On the application of the generating series for nonlinear systems with polynomial stiffness

T. Gowdridge, N. Dervilis, K. Worden
Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield
Mappin Street, Sheffield S1 3JD, UK

Abstract
Analytical solutions to nonlinear differential equations – where they exist at all – can often be very difficult to find. For example, Duffing’s equation for a system with cubic stiffness requires the use of elliptic functions in the exact solution. A system with general polynomial stiffness would be even more difficult to solve analytically, if such a solution was even to exist. Perturbation and series solutions are possible, but become increasingly demanding as the order of solution increases. This paper aims to revisit, present and discuss a geometric/algebraic method of determining system response which lends itself to automation. The method, originally due to Fliess and co-workers, makes use of the generating series and shuffle product, mathematical ideas founded in differential geometry and abstract algebra. A family of nonlinear differential equations with polynomial stiffness is considered; the process of manipulating a series expansion into the generating series follows and is shown to provide a recursive schematic, which is amenable to computer algebra. The inverse Laplace-Borel transform is then applied to derive a time-domain response. New solutions are presented for systems with general polynomial stiffness, both for deterministic and Gaussian white-noise excitation.

Key words: Generating series; Nonlinear system modelling; Polynomial stiffness; Gaussian white noise.

1 Introduction
One of the main problems in the discipline of nonlinear structural dynamics, is that the equations of motion concerned very rarely admit closed-form exact solutions, and this usually forces a dependence on approximate solutions [1]. The most common approximation methods are series solutions, often perturbation expansions in the coefficients of the nonlinear terms, which must be ‘small’ in order that the series converge and that low-order truncations are useful. An alternative series formulation is provided by the Volterra series [2, 3], which is not formally defined in perturbative terms, but as a functional series. One of the attractive features of the Volterra series is that the generalised coefficients of the series have physical interpretations; the coefficients are actually linear and nonlinear impulse responses, and their Fourier transforms can be regarded as Higher-dimensional Frequency Response Functions (HFRFs) [1]. Unfortunately, calculations with the Volterra series are very demanding in algebraic terms and rapidly become intractable (at least by hand), as the order of the expansion increases. One can bring computer algebra to bear on the problem, in order to automate calculations, but this does require a careful reformulation of the problem.

Fortunately, such a reformulation exists in the form of the generating series; this is an geometric-algebraic nonlinear system analysis developed in the 1980s, by Michel Fliess and co-workers [4–9]. The ground-breaking idea involved the representation of the Volterra series – a sequence of high-dimensional integrals – as a purely algebraic expansion. The cost of the approach was that the expansion variables were non-commutative. The over-riding benefit of the approach was that the operations in the series expansion could be implemented in computer algebra. Like the Laplace or Fourier transforms, the generating series offered a duality between time responses mediated by differential equations and a purely algebraic approach based on polynomials in the transform variables. Once solutions were established in the transform domain, they could be taken back into the time domain using the Laplace-Borel transform [10, 11]. Although the theory of the generating series is extremely elegant, it sadly did not gain a great deal of traction in the
structural dynamics community. However, with interest turning back towards HFRFs and concrete calculations using the Volterra series [1], it is arguably time to revisit the approach.

The modest objective of the current paper is to give an accelerated tutorial on the generating series and illustrate its use. A novel result presented here will be from the analysis of a family of nonlinear systems with general polynomial terms; in the notation of [4], the equation of motion is,

\[ \sum_{i=0}^{n} l_i \frac{d^i y(t)}{dt^i} + \sum_{i=2}^{m} \varepsilon_i y(t) = u(t) \] (1)

where the first term is a general linear differential operator. For a Single-Degree-of-Freedom (SDOF) oscillator, \( n = 2 \) and the first and second derivatives represent damping and inertia terms respectively. (The equation can always be scaled so that \( l_n = 1 \).) The second term on the LHS is the polynomial stiffness term.

With \( n = 2 \) and the order of nonlinearity \( m = 3 \), the equation becomes that of the asymmetric Duffing oscillator [1, 12],

\[ m\ddot{y} + c\dot{y} + ky + k_2 y^2 + k_3 y^3 = x(t) \] (2)

in the standard notation where the overall (mass) scale has been restored; \( y \) is the displacement (response) and \( x \) is the force (excitation). Setting \( k_2 = 0 \) gives the symmetric Duffing oscillator,

\[ m\ddot{y} + c\dot{y} + ky + k_3 y^3 = x(t) \] (3)

Previous papers using the generating series showed results for systems with a single nonlinear term; in the current paper, results will be given for the asymmetric quadratic-cubic equation (2). Responses for the system under both deterministic and random excitations will be considered.

The layout of the paper is as follows: Section 2 will provide the basic terminology and definitions in order that one can motivate the generating series. Section 3 sketches the basic of related perturbation theory and shows how to construct a diagrammatic representation of functional expansions. Section 4 outlines the main ideas of the generating series in the context of the asymmetric Duffing oscillator system, and Section 5 outlines how the series can be used to compute system responses. Section 6 discusses how one calculates responses to white noise excitations and the paper then concludes.

2 Background Theory and Definitions

2.1 Algebraic Structure

As mentioned above, the generating series is an expansion in non-commuting variables; this clearly means that the variables themselves are not standard real or complex numbers and/or the product of the variables is not as standard. In fact both of these complexities are present, in the algebra of interest. The ‘basis’ of the algebra is provided by a set \( X = \{ x_0, x_1, ..., x_n \} \) of symbols called the alphabet; the elements in \( X \) are called letters. The alphabet \( X \) generates a set \( X^* \), which is called the free monoid over \( X \), whose elements are sequences of the form \( x_{j_1} ... x_{j_n} \), and are called words [7, 13]. \( X^* \) is thus the set of all words formed from the letters of the alphabet \( X \). The ‘polynomial’ terms of the generating series will be sets of words in some \( X^* \), to be elaborated later.

Having defined the variables in the algebra of interest, it remains to specify how they are multiplied together; this is by using the shuffle product. The shuffle product is a binary operator which represents the sum of all the interleaved products formed from a riffle shuffle over the two operands. The product is best explained in terms of a number of basic identities, which are essential in working with the generating series:

1. \( 1 \sqcup 1 = 1 \)
2. \( 1 \sqcup w = w \sqcup 1 = w \)
3. \( x_j w \sqcup x'_j w' = x_j(w \sqcup x'_j w') + x'_j(x_j w \sqcup w') \)
4. \( x^k \sqcup x^{n-k} = \binom{n}{k}x^n \)

Of these identities, the most useful is arguably the third which provides a recursive means of evaluating the product when the words concerned are specified in terms of their constituent letters. An example of the product showing the basic riffle nature is,

\[
ab \sqcup cd = a(b \sqcup cd) + c(ab \sqcup d) = abcd + acbd + cabd + cadb + cdab
\]

In the shuffle product, the terms inside each argument do not change their order. For example, in the third case listed above, \( x_j \) always appears before \( w \), and \( x'_j \) always before \( w' \), where \( \{x_j, x'_j, \ldots\} \in X \) and \( \{w, w', \ldots\} \in X^* \) [13]. This recursive definition for the shuffle product of two generating series is discussed and explored in greater detail in section 4.1. The recursion for the shuffle product is completed when at least one argument is reduced to the identity element (above in cases 1 and 2).

While this algebra may seem rather strange and counter-intuitive, it will be shown that it arise naturally in automating the transition between the nonlinear differential equation of interest and the corresponding Volterra series and thence to the generating series. The algebra arises because of the presence of *iterated integrals*.

### 2.2 Iterated Integrals

There is nothing mysterious about iterated integrals, they are simply multiple integrals with the individual integrations carried out in a prescribed order. It is well-known that changing the order of integration in a multiple integral will change the integrand, and this is what will cause non-commutativity here. As a matter of notation, iterated integrals will be denoted here like,

\[
\int_0^t dx_0...dx_n
\]

and the convention will be that individual integrations will work inwards from the right in terms of variables; here the first integral will be over \( x_0 \) and the last will be over \( x_n \). In this example, all the limits on the integrals are the same and so the integrals are represented by a single symbol; in the general case, each integral would have its own symbol and limits and these would also be traversed working inwards, this time from the left. In the generating series algebra, single integrals will correspond to the letters and multiple integrals to the words of the corresponding free monoid.

It is possible to see how the shuffle product might arise on such an algebra; consider a product of two iterated integrals,

\[
\left(\int_0^t d\xi_{j_1}...d\xi_{j_0}\right) \left(\int_0^t d\xi_{k_1}...d\xi_{k_0}\right)
\]

After a certain amount of standard calculus, one finds that integration by parts results in,

\[
\int_0^t d\xi_{j_0}(\tau) \left[\left(\int_0^\tau d\xi_{j_{e-1}}...d\xi_{j_0}\right) \left(\int_0^\tau d\xi_{k_{e-1}}...d\xi_{k_0}\right)\right] + \int_0^t d\xi_{k_0}(\tau) \left[\left(\int_0^\tau d\xi_{j_{e-1}}...d\xi_{j_0}\right) \left(\int_0^\tau d\xi_{k_{e-1}}...d\xi_{k_0}\right)\right]
\]

and this is very suggestive of the relation for the shuffle product,

\[
(xw) \sqcup (x'w') = x[w \sqcup (x'w')] + x'[xw \sqcup w']
\]

In fact, this is evidence of the correspondence with nonlinear differential equations; the nonlinear terms in such equations engender products of iterated integrals which map to shuffle products in the algebra of the generating series. In the language of [5, 14]:

**Theorem 2.1.** The product of two analytic causal functionals is again an analytic causal functional of the same kind, the generating power series of which is the shuffle product of the two generating power series. Formally, this represents

\[
y_1 \times y_2 \Leftrightarrow g_1 \sqcup g_2
\]
This theorem can be extended to higher-order products of terms. In the differential equation, terms of the form $y^n$, interpreted as $y \times \ldots \times y$, $n$ times, map directly to ‘powers’ in the generating series where the product is the shuffle. The simplest way to demonstrate this will be via the concrete examples to be pursued shortly.

2.3 Volterra Series

As mentioned in the introduction, another key ingredient in methodology here is the Volterra series; this is essentially a functional Taylor series for the response of a nonlinear input-output system [2, 3]. The series generalises the Duhamel integral for a linear system $x(t) \rightarrow y(t)$, given by [1],

$$y(t) = \int_{-\infty}^{+\infty} h(\tau) x(t - \tau) d\tau$$  (9)

where $h(\tau)$ represents the impulse response of the system. For a nonlinear system, one obtains instead an infinite series,

$$y(t) = y_0(t) + y_1(t) + y_2(t) + \ldots + y_i(t) + \ldots$$  (10)

where the general term is,

$$y_i(t) = \int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} h_i(\tau_1, \ldots, \tau_i) x(t - \tau_1) \ldots x(t - \tau_i) d\tau_1 \ldots d\tau_i$$  (11)

which is of course, an iterated integral. The functions $h_i(\tau_1, \ldots, \tau_i)$ are the ‘coefficients’ in the functional expansion and have a direct physical interpretation as higher-dimensional impulse response functions [1]; they are termed Volterra kernels. Clearly, the problem of establishing a Volterra series is that of determining the kernels for a given nonlinear system. The multi-dimensional Fourier transforms also have a physical interpretation as higher-dimensional frequency response functions (HFRFs). One way to find the Volterra kernels is by determining the HFRFs directly and then using an inverse Fourier transform; although this sounds rather indirect, it is possible because the HFRFs can be found from the nonlinear equations of motion by harmonic probing as introduced, in [15].

In fact, a variant of the strategy just described will be used in this paper to find the Volterra kernels and system responses. Rather than computing objects in the Fourier domain and using an (inverse) Fourier transform to find time-domain objects, the idea will be to compute objects in the algebra of the generating series and transform back; the relevant transform is called the inverse Laplace-Borel transform. Like the Laplace transform, the forward map is easier to find than the inverse, so the usual means of inversion is to use a table of inverse Laplace-Borel transforms [6, 8, 16].

3 The Consolidated Expansion and a Diagrammatic Representation

As mentioned earlier, the best way of illustrating the difficult concepts here is via concrete examples. With this in mind, this section will single out the asymmetric Duffing oscillator (equation (2)) as the system of interest. The system is actually of considerable practical interest as it represents the lowest-order approximation to a general SDOF system with both odd and even nonlinearities. Before proceeding, it is important to note that the form of equation can be simplified without losing generality. By scaling the independent ($t$) and dependent variables ($x, y$), the number of parameters can be reduced, so that the equation becomes,

$$\ddot{y} + a\dot{y} + y + \varepsilon_1 y^2 + \varepsilon_2 y^3 = x(t)$$  (12)

and it is this form which is considered from now on, in order to simplify the notation and algebra. Note that there are parameters associated with each of the nonlinear terms, $\varepsilon_1$ (quadratic) and $\varepsilon_2$ (cubic); in a standard perturbation approach, these would be the expansion parameters, and this will also be the case here. It is useful at this point to look at the perturbation approach, even if it will not be pursued directly here; the formulation will show the complexity of the problem and also allow the construction of a useful and intuitive diagrammatic representation.
3.1 Developing the Diagrammatic Representation

The first stage in the analysis here is to pass to the frequency domain via the Fourier transform. The standard operations on equation (12) yield, via the convolution theorem,

\[(1 + i\alpha \omega - \omega^2)Y(\omega) + \varepsilon_1(Y * Y)(\omega) + \varepsilon_2(Y * Y * Y)(\omega)\]  

(13)

where,

\[(Y * Y)(\omega) = \int_{-\infty}^{\infty} Y(\omega - \Omega)Y(\Omega)d\Omega\]  

(14)

and,

\[(Y * Y * Y)(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Y(\omega - \Omega)Y(\Omega - \Omega')Y(\Omega')d\Omega d\Omega'\]  

(15)

These latter expressions are cumbersome and unsymmetrical and can be rewritten as symmetrical integrals,

\[(Y * Y)(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Y(\omega_1)Y(\omega_2)\delta(\omega - \omega_1 - \omega_2)d\omega_1 d\omega_2\]  

(16)

and,

\[(Y * Y * Y)(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Y(\omega_1)Y(\omega_2)Y(\omega_3)\delta(\omega - \omega_1 - \omega_2 - \omega_3)d\omega_1 d\omega_2 d\omega_3\]  

(17)

With these modifications, and a little more rearrangement, equation (13) becomes,

\[Y(\omega) = H(\omega)X(\omega) - \varepsilon_1 H(\omega)\int_{-\infty}^{\infty} Y(\omega_1)Y(\omega_2)d\mu_2 - \varepsilon_2 H(\omega)\int_{-\infty}^{\infty} Y(\omega_1)Y(\omega_2)Y(\omega_3)d\mu_3\]  

(18)

where the integral signs have been coalesced and the measures are \(d\mu_2 = \delta(\omega - \omega_1 - \omega_2)d\omega_1 d\omega_2\) and \(d\mu_3 = \delta(\omega - \omega_1 - \omega_2 - \omega_3)d\omega_1 d\omega_2 d\omega_3\). Furthermore \(H(\omega) = 1/(1 + i\alpha \omega - \omega^2)\), which is the standard FRF of the underlying linear system \((\varepsilon_1 = \varepsilon_2 = 0)\). Note that the equation is recursive; i.e. \(Y(\omega)\) is expressed as a nonlinear function of itself. In the case that \(\varepsilon_1\) and \(\varepsilon_2\) were small perturbation parameters, the equation could be used to compute \(Y(\omega)\) iteratively, starting from the response of the linear system. With this observation in mind, it makes sense to compare the result with the actual two-parameter perturbation expansion. In the time domain one has,

\[y(t) = y_{00}(t) + \varepsilon_1 y_{10}(t) + \varepsilon_2 y_{01}(t) + \varepsilon_1^2 y_{20}(t) + \varepsilon_1^2 \varepsilon_2^1 y_{11}(t) + \ldots = \sum_{j=0}^{\infty} \sum_{i=0}^{j} \varepsilon_1^i \varepsilon_2^{j-i} y_{i,j-i}(t)\]  

(19)

and the Fourier transform is,

\[Y(\omega) = Y_{00}(\omega) + \varepsilon_1 Y_{10}(\omega) + \varepsilon_2 Y_{01}(\omega) + \varepsilon_1^2 Y_{20}(\omega) + \varepsilon_1^2 \varepsilon_2^1 Y_{11}(\omega) + \ldots = \sum_{j=0}^{\infty} \sum_{i=0}^{j} \varepsilon_1^i \varepsilon_2^{j-i} Y_{i,j-i}(\omega)\]  

(20)

with the obvious notation.

Now equating equations (20) and (18) at each level of perturbation, one obtains an infinite sequence of equations; the first nine, corresponding to a truncation at third nonlinear order, are:
The rules associated with individual terms $Y_{ij}$ are:

**Rule 1:** The tree(s) will have $i$ vertices with 3 incident branches and $j$ vertices with 4 incident branches.

**Rule 2:** There will be $i + j$ nodes and $i + j$ solid lines in the tree.

**Rule 3:** A tree will have $i + 2j + 1$ dashed branches.

**Rule 4:** Any two distinct vertices are connected by a single path.

**Rule 5:** Frequency is conserved at a vertex. The sums of the frequencies either side of a vertex are equal.

The diagrammatic representation does not add anything to the algebraic expansion; however, exactly as it does in quantum field theory, it helps considerably in doing calculations by hand. Up to this point, the analysis has not really strayed beyond classical theory – although Fliess did draw upon the diagrammatic representation in [8]. The generating series proper will be introduced in the next section via its calculation for the asymmetric Duffing oscillator.
Figure 1: Diagrammatic representation of first nine terms of the consolidated expansion for the quadratic-cubic oscillator.

4 Generating Series for the Asymmetric Duffing Oscillator

As discussed above, the analysis will concentrate on the system specified in equation (1) at first, but then specialise to the asymmetric (quadratic-cubic) Duffing oscillator. The first stage in the analysis is to manipulate the equation into integral form; one integrates $n$ times in order to remove all the differential operators, with result,

$$y(t) + L_{n-1} \int_0^t y(\tau_1) + \ldots + L_0 \int_0^t d\tau_n \int_0^{\tau_n} d\tau_{n-1} \ldots \int_0^{\tau_2} y(\tau_1) d\tau_1 + \sum_{i=2}^m \epsilon_i \int_0^t d\tau_n \ldots \int_0^{\tau_2} y(\tau_1) d\tau_1 = \int_0^t d\tau_n \ldots \int_0^{\tau_2} u(\tau_1) d\tau_1$$

if one assumes that all relevant initial conditions are zero. The equation of motion now consists of iterated integrals and can be converted into the generating series domain. The symbol $g$ will denote the generating series associated with $y(t)$ here; the two basis letters associated with the free monoid will be denoted: $x_0 = \int_0^t y(\tau) d\tau$ and $x_1 = \int_0^t x(\tau)$. The rules for the transformation follow from Fliess’ fundamental formula and the Peano-Baker formula [19], as detailed in [4]. The main formal rules for the transformation are as follows:

1. The transform acts on linear combinations linearly.
2. Just as differentiation in the Fourier transform is represented by pre-multiplication by $i\omega$, integration in the $g$-domain is represented by pre-multiplication by the word $x_0$.
3. $n^{th}$ powers of $y$ will transform to $n$-fold shuffle produces of $g$. 
Applying these rules to equation (21) yields,
\[ g + l_n x_0 g + \ldots + l_1 x_0^{n-1} g + x_0^n (\varepsilon_2 g \cup g + \ldots + \varepsilon_m g \cup \ldots \cup g) = x_0^{n-1} x_1 \]  
(22)

or the more compact form,
\[ (1 + \sum_{j=0}^{n-1} l_j x_0^{n-j}) g + x_0^n \sum_{i=2}^{m} \varepsilon_i g^{\cup_i} = x_0^{n-1} x_1 \]  
(23)

where \( g^{\cup_i} = g \cup \ldots \cup g, i \) times.

Equation (23) can be further simplified by factorising the polynomial in \( x_0 \) that multiplies \( g \) as follows,
\[ 1 + \sum_{j=0}^{n-1} l_j x_0^{n-j} = p \prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}, \quad \alpha_1 + \alpha_2 + \ldots + \alpha_p = n \]  
(24)

One can now formally write equation (23) as,
\[ g = g_0 + \frac{x_0^n \sum_{i=2}^{m} \varepsilon_i g^{\cup_i}}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \]  
(25)

where,
\[ g_0 = \frac{x_0^{n-1} x_1}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \]  
(26)

is seen to be the generating series representation of the underlying linear system.

The solution to equation (22) can now be constructed recursively; starting with \( g_0 \), one computes,
\[ g_{i+1} = -\frac{x_0^n \times \sum_{j=2}^{m} \varepsilon_i \sum_{\nu_1+\ldots+\nu_j = i} g_{\nu_1} \cup \ldots \cup g_{\nu_j}}{\prod_{i=0}^{p} (1 - a_i x_0)^{\alpha_i}} \]  
(27)

and the representation of the full nonlinear system response is then,
\[ g = g_0 + g_1 + \ldots + g_i \]  
(28)

At this point, it is important to recall that the algebra of the generating series is not commutative, so objects like equation (26) are actually ambiguous. A careful analysis reveals that \( g_0 \), actually takes the form \( R_1(x_0) x_1 R_2(x_0) x_1 \ldots x_1 R_p(x_0) \) where \( R_j(x_0) \) represents a rational fraction and \( \{i_1, i_2, \ldots, i_p\} \in \{0,1,2\} \) [7]. The quotient in the recursive scheme of equation (27 is of a similar form; meaning that all the successive \( g_i \) terms will also be of this form. In fact, the general form of the terms of interest can be written,
\[ \frac{1}{(1 - a_0 x_0)^{\alpha_0}} x_1 \frac{1}{(1 - a_1 x_0)^{\alpha_1}} x_1 \ldots x_1 \frac{1}{(1 - a_p x_0)^{\alpha_p}} \]  
(29)

Expressions of this type can be simplifying by using the identity,
\[ \frac{1}{(1 - a x_0)^\alpha} = \frac{1}{(1 - a x_0)^{\alpha-1}} + \frac{a x_0}{(1 - a x_0)^{\alpha-1}} \]  
(30)

and decomposing as partial fractions. In this way, by repeated application of the identity, all the exponents in the denominators can ultimately be reduced to unity, and the general object of interest becomes,
\[
\frac{1}{(1 - a_0 x_0)^{x_{i_1}}} \frac{1}{(1 - a_1 x_0)^{x_{i_2}} \cdots x_{i_p}} \frac{1}{(1 - a_p x_0)\)}
\]  
(31)

The general problem of computing shuffle products is thus reduced to that of computing shuffle products of terms like that above. Summation of such terms is not an issue as the shuffle product is distributive over addition [13].

### 4.1 Shuffle Product of Series Terms

Shuffle products of terms of the specific form in (31) can now be considered in more detail. Suppose the two terms of interest are,

\[
g_1^p = \frac{1}{1 - b_0 x_0} x_{i_1} \frac{1}{1 - b_1 x_0} x_{i_2} \cdots x_{i_p} \frac{1}{1 - b_p x_0} = g_1^{p-1} \frac{x_{i_p}}{1 - b_p x_0}
\]  
(32)

and,

\[
g_2^q = \frac{1}{1 - d_0 x_0} x_{j_1} \frac{1}{1 - d_1 x_0} x_{j_2} \cdots x_{j_q} \frac{1}{1 - d_q x_0} = g_2^{q-1} \frac{x_{j_q}}{1 - d_q x_0}
\]  
(33)

where \( \{p, q\} \in \mathbb{N} \) and \( \{i_1, \ldots, i_p, j_1, \ldots, j_q\} \subseteq \{0, 1\} \).

By assuming that the generating series is represented as a series of products, one can readily compute their shuffle products; results can be defined recursively as shuffle products of lower-order terms. This recursion ends when any of the following terms occur: \( x \sqcup 1, 1 \sqcup x, \) or \( 1 \sqcup 1 \). The process halts as the shuffle product of a term with the identity element simply returns the term itself.

Looking in more detail: the term \( \frac{1}{1 - b_0 x_0} \) can be rearranged into \( 1 + \frac{b_0 x_0}{1 - b_0 x_0} \); this operation can be applied to the highest-order fraction to give,

\[
g_1^p \sqcup g_2^q = \left[ g_1^{p-1} x_{i_p} \left( 1 + \frac{b_p x_0}{1 - b_p x_0} \right) \right] \sqcup \left[ g_2^{q-1} x_{j_q} \left( 1 + \frac{d_q x_0}{1 - d_q x_0} \right) \right]
\]  
(34)

Recalling that the shuffle product is distributive over addition [13]: the above product can be expanded to give [7],

\[
g_1^p \sqcup g_2^q = \left[ g_1^{p-1} x_{i_p} \frac{1}{1 - b_p x_0} \sqcup g_2^{q-1} \right] x_{j_q} + \left[ g_1^{p-1} \sqcup g_2^{q-1} x_{j_q} \frac{1}{1 - d_q x_0} \right] x_{i_p}
\]  
\[
+ \left[ g_1^{p-1} x_{i_p} \frac{1}{1 - b_p x_0} \sqcup g_2^{q-1} x_{j_q} \frac{1}{1 - d_q x_0} \right] b_p x_0 + \left[ g_1^{p-1} x_{i_p} \frac{1}{1 - b_p x_0} \sqcup g_2^{q-1} x_{j_q} \frac{1}{1 - d_q x_0} \right] d_q x_0
\]  
(35)

(remembering that order of the terms is important).

Careful regrouping of terms gives a compact recursion for the shuffle product of two generating series of the form [7],

\[
g_1^p \sqcup g_2^q = \frac{1}{1 - (b_p + d_q) x_0} \left[ (g_1^p \sqcup g_2^q) x_{j_q} + (g_1^{p-1} \sqcup g_2^q) x_{i_p} \right]
\]  
(36)

Noting that \( \frac{1}{1 - b_0 x_0} \sqcup \frac{1}{1 - d_0 x_0} = \frac{1}{1 - (b_0 + d_0) x_0} \).

Fortunately, the shuffle product is associative [13] i.e., \( (g_j \sqcup g_k) \sqcup g_\ell = g_j, \sqcup (g_j \sqcup g_k) \) and this means that equation (36) can be extended straightforwardly (if tediously) to higher-order products, because the order in which the pairwise products are taken does not matter.

One of the strengths of the generating series approach is that the formalism above is amenable to computer implementation; the calculations presented in this paper are the result of a Python implementation of the necessary algebra.
4.2 The Asymmetric Duffing Oscillator

When equation (22) is applied to the asymmetric Duffing equation in the canonical form in (12), the result is,

\[ g + x_0 g + x_0^2 g + \varepsilon_1 x_0^2 [g \| g] + \varepsilon_2 x_0^2 [g \| g \| g] = x_0 x_1 \]  

(37)

By collecting like terms and factorising the quadratic expression in \( x_0 \), this equation can be rearranged into the form,

\[ g = \frac{x_0 x_1}{(1 - a_1 x_0)(1 - a_2 x_0)} - \varepsilon_1 \frac{x_0^2}{(1 - a_1 x_0)(1 - a_2 x_0)} g \| g - \varepsilon_2 \frac{x_0^2}{(1 - a_1 x_0)(1 - a_2 x_0)} g \| g \| g \]  

(38)

where \((1 - a_1 x_0)(1 - a_2 x_0) = 1 + a x_0 + x_0^2\).

The generating series solution can now be obtained for the oscillator; an iterative procedure can be followed where,

\[ g_{i+1} = -\frac{x_0}{1 - a_1 x_0} \frac{x_0}{1 - a_2 x_0} \left\{ \varepsilon_1 \sum_{i_1 + i_2 = i} [g_{i_1} \| g_{i_2}] + \varepsilon_2 \sum_{j_1 + j_2 + j_3 = i} [g_{j_1} \| g_{j_2} \| g_{j_3}] \right\} \]  

(39)

The iteration begins with \( g_1 \) from the underlying linear system with \( \varepsilon_1 = \varepsilon_2 = 0 \),

\[ g_1 = \frac{1}{1 - a_1 x_0} x_0 \frac{1}{1 - a_2 x_0} x_1 \]  

(40)

The generating series \( g \), is then the sum,

\[ g = g_0 + g_1 + g_2 + \ldots + g_n + \ldots \]  

(41)

allowing for the possibility for a constant offset \( g_0 \) in the response.

For the calculation here, only the first two iterations will be displayed. The additional nonlinear term here causes the iterations to have exponentially more terms compared to the single nonlinearities considered in other works [5, 7].

For a more compact notation, the terms in the generating series, as shown in equation (32) for example, can be expressed in the form of a \((2 \times p)\) array; in the calculation here for example, one has,

\[ g_1 = \begin{bmatrix} x_0 & x_1 \\ -a_1 & -a_2 \end{bmatrix} \]

where the notation shows a word in the numerator in the first row and the corresponding coefficient in the denominator in the second row. Each column represents a term in the rational fraction, and the overall coefficient/multiplier is found outside of the array.

The first iteration of the algorithm gives,

\[ g_1 = -\frac{x_0}{1 - a_1 x_0} \frac{x_0}{1 - a_2 x_0} \left\{ \varepsilon_1 [g_0 \| g_0] + \varepsilon_2 [g_0 \| g_0 \| g_0] \right\} \]  

(42)

(noticing that order is unimportant in expressions considering a single letter).

Now expanding the shuffle products using equation (36) yields,
The analysis up to now has allowed the input-output relationship for the system to be expressed in terms of the form \( q_{\text{back}} \) made with general \( x \), as encoded in the letter \( g \), as there are 360 terms in the full expansion.

For \( g_2 \),

\[
g_2 = \frac{x_0 x_0}{1 - a_1 x_0} \left\{ \varepsilon_1 [g_1 \sqcup g_0 \sqcup g_1] + \varepsilon_2 [g_1 \sqcup g_0 \sqcup g_0 + g_0 \sqcup g_1 \sqcup g_0 + g_0 \sqcup g_0 \sqcup g_1] \right\} \tag{43}
\]

Only the first 10 terms have been shown for \( g_2 \), as there are 360 terms in the full expansion.

## 5 Determining System Response

The analysis up to now has allowed the input-output relationship for the system to be expressed in terms of the generating series algebra. In order to compute an actual response, one needs to substitute for the relevant excitation \( x(t) \), as encoded in the letter \( x_1 \) in the free monoid, and then transform back to the time domain. The analysis has provided terms of a specific form; words in the letters \( x_0 \) and \( x_1 \) and rational fractions of them. If the transformation back is made with general \( x_1 \), the result will be a Volterra expansion, and one will be able to read off the Volterra kernels. Each term in the series – of the form given by equation (31) – corresponds to a specific iterated integral. Each appearance of \( x_0 \) represents an integration, so a term with \( q \) occurrences of \( x_0 \) represents a \( q \)-fold iterated integral. Fließ and co-workers computed the general inverse transform of a \( q \)-fold product in the generating series, it takes the form \( [7] \).
\[
\int_0^t \int_0^{\tau_q} \cdots \int_0^{\tau_2} \int_0^{\tau_1} f_{a_1}^a(t - \tau_q) \cdots f_{a_{q-1}}^a(\tau_2 - \tau_1) f_{a_q}^a(\tau_1) x(\tau_1) \cdots x(\tau_1) d\tau_q \cdots d\tau_1
\]  
(44)

where

\[
f_a^a = \left[ \sum_{j=0}^{a-1} \left( \frac{a-1}{j} \right) \frac{a^j t^j}{j!} \right] e^{at}
\]  
(45)

This relationship shows how the generating series and Volterra series are so strongly linked. The correspondence between the terms in equation (41) and the Volterra terms can be shown to be,

\[
g_0 \leftrightarrow y_0 = h_0
\]  
(46)
\[
g_1 \leftrightarrow y_1 = \int_0^\infty h_1(\tau_1) x(\tau_1) d\tau_1
\]  
(47)
\[
g_2 \leftrightarrow y_2 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_2, \tau_1) x(\tau_2) x(\tau_1) d\tau_2 d\tau_1
\]  
(48)

By computing the inverse Laplace-Borel transform of the generating series derived for the oscillators in Section 4.2, the Volterra kernels are determined. For specific excitations \(x(t)\), the system response can be computed. The relevant inverse transforms are tabulated below:

| \(x(t)\)          | \(g[x(t)]\)          |
|-------------------|-----------------------|
| Unit Step \(\frac{t^n}{n!}\) | \(1 \quad x_0^n\) |
| \(\sum_{i=0}^{n-1} \left( \frac{a^i t^i}{i!} \right) e^{at}\) cos(\(\omega t\)) | \((1 - ax_0)^{-n}\) |
| \(1 + \omega^2 x_0^{-1}\) | \(1 + \omega^2 x_0^{-1}\) |

Table 1: Laplace-Borel Transforms of Common Functions [7, 11]

Rather than give the asymmetric Duffing system response for a deterministic excitation, a little more work will allow characterisation of the response to a random excitation.

6 Response to Gaussian White Noise

Clearly, the machinery provided up to now can not provide a time-series response to a truly random excitation; however, it can be adapted to give output statistics of the response and this is simplest when the excitation is a Gaussian white noise process. Such a process is specified by a demand that it have zero mean, and an auto-correlation function of the form [20],

\[
E[x(t)x(\tau)] = \langle x(t)x(\tau) \rangle = \sigma^2 \delta(t - \tau)
\]  
(49)

where \(\sigma^2\) denotes the ‘noise-power’ (one must take care in interpreting this as a variance) and angle brackets denote expectations.

The most basic statistic one can estimate is the mean of the response, or its expectation \(\langle y(t) \rangle\). In [15], the authors developed an appropriate form of the Volterra series for random excitation, based on stochastic calculus [21]. This formulation was adapted by Fliess [5], in order to compute statistics from the generating series. The expectations are interpreted as ensemble averages so that one can take the expectations in the transform domain and then map back. In this way \(\langle g \rangle\) corresponds to \(\langle y(t) \rangle\), and the higher-order statistics \(\langle y(t)^n \rangle\) are obtained by mapping back the shuffle products \(\langle g^{[n]} \rangle\).

As usual now, it is sufficient to consider only the calculation for terms of the form shown in equation (31); the basic rules are [22],
As discussed above, because of the third condition in the recursion (50), many terms are zero and do not contribute. The autocorrelation of the response is a little more complicated, this has the form,

$$ S_{yy} = \langle y(t_1) y(t_2) \rangle $$

and the product of $g$s will produce a shuffle product in the domain of the generating series.

For the asymmetric Duffing oscillator under investigation here, performing the ensemble average for all the terms in the generating series, the following result is obtained,

$$ \langle g \rangle = -4\varepsilon_1 \left( \frac{\sigma^2}{2} \right) \left[ x_0 \begin{array}{cccc} a_1 & x_0 & x_0 & x_0 \\ -a_1 & -a_2 & -2a_1 & -a_1 - a_2 \\ -a_2 & -2a_1 & -a_1 - a_2 & -2a_1 - 2a_2 \end{array} \right] $$

$$ + 48\varepsilon_1\varepsilon_2 \left( \frac{\sigma^2}{2} \right)^2 \left[ x_0 \begin{array}{cccc} x_0 & x_0 & x_0 & x_0 \\ -a_1 & -a_2 & -2a_1 & -a_1 - a_2 \end{array} \right] $$

$$ + 48\varepsilon_1\varepsilon_2 \left( \frac{\sigma^2}{2} \right)^2 \left[ x_0 \begin{array}{cccc} x_0 & x_0 & x_0 & x_0 \\ -a_1 & -a_2 & -2a_1 & -a_1 - a_2 \end{array} \right] $$

As discussed above, because of the third condition in the recursion (50), many terms are zero and do not contribute. For example, the second significant term in the expansion above is actually the 39th term from the total of 360 in $g_2$.

To move the computation forward, it is necessary to carry out the partial fraction calculations implicit in the terms in equation (52). At this point, it is useful to specify numerical values for $a_1$ and $a_2$, as the partial fractions calculations can be cumbersome when carried out algebraically. In the case of the asymmetric Duffing equation, the system parameters are $m, c, k_1, k_2, k_3$. The relevant parameters in the generating series can then be calculated via the scaled version of the Duffing equation in equation (12); starting with values here of $m = 1 kg$, $c = 15 N sm^{-1}$, $k_1 = 25 N m^{-1}$, $k_2 = 625 N m^{-1}$ and $k_3 = 7500 N m^{-1}$, one obtains $a_1 = -\frac{1}{2}(3 + \sqrt{5})$, $a_2 = -\frac{1}{2}(3 + \sqrt{5})$, $\varepsilon_1 = 1$ and $\varepsilon_2 = 0.5$.

By decomposing the terms in equation (52) into partial fractions and applying the inverse Laplace-Borel transforms as given in Table 1, the mean response of the system $E[y(t)]$ can be computed; the result is,

$$ \langle y(t) \rangle = -4\varepsilon_1 \left( \frac{\sigma^2}{2} \right) \left[ 0.08332 - 0.000535 \frac{e^{-\frac{t}{1.309}}}{0.1910} + 0.02288 \frac{e^{-\frac{t}{0.3333}}}{0.3333} - 0.6315 \frac{e^{-\frac{t}{2.618}}}{2.618} \\ - 0.03514 \frac{e^{-\frac{t}{0.3820}}}{0.3820} + 0.2409 \frac{e^{-\frac{t}{1.309}}}{0.1910} \right] + 48\varepsilon_1\varepsilon_2 \left( \frac{\sigma^2}{2} \right)^2 \left[ 0.0001508 - 0.0005778 \frac{e^{-\frac{t}{0.01667}}}{0.01667} \\ - 0.0002746 \frac{e^{-\frac{t}{0.1910}}}{0.1910} - 0.000006524 \frac{e^{-\frac{t}{0.9551}}}{0.9551} \right] $$

$$ + \left[ 0.004057 \frac{e^{-\frac{t}{2.618}}}{0.1910^2} - 0.000002157 \frac{e^{-\frac{t}{0.1910}}}{0.1910} \right] + 0.0007298 \frac{e^{-\frac{t}{0.3333}}}{0.3333} + 0.0001947 \frac{e^{-\frac{t}{1.309}}}{1.309} \right] + \ldots \quad (53) \]
Choosing the somewhat arbitrary value $\sigma = 1$, the result in Figure 2 is obtained. One observes a transient which occurs from ‘switching on’ the excitation at $t = 0$. In fact, because the asymmetric Duffing oscillator has stationary response if the input is stationary, the expectation will tend to a constant value as $t \to \infty$; because the restoring force is asymmetric, that constant value will be non-zero.

![Figure 2: Expectation of the response of an asymmetric Duffing oscillator to a Gaussian white-noise excitation.](image)

7 Conclusions

Long conclusions are not warranted here as the aim of this paper was simply to revisit the generating series of Fliess and co-workers, as an elegant means of nonlinear system analysis. In order to introduce a novel element, the analysis has been extended beyond previous work in order to deal with the case of two nonlinear terms in the equation of motion. This new analysis is also extended to the diagrammatic representation, where the presence of two nonlinearities produces two types of vertices in the ‘Feynman’ rules for the diagrams.

Further work on the series is considering how it can be used in an automated manner in order determine Higher-order Frequency Response Functions for nonlinear structural dynamic systems.

Acknowledgements

The authors would like to thank the UK EPSRC for funding through the Established Career Fellowship EP/R003645/1 and the Programme Grant EP/R006768/1.

References

[1] K. Worden and G.R. Tomlinson. Nonlinearity in Structural Dynamics: Detection, Identification and Modelling. Institute of Physics Publishing, 2001.

[2] V. Volterra. Theory of Functionals and of Integral and Integro-Differential Equations. Blackie & Son Limited, 1930.

[3] J.F. Barrett. The use of functionals in the analysis of non-linear physical systems. International Journal of Electronics, 15:567–615, 1963.

[4] M. Fliess. Fonctionnelles causales non linéaires et indéterminées non commutatives. Bulletin de la Société Mathématique de France, 109:3–40, 1981.
[5] M. Fliess and F. Lamnabhi-Lagarrigue. Application of a new functional expansion to the cubic anharmonic oscillator. *Journal of Mathematical Physics*, 23:495–502, 1982.

[6] M. Lamnabhi. A new symbolic calculus for the response of nonlinear systems. *Systems & Control Letters*, 2:154–162, 1982.

[7] M. Fliess, M. Lamnabhi, and F. Lamnabhi-Lagarrigue. An algebraic approach to nonlinear functional expansions. *IEEE Transactions on Circuits and Systems*, 30:554–570, 1983.

[8] M. Lamnabhi. Functional analysis of nonlinear circuits: a generating power series approach. *IEE Proceedings H (Microwaves, Antennas and Propagation)*, 133:375–384, 1986.

[9] F. Lamnabhi-Lagarrigue. *Application des variables non commutatives à des calculs formels en statistique non linéaire*. PhD thesis, Université Paris-Sud, 1980.

[10] F. Lamnabhi-Lagarrigue and M. Lamnabhi. Algebraic computation of the solution of some nonlinear differential equations. In *Proceedings of the European Computer Algebra Conference*, pages 204–211. Springer, 1982.

[11] Y. Li and W.S. Gray. The formal Laplace-Borel transform, Fliess operators and the composition product. In *Proceedings of the 36th Southeastern Symposium on System Theory*, pages 333–337. IEEE, 2004.

[12] G. Duffing. *Erzwungene Schwingungen bei Veränderlicher Eigenfrequenz und ihre Technische Bedeutung*. F. Vieweg and Sohn, 1918.

[13] C. Reutenauer. *Free Lie Algebras*. Elsevier, 1993.

[14] R. Ree. Lie elements and an algebra associated with shuffles. *Annals of Mathematics*, pages 210–220, 1958.

[15] E. Bedrosian and S.O. Rice. The output properties of Volterra systems (nonlinear systems with memory) driven by harmonic and Gaussian inputs. *Proceedings of the IEEE*, 59:1688–1707, 1971.

[16] J.K. Lubbock and V.S. Bansal. Multidimensional Laplace transforms for solution of nonlinear equations. *Proceedings of the Institution of Electrical Engineers*, 116:2075–2082, 1969.

[17] J.B. Morton and S. Corrsin. Consolidated expansions for estimating the response of a randomly driven nonlinear oscillator. *Journal of Statistical Physics*, 2:153–194, 1970.

[18] R.P. Feynman. Space-time approach to non-relativistic quantum mechanics. *Reviews of Modern Physics*, 20:367–387, 1948.

[19] F.R. Gantmakher. *The Theory of Matrices*. American Mathematical Society, 2000.

[20] L. Arnold. *Stochastic Differential Equations*. Wiley-Blackwell, 1974.

[21] K. Ito. Stochastic integral. *Proceedings of the Imperial Academy*, 20:519–524, 1944.

[22] F. Lamnabhi-Lagarrigue and M. Lamnabhi. Algebraic computation of the statistics of the solution of some nonlinear stochastic differential equations. In *Proceedings of the European Conference on Computer Algebra*, pages 55–67. Springer, 1983.