0(\underbrace{\omega}_{(1,0,11)}+\underbrace{-1}_{(9,0)}))$ and its complex conjugate are considered to be secular. As a result the principal parametric resonance condition is given by equation (13).

$$\omega = \underbrace{2}_{(13,0)} \quad (13)$$

To study the near resonance condition, a small detuning parameter $(\epsilon\sigma)$ around the principal parametric resonance is defined, equation (14).

$$\omega = \underbrace{2}_{(13,0)} + \underbrace{\epsilon}_{(14,1)}\underbrace{\sigma}_{(14,1)} \quad (14)$$

To find the near-resonance solution, equation (14) must be substituted into the first order perturbation equation (12). In a standard multiple scale analysis, all the *epsilons* are equal and therefore, whenever it is required, the *epsilons* can be replaced with the definition of the perturbation parameter, equation (15).

$$\underbrace{\epsilon}_{(2,1)} = \frac{\tau_1}{\tau_0} \quad (15)$$

However, according to the SEEM concept, the *epsilons* in an analysis are not necessarily equal and they must not cancel out. Below (a) is a selected term from equation (12), and then applying the near-resonance condition leads to (b). For having a valid multiple scales solution procedure, $\epsilon \rightarrow (14,1)$ must be equal to $\frac{\tau_1}{\tau_0}$, resulting in expression (c). This provides enough evidence that the numerical value of the small parameter in the detuning parameter, $\epsilon \rightarrow (14,1)$, must be equal to the perturbation expansion, $\epsilon \rightarrow (2,1)$.

(a) :

$$\underbrace{exp}_{(*,12)}(\underbrace{i}_{(*,12)}\tau_0(\underbrace{\omega}_{(1,0,11)}+\underbrace{-1}_{(9,0)}))$$

(b) :

$$\underbrace{exp}_{(*,12)}(\underbrace{i}_{(*,12)}\tau_0(\underbrace{2}_{(13,0)}+\underbrace{\epsilon\sigma}_{(14,1)}+\underbrace{-1}_{(9,0)}))$$

(c) :

$$\underbrace{exp}_{(*,12)}(\underbrace{i}_{(*,12)}(\underbrace{2}_{(13,0)}\tau_0+\underbrace{\sigma}_{(14,1)}\tau_1+\underbrace{-1}_{(9,0)}\tau_0))$$

Continuing the solution procedure (as shown in the appendix) would lead to determination of the response for the non-resonant case within this problem. In the standard multiple scales analysis [11], without applying the SEEM, the symbolic statements for the amplitude (a) and phase (α) are given as $a = a_0 exp(-\beta\tau_1)$ and $\alpha = \alpha_0 - \frac{3}{8}{a_0}^2\frac{\gamma}{2\beta}exp(-2\beta\tau_1)$, respectively. Equations (27) and (28), in the appendix, show the full forms of these equations by avoiding any cancellation and then applying the SEEM.

$$a = \underbrace{a_0}_{(27,0)}\underbrace{exp}_{(27,0)}(\{\frac{\overbrace{2}^{(4,1,27)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,27)}}{\underbrace{\epsilon}_{(2,1,27)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{\beta}_{(1,1,27)}T_1)$$

$$\alpha = \underbrace{\alpha_0}_{(28,0)} - \frac{\overbrace{3}^{(10,0,28)}}{\underbrace{2}_{(2,0,9,10)}}(\overbrace{a_0}^{(27,0,28)})^{\overbrace{2}^{(2,0,9,10)}} \times \frac{\underbrace{2}_{(10,0,20)} \times (\underbrace{2}_{(10,0,19)})^{\overbrace{2}^{(10,0,19)}}}{\{\frac{\overbrace{\epsilon}^{(1,1,28)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp}^{(10,0,19)}(\overbrace{i}^{(10,0,19)}\overbrace{2}^{(2,0,9,10)})}{\underbrace{exp(i\alpha)}_{(10,0,20)}\underbrace{exp(i\alpha)}_{(26,0,28)}}\}\underbrace{\gamma}_{(1,1,28)}} \times$$

$$\{\frac{\overbrace{2}^{(4,1,20)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,28)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{2}_{(2,0,9,10)}\underbrace{\beta}_{(1,1,28)}$$

$$\underbrace{exp}_{(27,0,28)}(\{\frac{\overbrace{2}^{(4,1,27)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,27)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{-2}_{(2,0,9,10)}\underbrace{\beta}_{(1,1,28)}T_1)$$

Logistics of the Source and Evolution Encoding Method

As shown in the case study the purpose of the Source and Evolution Encoding Method (SEEM) is to highlight the contribution of each quantity to the analysis by applying a relevant encoding. The generated encodings are stored in their encoding multipliers, and these are of variable length, depending on the encoding level. The encoding multiplier mainly contains from two to four elements, with

four as the norm *(i,j,k,l)*. The first element (i) indicates the origin of the quantity (i.e. the equation number in which the quantity is initially introduced), while j represents the order of the corresponding term containing the quantity in the original differential equation of motion, k shows the equation number where the quantity appears in an explicit form for the first time, and l is the equation number where this explicit quantity has first been significantly modified. This information assists in identifying the important stages of analysis for each quantity, at all points the analysis. It is very important to be able to identify the origin of the variable (i) and its relative numerical significance (j). The explicit equations that emerge within the early stages of the approximate analytical solution method play a major role in the final solution, and so the encoding logic upon which the SEEM is based is summarised in Table (1).

Table 1. The SEEM term-tracking guidelines

The SEEM encoding logic		
Encoding level	**Description**	**Visualisation**
Fist level	Origin of the variable and its order	(i,j)
Second level	Equation number in which the variable has appeared in an explicit equation	(i,j,k)
Third level	Equation number in which the explicit form of the variable has been modified	(i,j,k,l)
Compound level	$(i,j,k,l)_n$ subsumed with $(i,j,k,l)_m$	$(*,k)$

The main policy of the SEEM is to avoid any possible cancellation in order to ensure that there is no information loss. Having said this it is obvious that keeping all the quantities unmodified in their original form would make the equation forms unnecessarily complicated. Therefore a compound level of encoding is suggested, and is found mainly to focus on the exponential functions (***e*** being the base of the natural logarithm and ***i*** the imaginary number).

The following arbitrary expression demonstrates the application of the compound level encoding. It is assumed that two exponential quantities, with different origins, are to be subsumed together in an arbitrary *equation (x)*. Considering the right hand side (RHS) of this expression, one can see that this term is created by subsuming two exponential terms created from arbitrary *equation (n)* and *(m)* of the preceding analysis. If the exponential term is subsequently subsumed for a second, third and fourth time then the $\#$, $\$$ and $\pounds$ symbols are used, respectively to denote that this further compounding has happened.

$$\underbrace{exp}_{(n,0,21)}\Big(\overbrace{i}^{(n,0,21)}\ \tau_0\Big)\times\underbrace{exp}_{(m,1,10)}\Big(\overbrace{i\omega}^{(m,1,10)}\ \tau_0\Big)$$
$$\rightarrow$$
$$\underbrace{exp}_{(*,x)}\Big(\overbrace{i}^{(*,x)}\Big(\overbrace{1}^{(n,0,21)}+\overbrace{\omega}^{(m,1,10)}\ \tau_0\Big)\Big)$$

Computerisation of the SEEM

Considering the relatively complex logic behind the SEEM approach a true generalisation and therefore a guaranteed and completely accurate implementation of the method is not a straightforward task. This method has taken the encoding level further than was originally defined in the SEM method of [9] and so it can detect and track the history of sub-terms as well. Therefore the SEEM must be capable of indicating and recognising the precise type of functions, within each term, to be encoded or updated accordingly.

Figure 3 provides an overview of the process in which an input equation is converted to the encoded form in 10 steps. The input of the SEEM algorithm is an equation which can be either unencoded or encoded. The input equation is decomposed into two expressions comprising the left-hand side and right hand side. Then, depending on the type of equation, the encoding information is defined for each quantity for equations that are as yet unencoded, or updated. Then the encoded quantities are combined in term format and the encoded equation will be rebuilt from the encoded quantities.

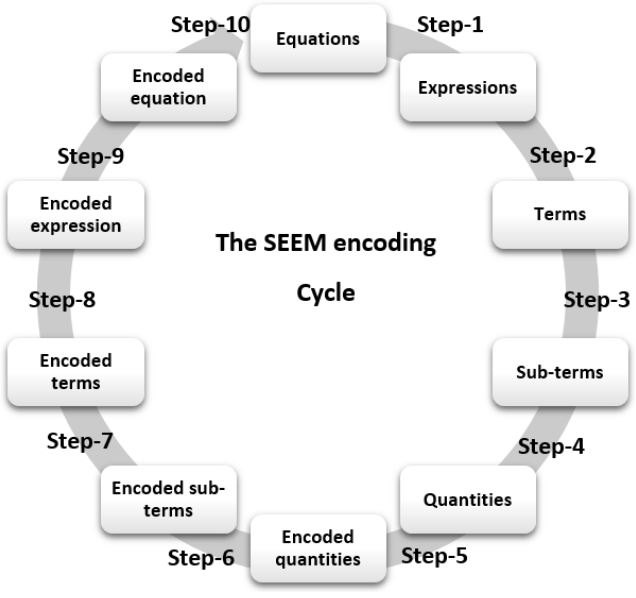

Figure 3. A graphical representation of the SEEM encoding process

The policy of the SEEM is to gather information about the fundamental quantities in each equation, therefore the algorithm must be able to decompose each equation into expressions, terms, sub-terms, and finally quantities. A

schematic view of this action for the equation of motion of a parametrically excited pendulum is shown in Figure (5). It is possible to notice that this process completely depends on the function types in each term, so the algorithm should have enough information to distinguish between different function types and decompose sub-terms accordingly. For example, the trigonometric function of $cos(\omega\tau_0)$ is decomposed down to ω ; however, the power function of ω^2 is decomposed into ω and 2.

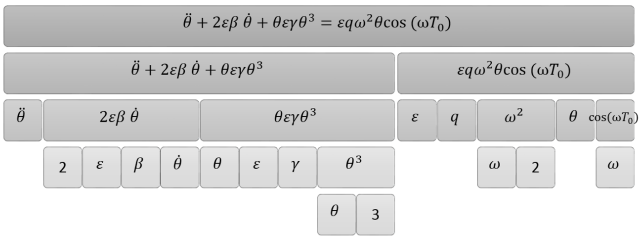

Figure 4. The decomposition of an equation into fundamental quantities

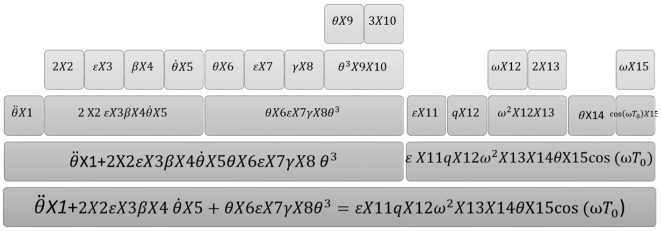

Figure 5. The process of generating an encoded equation from the encoded quantities

The SEEM is implemented within a code written in *Mathematica* by defining the SEEM algorithm. The output generated by the code for this algorithm can be processed by any subsequently developed functions. Currently a displayer function has been developed to show encoding outputs using the built-in *Mathematica* Tooltip function. The encoding information for each quantity becomes visible when the mouse is hovered around it. To make it possible for the data to be used in later stages of analysis the encoding data must be stored in a special format inside each equation, and, importantly, this has to be without creating any interruptions to the ongoing solution procedure.

In the *Mathematica* language everything is an expression, so Head is used to show the type of expression. Therefore a new class of Head in *Mathematica* named *envec* (where this is an abbreviation for encoding vector) is introduced for this purpose. As all quantities inside the term are encoded separately, an *identity factor* is defined as the string form of the quantity and is stored as the first element inside the encoding multiplier. Therefore it is possible for the algorithm to identify the specific encoding multiplier that describes each quantity in the later stages. Consequently, the encoding vector is defined with a maximum of five elements, including the four encoding digits of the SEEM encoding and identity factor. For example the third term of equation (1) has to be encoded as $q\epsilon\omega^2\cos(\omega t) \times envec[\epsilon, 1, 1, nan, nan] \times envec[q, 1, 1, nan, nan] \times envec[\omega^2, 1, 1, nan, nan] \times envec[\omega, 1, 0, nan, nan]$. In the case that one element of the encoding vector is not available, 'nan' (Not-A-Number) will be used instead.

The main duty of the encoder function is to assign or identify a valid encoding multiplier for each quantity. The encoder function has three inputs, equation number (n), the corresponding side of equation (m) and the reference origin (p). In most cases equation number (n) and origin (p) are the same, except where a different form of a variable is described in a new equation. For example the polar form of $\cos(\omega\tau_0)$ in equation (1) is given in equation (11), therefore the equation number (n) is 11, while the origin (p) is 1.

$$\cos\underbrace{\omega}_{(1,0)}\tau_0 = \underbrace{\frac{1}{2}exp(-i\omega\tau_0)}_{(1,0,11)} + \underbrace{\frac{1}{2}exp(i\omega\tau_0)}_{(1,0,11)}$$

The SEEM algorithm is given in Algorithm (1) and it should be noted that all the input *expressions*, *terms* and *sub-terms* are distinguished from the updated output by using an accent ($\prime$). At first the function specifies the third digit (k) of the encoding (lines 2 to 6). Then it decomposes the *expression* into *terms*, and the second digit of encoding is extracted by specifying the ϵ order (j). Then the encoding multipliers (*envec*) for each term are defined and are stored in the name of *indextrm*. It is very important that the function identifies the variable type correctly, and applies the encoding accordingly. For example, for an exponential function the index and the base must be encoded separately. Also to avoid any cancellation through the analysis this function converts all the inputs to symbols. If a selected variable is matched with any *identity factor* inside the encoding multipliers (line 14), the encoding vector must be updated, otherwise a new encoding multiplier would need to be defined (line 25). During the updating process the third digit of the previously defined encoding vector (k) is updated (line 16). Furthermore if the explicit form of the variable is modified (line 18), by that means, the *identity factor* does not appear the same as the variable, and hence the fifth digit (l) is correspondingly updated. Then the encoding multiplier is multiplied by the original sub-term (line 27) to create the encoded terms, after which the encoded expression is formed by summing all the encoded terms.

The compound level of encoding is applied to the exponential form of function. This function has a logic in order to identify the correct encoding information for a variable, especially when there is more than one instance of the same variable in the term. Also, according to the SEEM logic, the dependent variable and its derivatives should not be encoded before applying the perturbational form.

Algorithm 1 The SEEM encoding algorithm

1: **function** SEEMENCODE(n, m, p)
2: **if** eqn[n] = explicit **then**
3: $k \leftarrow p$
4: **else**
5: $k \leftarrow nan$
6: **end if**
7: break the expression' to terms'
8: **for** each term' **do**
9: $indextrm \leftarrow$ envec'
10: $j \leftarrow \epsilon order$
11: break the $terms'$ to $subterm'$
12: **for** each $subterm'$ **do**
13: Check the variable type (exponential,trigonometric,integer, and etc.)
14: **if** subterm' $\in indextrm$ **then**
15: **if** 4th' = "nan" **then**
16: $envec \leftarrow$ ["$subterm$",*i'*,*j'*,*k*,*nan*]
17: **else**
18: **if** subterm' $\neq$ $subterm$ and *l'*=*nan* **then**
19: $envec \leftarrow$ ["$subterm$",*i'*,*j'*,*k'*,*p*]
20: **else**
21: $envec \leftarrow$ envec'["$subterm$",*i'*,*j'*,*k*,*l'*]
22: **end if**
23: **end if**
24: **else**
25: $envec \leftarrow$["subterm",*p*,*j*,*k*,*nan*]
26: **end if**
27: $subterm \leftarrow$subterm' $*$ *envec*
28: **end for**
29: $term \leftarrow$*subterms*
30: **end for**
31: $expression \leftarrow \sum$*terms*
32: **end function**

Algorithm 2 Displayer

1: **function** DISPLAYER(expression)
2: break the *expression* to *terms*
3: **for** each *term* **do**
4: indextrm $\leftarrow$ *envec*
5: break the $terms$ to $subterm$
6: **for** each $subterm$ **do**
7: Check the variable type (exponential,trigonometric,integer, and etc.)
8: Find the *envec* describing *subterm*
9: Remove *identity factor* and *"nan"* from *envec*
10: $subterm \leftarrow$ **Tooltip**[subterm,(i,j,k,l)]
11: **end for**
12: $term \leftarrow$*subterms*
13: $expression \leftarrow \sum$*terms*
14: **end for**
15: **end function**

Displaying term-tracking information

After the encoding vector is assigned to the equations, a function is required to process the encoding into an understandable form for the user. To make this possible a displayer function has been developed, see Algorithm (2). In order to ensure valid results equations should be always encoded before being displayed. The displayer function separates the encoding vectors term by term. Then it matches each variable with the identity factor inside the encoding (line 8) to find the correct encoding multiplier. All the unnecessary information, such as the *nan* and the *identity factor* are removed (line 9), and the rest of the encoding information is displayed by using a Tooltip function (line 10). Finally, the updated sub-term is replaced inside the equation.

Figure (6) is an example that shows how a sub-term such as $e^{(i\tau_0+i\omega\tau_0)}$ with the encoding vectors of $\{[e^i,1,0,9],[e^i,str,12],[e^i,1,0,11]\}$ is displayed by applying the display function. Also Figure (7) illustrates the way that an input of a derivative function and itself can be displayed. This can be compared to $D_0\theta_0$ in equation (6).

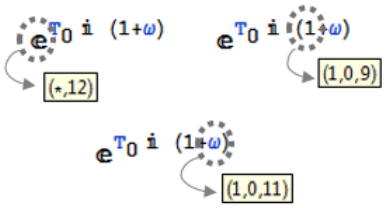

Figure 6. The displayer function output for an exponential sub-term

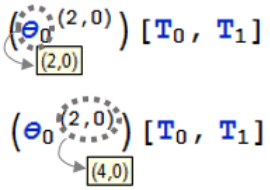

Figure 7. The displayer function output for a differentiated function

The problem of a parametrically excited pendulum is solved with the developed SCD solver. As Figure (8) shows, the equation of motion of this problem is encoded by this method. This equation can be compared with equation (1). Then the zeroth and first order perturbations equations are shown in Figures (9) and (10). These equations can be compared with equation (7) and (8).

$$\gamma_{ns_{(1,1)}}\ \epsilon_{ns_{(1,1)}}\ \theta[\tau]^3 + \theta\left[\tau\right] -$$

$$\omega^2_{(1,1)}\ \mathrm{Cos}\left[\tau\,\omega_{(1,0)}\right]\ q_{ex_{(1,1)}}\ \epsilon_{ex_{(1,1)}}\ \theta\left[\tau\right] +$$

$$2_{dp_{(1,1)}}\ \beta_{dp_{(1,1)}}\ \epsilon_{dp_{(1,1)}}\ \theta'[\tau] + \theta''[\tau] == 0$$

Figure 8. Screen-shot of the equation of motion of a parametrically excited pendulum

$$D_{0,(4,0)}^2\ \theta_{0_{(2,0)}}[T_0, T_1] + \theta_{0_{(2,0)}}\left[T_0, T_1\right] == 0$$

Figure 9. Screen-shot of the zeroth order perturbation equation for a parametrically excited pendulum

$$D_{0,(4,0)}^2\ \theta_{1_{(2,1)}}[T_0, T_1] + \theta_{1_{(2,1)}}\left[T_0, T_1\right] ==$$

$$-\frac{D_{0_{(3,0)}}\ 2_{dp_{(1,1)}}\ \beta_{dp_{(1,1)}}\ \epsilon_{dp_{(1,1)}}\ \theta_{0_{(2,0)}}[T_0, T_1]}{\epsilon_{p_{(2,1)}}} -$$

$$\frac{D_{0_{(4,1)}} D_{1_{(4,1)}}\ 2_{dr_{(4,1)}}\ \epsilon_{dr_{(4,1)}}\ \theta_{0_{(2,0)}}[T_0, T_1]}{\epsilon_{p_{(2,1)}}} -$$

$$\frac{\gamma_{ns_{(1,1)}}\ \epsilon_{ns_{(1,1)}}\ \theta_{0_{(2,0,5)}}[T_0, T_1]^3}{\epsilon_{p_{(2,1)}}} +$$

$$\frac{\omega^2_{(1,1)}\ \mathrm{Cos}\left[T_0\,\omega_{(1,0)}\right]\ q_{ex_{(1,1)}}\ \epsilon_{ex_{(1,1)}}\ \theta_{0_{(2,0)}}\left[T_0, T_1\right]}{\epsilon_{p_{(2,1)}}}$$

Figure 10. Screen-shot of the first order perturbation equation for a parametrically excited pendulum

The SEEM applied to autoparametric resonance in a coupled beam

As a further case study a two degree of freedom (DoF) problem involving autoparametric resonance in a coupled beam system [12] is now investigated by using the current version of the SCD solver. The system of a pair of coupled beams is shown in figure (11). A single frequency external excitation is applied transversally onto the primary beam (AB). Therefore, the transverse response of the main beam causes a predominantly axial excitation of the secondary beam (BC) at the coupling point, thereby introducing a parametric excitation of the secondary beam through stiffness modulation. The secondary beam is considered to have a high flexural stiffness ratio, therefore the bending deformation of BC in the ABC plane can be neglected. In this figure, *X* and *Y* represent the physical in-plane and out-of-plane responses, respectively.

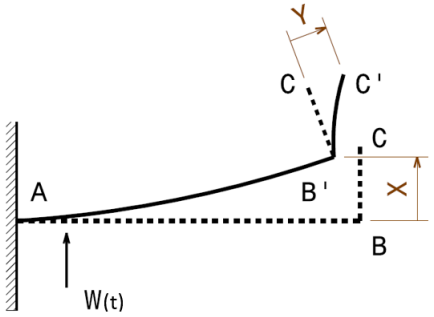

Figure 11. Coupled beam system based on [12]. *AB* is the primary beam, and *BC* is the secondary beam. *W(t)* represents the external excitation

The encoded equations of motion for this system are given in Figures (12) and (13). The encoding function is flexible, allowing any choice of symbols for the dependent variables, and in this case *X* and *Y* are used. The current version of the SCD solver is capable of processing problems of up to four degrees of freedom, so essentially any coupled nonlinear ODE problem of up to four simultaneous equations can be accommodated. There is no theoretical limitation here but practicality has so far enforced the use of relatively low dimensional systems in order to guarantee correct logical operation. The function structure is programmed in a way that makes it possible for the professional user to apply any modification into the solution process or the term-tacking part, if this is required.

$$-\ q_{(1,1)}\ \mathrm{Cos}\left[\tau\ \Omega_{(1,0)}\right]\ \epsilon_{ex_{(1,1)}} +$$

$$\omega_{1,(1,0)}^2\ X[\tau] - \mu_{(1,1)}\ \epsilon_{cplf_{(1,1)}}\ Y'[\tau]^2 +$$

$$2_{dmpf_{(1,1)}}\ \epsilon_{dmpf_{(1,1)}}\ \zeta_{1_{(1,1)}}\ \omega_{1_{(1,1)}}\ X'[\tau] +$$

$$X''[\tau] - \mu_{(1,1)}\ \epsilon_{cplf_{(1,1)}}\ Y[\tau]\ Y''[\tau] == 0$$

Figure 12. Screen-shot of the equation of motion for the in-plane system

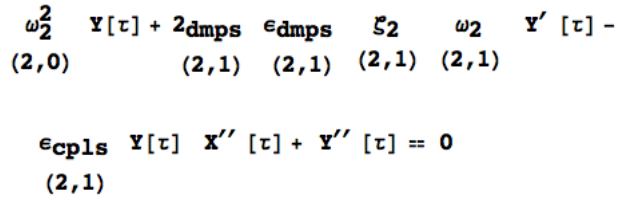

Figure 13. Screen-shot of the equation of motion for the out-of-plane system

The zeroth order perturbation equation for both the in and out of plane systems are shown in Figures (14) and (15), respectively.

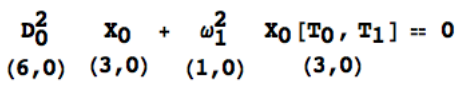

Figure 14. Screen-shot of the zeroth order perturbation equation the in-plane system

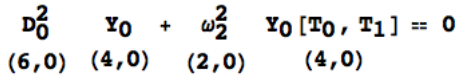

Figure 15. Screen-shot of the zeroth order perturbation equation the out-of-plane system

Discussion

After a general overview of the Symbolic Computational Dynamics solvers, a new version of the term-tracker, the Source and Evolution Encoding method, is introduced and fully computerised. The concept is fully explained using two cases studies; a parametrically excited pendulum and autoparametric resonance in a coupled beam system. The SEEM has introduced an extra dimension to the analysis, and it can directly connect each mathematical symbol and numerical value to the physical concept behind it. A large amount of information is created by the SEEM, and it can be useful in various ways. For the novice user it can simply provide an insight into the solution procedure by showing the reasoning behind each equation. As a result, the user is less likely to be intimidated by the long and complicated solution procedure as understanding any procedure is made much easier by having all the information available on the presence of a quantity in a particular form in the specific equation. Also, the more experienced user may find it deeply interesting to observe how physical quantities are expressed in a chosen mathematical form, and the way each physical quantity is modified through the solution procedure in order to construct a final solution according to a set of assumptions and approximations. Furthermore, it is potentially possible to compare the structure of different solution procedures and how they result in final solutions. For example, one could compare the SEEM data generated by applying the Harmonic Balance to that from the Multiple Scales method for a specific problem. There is also a strong argument to be made for the development of a comprehensive yet assimilable visual user interface for broad and deep interpretation of the SEEM data, and the inevitable fact that the utility of this information will be enriched by the capability of the visualisation system used to observe it. Work on visualisation continues apace and will be reported on at a later date.

Conclusions

In this paper a new class of symbolic computational dynamics (SCD) solver is introduced through the use of a novel encoding methodology. The Source and Evolution Encoding Method has been fully explained and an application to the problem of a parametrically excited pendulum is addressed in some detail. The implementation of this method within code using the *Mathematica* interface was generalised by the development and implementation of a novel algorithm. The computerisation of the method has been completed and has been demonstrated across two case studies, one of which involves a pair of coupled nonlinear ordinary differential equations, thereby paving the way for larger scale multi-degree-of-freedom problem modelling. The output generated by this type of SCD solver has the potential to be visualised in terms of breadth and depth, and therefore to provide a new level of insight into the problem and also its solution. This insight would otherwise be entirely lost to the user, and this is important because it has a significant potential for increasing user awareness of how all terms and quantities in the emerging solution interact, and also how initial assumptions within the solution process propagate, and ultimately manifest as properties of the solution itself.

References

[1] MP Cartmell and DIM Forehand. On the assumptions and decisions required for reduced order modelling of engineering dynamical systems. In *Proc. 6th Euromech Nonlinear Dynamics Conf.(ENOC-2008). IPACS Open Acces Library*, 2008.

[2] AH Nayfeh. *Introduction to perturbation techniques*. John Wiley & Sons, 2011.

[3] AH Nayfeh and DT Mook. *Nonlinear oscillations*. John Wiley & Sons, 1979.

[4] R Khanin, MP Cartmell, and A Gilbert. Applying the perturbation method of multiple scales. *Mathematica in Education and Research*, 8(2):19–26, 1999.

[5] R Khanin, MP Cartmell, and A Gilbert. A computerised implementation of the multiple scales perturbation method using mathematica. *Computers & Structures*, 76(5):565–575, 2000.

[6] R Khanin and MP Cartmell. Parallelization of perturbation analysis: Application to large-scale engineering problems. *Journal of Symbolic Computation*, 31(4): 461–473, 2001.

[7] J Warmiński, G Litak, MP Cartmell, R Khanin, and M Wiercigroch. Approximate analytical solutions for

primary chatter in the non-linear metal cutting model. *Journal of Sound and Vibration*, 259(4):917–933, 2003.

[8] David IM Forehand, R Khanin, and Matthew P Cartmell. A lagrangian multibody code for deriving the symbolic state-space equations of motion for open-loop systems containing flexible beams. *Mathematics and Computers in Simulation*, 67(1):85–98, 2004.

[9] DIM Forehand and MP Cartmell. The implementation of an automated method for solution term-tracking as a basis for symbolic computational dynamics. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 225(1):40–49, 2011.

[10] MP Cartmell. *Introduction to linear, parametric and nonlinear vibrations*. Chapman and Hall London, 1990.

[11] JJ Thomsen. Vibrations and stability, order and chaos. 1997.

[12] JW Roberts and MP Cartmell. Forced vibration of a beam system with autoparametric coupling effects. *Strain*, 20(3):123–131, 1984.

Notations

A	Unknown complex function of the slow time scale
a	The amplitude of the response of the pendulum case study
D	Partial derivative notation of the independent time scale ($D_j = \frac{\partial}{\partial \tau_i}$)
envec	Encoding vector
HOT	Higher Order Terms
l	Pendulum rod length
nan	Not A Number
q	The external excitation amplitude
SCD	Symbolic computational Dynamics solver
SEEM	Source and Evolution Encoding Method
t	The real time
θ_0	Fast time scale
θ_1	Slow time scale
$W(t)$	The external excitation
X	The physical in-plane response of the couple beam system
X_0	The zeroth order perturbation of the in-plane response
X_1	The first order perturbation term of the in-plane response
Y	out of plane response of the coupled beam system
Y_0	The zeroth order perturbation term for the out-of-plane response
Y_1	The first order perturbation term for the out-of-plane response
α	The phase response for the pendulum case study
γ	The nonlinear stiffness coefficient
γ_{ns}	The nonlinear stiffness coefficient
ϵ	A small parameter
ϵ_{cplf}	A small parameter that scales the nonlinear stiffness term
ϵ_{cpls}	A small parameter that scales the coupling term for the secondary beam
ϵ_{dmpf}	A small parameter that scales the damping term for the primary beam
ϵ_{dmps}	A small parameter that scales the damping term for the secondary beam
ϵ_{dp}	A small parameter that scales the damping term for the pendulum
ϵ_{ex}	A small parameter that scales the external excitation amplitude
ϵ_{ns}	A small parameter that scales the stiffness term for the pendulum
ϵ_p	A small perturbation parameter
ζ_1	The damping coefficient for the primary beam system
ζ_2	The damping coefficient for the secondary system
θ	Finite rotation of the pendulum
θ_0	The zeroth order perturbation term for the pendulum response
θ_1	First order perturbation term
μ	The coefficient of centripetal acceleration
Φ	The damping torque
ω	The first linear natural frequency of the pendulum
ω_1	The first linear natural frequency for the primary beam
ω_2	The first linear natural frequency for the secondary beam

Appendix

This section continues the non-resonant solution procedure for the parametric excitation of a pendulum. Continuing from equation (12), for calculating the non-resonant condition it is assumed that the numerical value of ω is far from the principal parametric resonance condition. Taking the secular terms, the coefficient of $exp(\tau_0)$, out from equation (12) and setting them to zero results in solvability conditions, equations (16) and (17).

$$\underbrace{-2}_{(4,1)}\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{D_1}_{(4,1)}\underbrace{A}_{(2,0,9)}\underbrace{i}_{(9,0)} - \underbrace{2}_{(4,1)}\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{\beta}_{(1,1)}\underbrace{A}_{(2,0,9)}\underbrace{i}_{(9,0)} - \underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{\gamma}_{(1,1)}\underbrace{3}_{(10,0)}\underbrace{A^2\bar{A}}_{(2,0,9,10)} = 0 \tag{16}$$

$$\underbrace{-2}_{(4,1)}\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{D_1}_{(4,1)}\underbrace{\bar{A}}_{(2,0,9)}\underbrace{-i}_{(9,0)} - \underbrace{2}_{(4,1)}\underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{\beta}_{(1,1)}\underbrace{A}_{(2,0,9)}\underbrace{-i}_{(9,0)} - \underbrace{\frac{\overbrace{\epsilon}^{(1,1)}}{\epsilon}}_{(2,1)}\underbrace{\gamma}_{(1,1)}\underbrace{3}_{(10,0)}\underbrace{\bar{A}^2A}_{(2,0,9,10)} = 0 \tag{17}$$

The first order perturbation equation (12), after removing the secular terms is shown in equation (18). It is possible

to calculate the solution for the first order perturbation equation, but due to the large amount of calculation required this step is skipped and only the amplitude and phase of the response is determined.

$$
\begin{aligned}
&\underbrace{D_0{}^2}_{(9,0)}\underbrace{\theta_1}_{(2,1)}+\underbrace{\theta_1}_{(2,1)}=\\
&-\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{\gamma}_{(1,1)}\underbrace{A^3}_{(2,0,9,10)}\underbrace{exp(\overbrace{i}^{(9,0)}\overbrace{3}^{(10,0)}\tau_0)}_{(9,0)}\\
&-\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{\gamma}_{(1,1)}\underbrace{\bar{A}^3}_{(2,0,9,10)}\underbrace{exp(\overbrace{-i}^{(9,0)}\overbrace{3}^{(10,0)}\tau_0)}_{(9,0)}\\
&+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{q\omega^2}_{(1,1)}\underbrace{\frac{1}{2}}_{(1,0,10)}\underbrace{A}_{(2,0,9)}\underbrace{exp(\overbrace{i}^{(*,12)}(\overbrace{\omega}^{(1,0,10)}+\overbrace{1}^{(9,0)})\tau_0)}_{(*,12)}\\
&+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{q\omega^2}_{(1,1)}\underbrace{\frac{1}{2}}_{(1,0,10)}\underbrace{A}_{(2,0,9)}\underbrace{exp(\overbrace{i}^{(*,12)}(\overbrace{-\omega}^{(1,0,10)}+\overbrace{1}^{(9,0)})\tau_0)}_{(*,12)}\\
&+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{q\omega^2}_{(1,1)}\underbrace{\frac{1}{2}}_{(1,0,10)}\underbrace{\bar{A}}_{(2,0,9)}\underbrace{exp(\overbrace{i}^{(*,12)}(\overbrace{\omega}^{(1,0,10)}-\overbrace{1}^{(9,0)})\tau_0)}_{(*,12)}\\
&+\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{q\omega^2}_{(1,1)}\underbrace{\frac{1}{2}}_{(1,0,10)}\underbrace{\bar{A}}_{(2,0,9)}\underbrace{exp(\overbrace{-i}^{(*,12)}(\overbrace{\omega}^{(1,0,10)}+\overbrace{1}^{(9,0)})\tau_0)}_{(*,12)}
\end{aligned}
\tag{18}
$$

The polar form of the $A(\tau_1)$ and its conjugate are given in equations (19) and (20). Note, $a(\tau_1)$ and $\alpha(\tau_1)$ are the amplitude and phase and are real functions.

$$A=\underbrace{\frac{a}{2}exp(i\alpha)}_{(10,0,19)} \tag{19}$$

$$\bar{A}=\underbrace{\frac{a}{2}exp(-i\alpha)}_{(10,0,20)} \tag{20}$$

The first derivatives of the complex function $A(\tau_1)$ and $\bar{A}(\tau_1)$ with respect to slow timescale τ_1, and based on $a(\tau_1)$ and $\alpha(\tau_1)$ are given in equations (21) and (22).

$$
\begin{aligned}
&\underbrace{D_1}_{(4,1)}\underbrace{A}_{(2,0,9)}=\\
&\frac{\overbrace{a'}^{(10,0,19,21)}}{\underbrace{2}_{(10,0,19)}}\underbrace{exp(i\alpha)}_{(10,0,19)}+\frac{\overbrace{a}^{(10,0,19)}\overbrace{i\alpha'}^{(10,0,19)}}{\underbrace{2}_{(10,0,19)}}\underbrace{exp(i\alpha)}_{(10,0,19,21)}
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
&\underbrace{D_1}_{(4,1)}\underbrace{\bar{A}}_{(2,0,9)}=\\
&\frac{\overbrace{a'}^{(10,0,20,22)}}{\underbrace{2}_{(10,0,20)}}\underbrace{exp(-i\alpha)}_{(10,0,20)}+\frac{\overbrace{a}^{(10,0,20)}\overbrace{-i\alpha'}^{(10,0,20)}}{\underbrace{2}_{(10,0,20)}}\underbrace{exp(-i\alpha)}_{(10,0,20,22)}
\end{aligned}
\tag{22}
$$

Equations (19), (20), (21), and (22) are substituted into the *solvability condition* equations (16) and (17), resulting in equations (23) and (24).

$$
\begin{aligned}
&-\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,19)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{exp(i\alpha)}_{(10,0,19)}\overbrace{a'}^{(10,0,19,21)}\underbrace{i}_{(9,0)}-\\
&\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,19)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{exp(i\alpha)}_{(10,0,19,21)}\overbrace{a}^{(10,0,19)}\overbrace{\alpha'}^{(10,0,19)}\underbrace{i}_{(9,0)}\underbrace{i}_{(10,0,20)}-\\
&\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{\beta}_{(1,1)}\underbrace{exp(i\alpha)}_{(10,0,19)}\underbrace{a}_{(10,0,19)}\underbrace{i}_{(9,0)}-\\
&\frac{1}{\underbrace{2}_{(10,0,20)}\times(\underbrace{2}_{(10,0,19)})^{\overbrace{2}^{(2,0,9,10)}}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{e}^{(10,0,19)}\overbrace{i}^{(2,0,9,10)}}{\underbrace{exp(i\alpha)}_{(10,0,20)}}\\
&\underbrace{3}_{(10,0)}\underbrace{\gamma}_{(1,1)}\underbrace{a}_{(10,0,19)}\overbrace{2}^{(2,0,9,10)}\underbrace{a}_{(10,0,20)}=0
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
&-\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{exp(-i\alpha)}_{(10,0,20)}\overbrace{a'}^{(10,0,19,21)}\underbrace{-i}_{(9,0)}-\\
&\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{exp(-i\alpha)}_{(10,0,20,22)}\overbrace{a}^{(10,0,19)}\overbrace{\alpha'}^{(10,0,20)}\underbrace{i}_{(9,0)}\underbrace{i}_{(10,0,20)}-\\
&\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\underbrace{\beta}_{(1,1)}\underbrace{exp(-i\alpha)}_{(10,0,20)}\underbrace{a}_{(10,0,20)}\underbrace{-i}_{(9,0)}-\\
&\frac{1}{\underbrace{2}_{(10,0,20)}\times(\underbrace{2}_{(10,0,19)})^{\overbrace{2}^{(2,0,9,10)}}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{e}^{(10,0,20)}\overbrace{-i}^{(2,0,9,10)}}{\underbrace{exp(i\alpha)}_{(10,0,20)}}\\
&\underbrace{3}_{(10,0)}\underbrace{\gamma}_{(1,1)}\underbrace{a}_{(10,0,20)}\overbrace{2}^{(2,0,9,10)}\underbrace{a}_{(10,0,19)}=0
\end{aligned}
\tag{24}
$$

Equations (25) and (26) are the imaginary and real parts of equation (23), respectively.

$$\{-\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0)}}\frac{\overbrace{i}^{(9,0)}}{\underbrace{i}_{(25,0)}}\}\overbrace{a'}^{(10,0,19,21)} - \{\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0)}}\frac{\overbrace{i}^{(9,0)}}{\underbrace{i}_{(25,0)}}\}\underbrace{\beta}_{(1,1)}\underbrace{a}_{(10,0,19)} = 0 \tag{25}$$

$$\{\frac{\overbrace{2}^{(4,1)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19,21)}}{\underbrace{exp(i\alpha)}_{(26,0)}}\underbrace{i}_{(9,0)}\underbrace{i}_{(10,0,20)}\}\overbrace{a}^{(10,0,20)}\overbrace{\alpha'}^{(10,0,19)} -$$

$$\frac{1}{\underbrace{2}_{(10,0,20)}\times(\underbrace{2}_{(10,0,19)})^{\overbrace{2}^{(2,0,9,10)}}}\{\frac{\overbrace{\epsilon}^{(1,1)}}{\underbrace{\epsilon}_{(2,1)}}\frac{\overbrace{exp}^{(10,0,19)}(\overbrace{i}^{(10,0,19)}\ \overbrace{2}^{(2,0,9,10)})}{\underbrace{exp(i\alpha)}_{(10,0,20)}\underbrace{exp(i\alpha)}_{(26,0)}}\}$$

$$\underbrace{3}_{(10,0)}\underbrace{\gamma}_{(1,1)}(\underbrace{a}_{(10,0,19)})^{\overbrace{2}^{(2,0,9,10)}}\underbrace{a}_{(10,0,20)} = 0 \tag{26}$$

It is then possible to obtain the final encoded symbolic form of the oscillation amplitude (a) and the phase (α).

$$a = \underbrace{a_0}_{(27,0)}\underbrace{exp}_{(27,0)}(\{\frac{\overbrace{2}^{(4,1,27)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,27)}}{\underbrace{\epsilon}_{(2,1,27)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{\beta}_{(1,1,27)}T_1) \tag{27}$$

$$\alpha = \underbrace{\alpha_0}_{(28,0)} - \frac{\overbrace{3}^{(10,0,28)}}{\underbrace{2}_{(10,0,20)}\times(\underbrace{2}_{(10,0,19)})^{\underbrace{2}_{(2,0,9,10)}}}(\overbrace{a_0}^{(27,0,28)})^{\overbrace{2}^{(2,0,9,10)}}\times$$

$$\frac{\{\frac{\overbrace{\epsilon}^{(1,1,28)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp}^{(10,0,19)}(\overbrace{i}^{(10,0,19)}\ \overbrace{2}^{(2,0,9,10)})}{\underbrace{exp(i\alpha)}_{(10,0,20)}\underbrace{exp(i\alpha)}_{(26,0,28)}}\}\underbrace{\gamma}_{(1,1)}}{\{\frac{\overbrace{2}^{(4,1,20)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,28)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{2}_{(2,0,9,10)}\underbrace{\beta}_{(1,1,28)}}\times$$

$$\underbrace{exp}_{(27,0,28)}(\{\frac{\overbrace{2}^{(4,1,27)}}{\underbrace{2}_{(10,0,20)}}\frac{\overbrace{\epsilon}^{(1,1,27)}}{\underbrace{\epsilon}_{(2,1,28)}}\frac{\overbrace{exp(i\alpha)}^{(10,0,19)}}{\underbrace{exp(i\alpha)}_{(25,0,27)}}\frac{\overbrace{i}^{(9,0,27)}}{\underbrace{i}_{(25,0,27)}}\}\underbrace{-2}_{(2,0,9,10)}\underbrace{\beta}_{(1,1,28)}T_1) \tag{28}$$