Operator Calculus Approach to Solving Analytic Systems

Philip Feinsilver[1] and René Schott[2]

[1] Southern Illinois University, Carbondale, IL. 62901, U.S.A.
pfeinsil@math.siu.edu
[2] Université Henri Poincaré-Nancy 1, BP 239, 54506 Vandoeuvre-lès-Nancy, France
schott@loria.fr

Abstract. Solving analytic systems using inversion can be implemented in a variety of ways. One method is to use Lagrange inversion and variations. Here we present a different approach, based on dual vector fields.

For a function analytic in a neighborhood of the origin in the complex plane, we associate a vector field and its dual, an operator version of Fourier transform. The construction extends naturally to functions of several variables.

We illustrate with various examples and present an efficient algorithm readily implemented as a symbolic procedure in Maple while suitable as well for numerical computations using languages such as C or Java.

1 Introduction

We introduce the operator calculus necessary to present our approach to (local) inversion of analytic functions. It is important to note that this is different from Lagrange inversion and is based on the flow of a vector field associated to a given function. It appears to be theoretically appealing as well as computationally effective.

Acting on polynomials in x, define the operators:

$$D = \frac{d}{dx} \text{ and } X = \text{multiplication by } x.$$

They satisfy commutation relations $[D, X] = I$, where I, the identity operator commutes with both D and X. Abstractly, the Heisenberg-Weyl algebra is the associative algebra generated by operators $\{A, B, C\}$ satisfying $[A, B] = C$, $[A, C] = [B, C] = 0$. The *standard* HW algebra is the one generated by the realization $A = D$, $B = X$, $C = I$.

An *Appell system* is a system of polynomials $\{y_n(x)\}_{n\geq 0}$ that is a basis for a representation of the standard HW algebra with the following properties:

1. y_n is of degree n in x
2. $D\, y_n = n\, y_{n-1}$

In several variables, $\mathbf{x} = (x_1, \ldots, x_N)$, with multi-indices $\mathbf{n} = (n_1, \ldots, n_N)$, and corresponding monomials
$\mathbf{x}^{\mathbf{n}} = x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N}$. Denote the partial derivative operators by $D_i = \dfrac{\partial}{\partial x_i}$ and corresponding multiplication operators by X_i. Then $[D_j, X_i] = \delta_{ij}\, I$. An Appell system is a system of polynomials $\{y_{\mathbf{n}}\}$ in the variables $\mathbf{x}$ such that:

1. the top degree term of $y_{\mathbf{n}}$ is a constant multiple of $\mathbf{x}^{\mathbf{n}}$
2. $D_i\, y_{\mathbf{n}} = n_i\, y_{\mathbf{n}-\mathbf{e}_i}$, where $\mathbf{e}_i$ has all components zero except for 1 in the i^{th} position

G.-C. Rota [3] is well-known for his *umbral calculus* development of special polynomial sequences, called *basic sequences.* From our perspective, these are "canonical polynomial systems" in the sense that they provide polynomial representations of the Heisenberg-Weyl algebra, in realizations different from the standard one. Our idea [2, 1] is to illustrate explicitly the rôle of vector fields and their duals, using operator calculus methods for working with the latter (in our volumes — this viewpoint is prefigured in [3]).

The main feature of our approach is that the action of the vector field may be readily calculated while the action of the dual vector field on exponentials is identical to that of the vector field. Then we note that acting iteratively with a vector field on polynomials involves the complexity of the coefficients, while acting iteratively with the dual vector field always produces polynomials from polynomials. So we can switch to the dual vector field for calculations.

Specifically, fix a neighborhood of 0 in $\mathbf{C}$. Take an analytic function $V(z)$ defined there, normalized to $V(0) = 0$, $V'(0) = 1$. Denote $W(z) = 1/V'(z)$ and $U(v)$ the inverse function, i.e., $V(U(v)) = v$, $U(V(z)) = z$. Then $V(D)$ is defined by power series as an operator on polynomials in x and $[V(D), X] = V'(D)$ so that $[V(D), XW(D)] = I$. In other words, $V = V(D)$ and $Y = XW(D)$ generate a representation of the HW algebra on polynomials in x. The basis for the representation is $y_n(x) = Y^n 1$, i.e., Y is a *raising operator*. And $V y_n = n\, y_{n-1}$ so that V is the corresponding *lowering operator.* The $\{y_n\}_{n\geq 0}$ form a system of *canonical polynomials* or generalized Appell system. The operator of multiplication by x is given by $X = YV'(D) = YU'(V)^{-1}$, which is a *recursion operator* for the system.

We identify vector fields with first-order partial differential operators. Consider a variable A with corresponding partial differential operator ∂_A. Given V as above, let $\tilde{Y}$ be the vector field $\tilde{Y} = W(A)\partial_A$. Then we observe the following identities

$$\tilde{Y}\, e^{Ax} = xW(A)\, e^{Ax} = xW(D)\, e^{Ax}$$

as any operator function of D acts as a multiplication operator on e^{Ax}. The important property of these equalities is that Y and $\tilde{Y}$ commute, as they involve independent variables. So we may iterate to get

$$\exp(t\tilde{Y})e^{Ax} = \exp(tY)e^{Ax} \tag{1}$$

On the other hand, we can solve for the left-hand side of this equation using the method of characteristics. Namely, if we solve

$$\dot{A} = W(A) \tag{2}$$

with initial condition $A(0) = A$, then for any smooth function f,

$$e^{t\tilde{Y}} f(A) = f(A(t))$$

Thus,

$$\exp(tY)e^{Ax} = e^{xA(t)}$$

To solve equation (2), multiply both sides by $V'(A)$ and observe that we get

$$V'(A)\, \dot{A} = \frac{d}{dt}\, V(A(t)) = 1$$

Integrating yields,

$$V(A(t)) = t + V(A) \qquad \text{or} \qquad A(t) = U(t + V(A))$$

Or, writing v for t,

$$\exp(vY)e^{Ax} = e^{xU(v+V(A))} \tag{3}$$

We can set $A = 0$ to get

$$\exp(vY)1 = e^{xU(v)}$$

on the one hand while

$$e^{vY}1 = \sum_{n=0}^{\infty} \frac{v^n}{n!}\, y_n(x)$$

In summary, we have the expansion of the exponential of the inverse function

$$e^{xU(v)} = \sum_{n=0}^{\infty} \frac{v^n}{n!}\, y_n(x)$$

or

$$\sum_{m=0}^{\infty} \frac{x^m}{m!}\,(U(v))^m = \sum_{n=0}^{\infty} \frac{v^n}{n!}\, y_n(x) \tag{4}$$

This yields an alternative approach to inversion of the function $V(z)$ rather than using Lagrange's formula. We see that the coefficient of $x^m/m!$ yields the expansion of $(U(v))^m$. In particular, $U(v)$ itself is given by the coefficient of x on the right-hand side.

Specifically, we have :

Theorem 1. *The coefficient of $x^m/m!$ in $Y^n 1$ is equal to $\tilde{Y}^n A^m\big|_{A=0}$, each giving the coefficient of $v^n/n!$ in the expansion of $U(v)^m$.*

Proof. Expand both sides of equation (1), using v for t, in powers of x and v, and let $A = 0$:

$$\sum_{n=0}^{\infty} \frac{v^n}{n!}\,\tilde{Y}^n \sum_{m=0}^{\infty} \frac{x^m}{m!}\, A^m\Big|_{A=0} = \sum_{n=0}^{\infty} \frac{v^n}{n!}\, Y^n 1$$

and compare with equation (4).

The same idea works in several variables.
We have $\mathbf{V}(\mathbf{z}) = (V_1(z_1,\ldots,z_N),\ldots,V_N(z_1,\ldots,z_N))$ analytic in a neighborhood of 0 in $\mathbf{C}^N$. Denote the Jacobian matrix $\left(\dfrac{\partial V_i}{\partial z_j}\right)$ by V' and its inverse by W. The variables

$$Y_i = \sum_{k=1}^{N} x_k W_{ki}(D)$$

commute and act as raising operators for generating the basis $y_{\mathbf{n}}(\mathbf{x})$. I.e., $Y_i y_{\mathbf{n}} = y_{\mathbf{n}+\mathbf{e}_i}$. And $V_i = V_i(\mathbf{D})$, $\mathbf{D} = (D_1,\ldots,D_N)$, are lowering operators: $V_i y_{\mathbf{n}} = n_i\, y_{\mathbf{n}-\mathbf{e}_i}$.

Denote $\sum_i a_i b_i$ by $a\cdot b$. With variables A_i and corresponding partials ∂_i, define the vector fields

$$\tilde{Y}_i = \sum_k W_{ki}(A)\partial_k$$

For a vector field $\tilde{Y} = \sum_i W_i(A)\partial_i$, we have the identities

$$\tilde{Y}\, e^{A\cdot x} = x\cdot W(A)\, e^{A\cdot x} = x\cdot W(D)\, e^{A\cdot x}$$

The method of characteristics applies as in one variable and as in equation (3)

$$\exp(v\cdot Y)e^{A\cdot x} = e^{x\cdot U(v+V(A))}$$

Thus, we have the expansion

$$\exp\bigl(x\cdot U(v)\bigr) = \sum_{\mathbf{n}} \frac{\mathbf{v}^{\mathbf{n}}}{\mathbf{n}!}\, y_{\mathbf{n}}(\mathbf{x}) \tag{5}$$

In particular, the k^{th} component, U_k, of the inverse function is given by the coefficient of x_k in the above expansion.

An important feature of our approach is that to get an expansion to a given order requires knowledge of the expansion of W just to that order. The reason is that when iterating $xW(D)$, at step n it is acting on a polynomial of degree $n-1$, so all terms of the expansion of $W(D)$ of order n or higher would yield zero acting on y_{n-1}. This allows for streamlined computations.

For polynomial systems $\mathbf{V}$, V' will have polynomial entries, and W will be rational in $\mathbf{z}$. Hence raising operators will be rational functions of $\mathbf{D}$, linear in $\mathbf{x}$. Thus the coefficients of the expansion of the entries W_{ij} of W would be computed by finite-step recurrences.

Remark 1. Note that to solve $V(z) = v$ for z near z_0, with $V(z_0) = v_0$, apply the method to $V_1(z) = V(z + z_0) - v_0$, so that $V_1(0) = 0$. The inverse is $U_1(v) = U(v + v_0) - z_0$. Then $U(v) = z_0 + U_1(v - v_0)$.

2 One-variable Case

In this section we focus on the one-variable case. We illustrate the method with examples, then present an algorithm suitable for symbolic computation.

Example 1. In one variable, solving a cubic is interesting as the expansion of W can be expressed in terms of Chebyshev polynomials.

Let $V = z^3/3 - \alpha z^2 + z$. Then $V' = z^2 - 2\alpha z + 1$. Thus

$$W = \frac{1}{1 - 2\alpha z + z^2} = \sum_{n=0}^{\infty} z^n U_n(\alpha)$$

where U_n are Chebyshev polynomials of the second kind.

Specializing α provides interesting cases. For example, let $\alpha = \cos(\pi/4)$, or $V = z^3/3 - z^2/\sqrt{2} + z$. Then the coefficients in the expansion of W are periodic with period 8 and, in fact,

$$W = \frac{1 + z^2 + \sqrt{2}\,z}{1 + z^4}$$

The coefficient of x in the polynomials y_n yield the coefficients in the expansion of the inverse U. Here are some polynomials starting with $y_0 = 1$, $y_1 = x$:

$$y_2 = x^2 + x\sqrt{2}, \quad y_3 = x^3 + 3\,x^2\sqrt{2} + 4\,x,$$
$$y_4 = x^4 + 6\,x^3\sqrt{2} + 22\,x^2 + 10\,x\sqrt{2},$$
$$y_5 = x^5 + 10\,x^4\sqrt{2} + 70\,x^3 + 90\,x^2\sqrt{2} + 40\,x$$
$$y_6 = x^6 + 15\,x^5\sqrt{2} + 170\,x^4 + 420\,x^3\sqrt{2} + 700\,x^2 - 140\,x\sqrt{2}$$

This gives to order 6:

$$U(v) = \left(v + \frac{2}{3}\,v^3 + \frac{1}{3}\,v^5 + \dots\right) + \sqrt{2}\,\left(\frac{1}{2}\,v^2 + \frac{5}{12}\,v^4 - \frac{7}{36}\,v^6 + \dots\right)$$

This expansion will give approximate solutions to

$$z^3/3 - z^2/\sqrt{2} + z - v = 0$$

for v near 0.

Example 2. Inversion of the Chebyshev polynomial $T_3(z) = 4z^3 - 3z$ can be used as the basis for solving general cubic equations ([4]).

To get started we have, with $V(z) = 4z^3 - 3z$,

$$W(z) = \frac{-1}{3}\,\frac{1}{1-4z^2} = \frac{-1}{3}\sum_{n=0}^{\infty} 4^n z^{2n}$$

So $y_1 = (-1/3)x$, $y_2 = (1/9)x^2$, $y_3 = (-1/27)(x^3 + 8x)$, etc. We find

$$U(v) = -\frac{1}{3}\,v - \frac{4}{81}\,v^3 - \frac{16}{729}\,v^5 - \frac{256}{19683}\,v^7 - \cdots$$

In this case, we can find the expansion analytically. To solve $T_3(z) = v$, write

$$T_3(\cos\theta) = \cos(3\theta) = v$$

Invert to get, for integer k, $\theta = (1/3)(2\pi k \pm \arccos v)$, with arccos denoting the principal branch. Then

$$z = \cos((1/3)(2\pi k \pm \arccos v))$$

We want a branch with $v = 0$ corresponding to $z = 0$. With $\arccos 0 = \pi/2$, we want the argument of the cosine to be $\pi/2 + \pi l$, for some integer l. This yields the condition $\dfrac{1}{3} = \dfrac{2l+1}{4k \pm 1}$. Taking $l = 0$, we get $k = 1$, with the minus sign. I.e.,

$$U(v) = \cos((1/3)(2\pi - \arccos v))$$

Using hypergeometric functions (see next example) and rewriting, we find the form

$$U(v) = -\frac{1}{3}\sum_{n=0}^{\infty}\binom{3n}{n}\left(\frac{4}{27}\right)^n \frac{v^{2n+1}}{2n+1}$$

If we generate the polynomials y_n, we can find the expansion of $U(v)^m$ to any order.

Example 3. A similar approach is interesting for the Chebyshev polynomial $T_n(z)$.

$F(v) = \cos(\lambda(\mu \pm \arccos v))$ satisfies the hypergeometric differential equation

$$(1 - v^2)\, F'' - v\, F' + \lambda^2\, F = 0$$

which can be written in the form

$$[(vD_v)^2 - D_v^2]F = \lambda^2\, F$$

with here D_v denoting d/dv. For integer λ, this is the differential equation for the corresponding Chebyshev polynomial. In general, these are *Chebyshev functions*. As noted above, for $F(0) = 0$, we take $\mu = 2\pi k$, and, as above, we require

$$\lambda = \frac{2l+1}{4k \pm 1}$$

With $F'(0) = \pm\lambda$, we have the solution

$$F(v) = \pm\lambda v\, {}_2F_1\left(\left.\begin{matrix}\frac{1+\lambda}{2}, \frac{1-\lambda}{2} \\ \frac{3}{2}\end{matrix}\,\right|\, v^2\right)$$

2.1 Using Maple

For symbolic computation using Maple, one can use the Ore_Algebra package.

1. First fix the degree of approximation. Expand W as a polynomial to that degree.

2. Declare the Ore algebra with one variable, x, and one derivative, D.
3. Define the operator $xW(D)$ in the algebra.
4. Iterate starting with $y_0 = 1$ using the `applyopr` command.
5. Extract the coefficient of $x^m/m!$ to get the expansion of $U(v)^m$.

3 Algorithm as a matrix computation

Here is a matrix approach that can be implemented numerically.
Fix the order of approximation n.
Cut off the expansion

$$W(z) = w_0 + w_1 z + w_2 w^2 + \cdots + w_k z^k + \cdots$$

at $w_n z^n$.
Let the matrix

$$W = \begin{pmatrix} w_1 & w_0 & 0 & \dots & 0 \\ w_2 & w_1 & w_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n-1} & w_{n-2} & w_{n-3} & \dots & w_0 \\ w_n & w_{n-1} & w_{n-2} & \dots & w_1 \end{pmatrix}$$

Define the auxiliary diagonal matrices

$$P = \begin{pmatrix} 1! & 0 & \dots & 0 \\ 0 & 2! & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n! \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n \end{pmatrix}$$

$$Q = \begin{pmatrix} 1/\Gamma(1) & 0 & \dots & 0 \\ 0 & 1/\Gamma(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\Gamma(n) \end{pmatrix}$$

Note that $QP = M$.

Denoting $y_k(x) = \sum c_j^{(k)} x^j$, we have the recursion

$$[c_1^{(k+1)}, c_2^{(k+1)}, \ldots, c_n^{(k+1)}] = [c_1^{(k)}, c_2^{(k)}, \ldots, c_n^{(k)}] PWQ$$

The condition $U(0) = 0$ gives $y_0 = 1$. Then $y_1 = XW(D)y_0$ yields $y_1 = w_0 x$. We see that $c_0^{(k)} = 0$ for $k > 0$. We iterate as follows:
1. Start with w_0 times the unit vector $[1, 0, \ldots, 0]$ of length n.
2. Multiply by W.
3. Iterate, multiplying on the right by MW at each step.
4. Finally, multiply on the right by Q.

The top row will give the coefficients of the expansion of $U(v)$ to order n.

4 Higher-order Example

Here is a simple 2×2 system for illustration.

$$\begin{aligned} V_1 &= z_1 + z_2^2/2 \\ V_2 &= z_2 - z_1 z_2 \end{aligned}$$

So

$$V' = \begin{pmatrix} 1 & z_2 \\ -z_2 & 1 - z_1 \end{pmatrix} \qquad \text{and} \qquad W = \frac{1}{1 - z_1 + z_2^2} \begin{pmatrix} 1 - z_1 & -z_2 \\ z_2 & 1 \end{pmatrix}$$

The raising operators are

$$\begin{aligned} Y_1 &= \big(x_1(1 - D_1)) + x_2 D_2\big)\,(1 - D_1 + D_2^2)^{-1} \\ Y_2 &= (-x_1 D_2 + x_2)\,(1 - D_1 + D_2^2)^{-1} \end{aligned}$$

Expanding $(1 - D_1 + D_2^2)^{-1} = \sum_{n=0}^{\infty} (D_1 - D_2^2)^n$ yields, with $y_{00} = 1$,

$$\begin{gathered} y_{01} = x_2, \qquad y_{10} = x_1 \\ y_{02} = x_2^2 - x_1, \quad y_{11} = x_2 + x_1 x_2, \quad y_{20} = x_1^2 \end{gathered}$$

Thus

$$\begin{aligned}
&\exp\bigl(\mathbf{x}\cdot\mathbf{U}(\mathbf{v})\bigr)\\
&= 1 + x_1v_1 + x_2v_2\\
&+(x_2+x_1x_2)v_1v_2 + (x_2^2 - x_1)\,\frac{v_1^2}{2} + x_1^2\,\frac{v_2^2}{2} + \ldots
\end{aligned}$$

so

$$\begin{aligned}
U_1(\mathbf{v}) &= v_1 - v_1^2/2 + \ldots\\
U_2(\mathbf{v}) &= v_2 + v_1v_2 + \ldots
\end{aligned}$$

5 Another matrix approach

For any given order n, the polynomials of degree n are an invariant subspace for the operator Y up until the last step. We can formulate an alternative matrix computation as follows. Let $\bar{D}$ and $\bar{X}$ denote the matrices of the operators of differentiation and multiplication by x respectively on polynomials of degree less than or equal to n. The space is invariant under differentiation, and we cut off multiplication by x to be zero on x^n. We get

$$\bar{D}_{ij} = i\,\delta_{i+1,j}, \qquad \text{and} \qquad \bar{X}_{ij} = \delta_{i-1,j}$$

with the first row of $\bar{X}$ all zeros. We then compute the matrix $\bar{X}$ times $W(\bar{D})$, where $W(\bar{D})$ is computed as a matrix polynomial by substituting in $W(z)$ up to order n. Then Y has a matrix representation, $\bar{Y} = \bar{X}W(\bar{D})$, on the space and we iterate multiplying by $\bar{Y}$ acting on the unit vector $\mathbf{e_1}$. These give the coefficients of the polynomials y_n.

In several variables, one constructs matrices for D_j and X_i using Kronecker products of $\bar{D}$ and $\bar{X}$ with the identity. For example,

$$\bar{D}_j = I \otimes I \otimes \cdots \otimes \bar{D} \otimes I \cdots \otimes I$$

with $\bar{D}$ in the j^{th} spot. Similarly for $\bar{X}_i$. Then one has explicit matrix representations for the dual vector fields and the polynomials can be found accordingly.
This approach is explicit, but seems to much slower than using the built-in Ore_algebra package.

6 Worksheets

```
> with(linalg):
> with(Ore_algebra):
```

One Variable

```
>
> n:=10;
```

$$n := 10$$

```
> unassign('z','y','Y','x','V','W'):
> V:=z-z^2/2;
```

$$V := z - \frac{1}{2} z^2$$

```
> W:=diff(V,z)^(-1);
> W:=convert(taylor(W,z=0,n),polynom);
```

$$W := \frac{1}{1 - z}$$

$$W := 1 + z + z^2 + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9$$

```
> ############ ORE ALGEBRA STARTS HERE ####################
> A:=diff_algebra([z,x]);
```

$$A := Ore\_algebra$$

```
> YY:=x*W;
```

$$YY := x (1 + z + z^2 + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9)$$

```
> y[0]:=1:
> for i from 1 to n-1 do y[i]:=simplify(applyopr(YY,y[i-1],A)) od;
```

$$y_1 := x$$

$$y_2 := x^2 + x$$

$$y_3 := x^3 + 3 x^2 + 3 x$$

$$y_4 := x^4 + 6 x^3 + 15 x^2 + 15 x$$

$$y_5 := x^5 + 10 x^4 + 45 x^3 + 105 x^2 + 105 x$$

Several Variables

Example in two variables from an analytic function

```
> V1:=evalc(Re(expand(subs(z=zc,f))));V2:=evalc(Im(expand(subs(z=zc,f))));
```

$$V1 := z1 - 4\, z1^2 + 4\, z1\, z2 + 4\, z2^2$$

$$V2 := z2 - 2\, z1^2 - 8\, z1\, z2 + 2\, z2^2$$

```
> Jac:=jacobian([V1,V2],[z1,z2]):
> W:=evalm(Jac^(-1)):adj(Jac),factor(det(Jac));
> WMat:=map(mtaylor,W,[z1=0,z2=0],n);
```

$$\begin{bmatrix} 1 - 8\, z1 + 4\, z2 & -4\, z1 - 8\, z2 \\ 4\, z1 + 8\, z2 & 1 - 8\, z1 + 4\, z2 \end{bmatrix},\ 1 - 16\, z1 + 8\, z2 + 80\, z1^2 + 80\, z2^2$$

$$WMat := [[1 + 8\, z1 - 4\, z2 + 48\, z1^2 - 48\, z2^2 - 128\, z1\, z2 + 128\, z1^3 - 384\, z1\, z2^2 - 2112\, z2\, z1^2 + 704\, z2^3, -4\, z1 - 8\, z2 - 64\, z1^2 - 96\, z1\, z2 + 64\, z2^2 - 704\, z1^3 - 384\, z2\, z1^2 + 2112\, z1\, z2^2 + 128\, z2^3], [4\, z1 + 8\, z2 + 64\, z1^2 + 96\, z1\, z2 - 64\, z2^2 + 704\, z1^3 + 384\, z2\, z1^2 - 2112\, z1\, z2^2 - 128\, z2^3, 1 + 8\, z1 - 4\, z2 + 48\, z1^2 - 48\, z2^2 - 128\, z1\, z2 + 128\, z1^3 - 384\, z1\, z2^2 - 2112\, z2\, z1^2 + 704\, z2^3]]$$

```
> ############ ORE ALGEBRA STARTS HERE ####################
> A:=diff_algebra([z1,x1],[z2,x2]);
```

$$A := Ore\_algebra$$

```
> for ix to N do YY[ix]:=simplify(add(x||k*WMat[k,ix],k=1..N)) od;
```

$$YY_1 := x1 + 8\, x1\, z1 - 4\, x1\, z2 + 48\, x1\, z1^2 - 48\, x1\, z2^2 - 128\, x1\, z1\, z2 + 128\, x1\, z1^3 - 384\, x1\, z1\, z2^2 - 2112\, x1\, z2\, z1^2 + 704\, x1\, z2^3 + 4\, x2\, z1 + 8\, x2\, z2 + 64\, x2\, z1^2 + 96\, x2\, z1\, z2 - 64\, x2\, z2^2 + 704\, x2\, z1^3 + 384\, x2\, z2\, z1^2 - 2112\, x2\, z1\, z2^2 - 128\, x2\, z2^3$$

$$YY_2 := -4\, x1\, z1 - 8\, x1\, z2 - 64\, x1\, z1^2 - 96\, x1\, z1\, z2 + 64\, x1\, z2^2 - 704\, x1\, z1^3 - 384\, x1\, z2\, z1^2 + 2112\, x1\, z1\, z2^2 + 128\, x1\, z2^3 + x2 + 8\, x2\, z1 - 4\, x2\, z2 + 48\, x2\, z1^2 - 48\, x2\, z2^2 - 128\, x2\, z1\, z2 + 128\, x2\, z1^3 - 384\, x2\, z1\, z2^2 - 2112\, x2\, z2\, z1^2 + 704\, x2\, z2^3$$

```
> y[0,0]:=1:
> for i from 0 to n-1 do j:=0; if(i>0) then
  y[i,0]:=simplify(applyopr(YY[1],y[i-1,0],A)) fi; for j from 1 to n-1 do
  y[i,j]:=simplify(applyopr(YY[2],y[i,j-1],A));print("y["||i||","||j||"]=",
  [i,j]) od;  od;
```

$$j := 0$$

"y[0,1]=", $x2$

$$j := 0$$

$$\text{"y[0,1]=", } x2$$

$$\text{"y[0,2]=", } x2^2 - 4\,x2 - 8\,x1$$

$$\text{"y[0,3]=", } x2^3 - 12\,x2^2 - 24\,x2\,x1 + 192\,x1 - 144\,x2$$

$$j := 0$$

$$\text{"y[1,1]=", } x2\,x1 - 4\,x1 + 8\,x2$$

$$\text{"y[1,2]=", } x2^2\,x1 - 12\,x2\,x1 + 16\,x2^2 - 144\,x1 - 192\,x2 - 8\,x1^2$$

$$\text{"y[1,3]=", } 10560\,x1 - 1920\,x2 - 24\,x2^2\,x1 + 24\,x2^3 + x2^3\,x1 - 24\,x2\,x1^2 - 720\,x2\,x1 - 672\,x2^2 + 288\,x1^2$$

$$j := 0$$

$$\text{"y[2,1]=", } x2\,x1^2 + 24\,x2\,x1 + 4\,x2^2 - 8\,x1^2 - 192\,x1 + 144\,x2$$

$$\text{"y[2,2]=", } -1920\,x1 - 10560\,x2 + 40\,x2^2\,x1 - 8\,x1^3 + 4\,x2^3 + x2^2\,x1^2 - 20\,x2\,x1^2 - 960\,x2\,x1 + 400\,x2^2 - 320\,x1^2$$

$$\text{"y[2,3]=", } 497664\,x1 + 145152\,x2 + x2^3\,x1^2 - 24\,x2\,x1^3 - 2496\,x2^2\,x1 + 384\,x1^3 + 768\,x2^3 - 36\,x2^2\,x1^2 + 56\,x2^3\,x1 - 1392\,x2\,x1^2 + 4\,x2^4 - 13440\,x2\,x1 - 43200\,x2^2 + 30720\,x1^2$$

$$j := 0$$

$$\text{"y[3,1]=", } x2\,x1^3 + 48\,x2\,x1^2 + 12\,x2^2\,x1 + 720\,x2\,x1 + 288\,x2^2 - 12\,x1^3 - 672\,x1^2 - 10560\,x1 + 1920\,x2$$

$$\text{"y[3,2]=", } 145152\,x1 - 497664\,x2 - 28\,x2\,x1^3 + 1632\,x2^2\,x1 - 528\,x1^3 + 384\,x2^3 + 72\,x2^2\,x1^2 - 8\,x1^4 + 12\,x2^3\,x1 - 2496\,x2\,x1^2 + x2^2\,x1^3 - 73920\,x2\,x1 + 7680\,x2^2 - 5760\,x1^2$$

$$\text{"y[3,3]=", } 23553024\,x1 + 21829632\,x2 + 96\,x2^3\,x1^2 - 2160\,x2\,x1^3 - 233280\,x2^2\,x1 + 62400\,x1^3 + 19200\,x2^3 - 5760\,x2^2\,x1^2 + 480\,x1^4 - 24\,x2\,x1^4 + 2880\,x2^3\,x1 - 34560\,x2\,x1^2 - 48\,x2^2\,x1^3 + x2^3\,x1^3 + 480\,x2^4 + 1435392\,x2\,x1 + 12\,x2^4\,x1 - 2575872\,x2^2 + 2345472\,x1^2$$

>

References

1. P. Feinsilver and R. Schott. *Algebraic structures and operator calculus, Vols I–III.* Kluwer Academic Publishers, 1993, 1994, 1996.
2. P. Feinsilver and R. Schott. Vector fields and their duals. *Adv. in Math.*, 149:182–192, 2000.
3. G.-C. Rota, D. Kahaner, and A. Odlyzko. *Finite operator calculus.* Academic Press, 1975.
4. http://en.wikipedia.org/wiki/Cubic_equation.