Automatic Differentiation in MATLAB®

Kenneth C. Johnson
KJ Innovation
(Version 01-Oct-2025)
https://vixra.org/

Abstract

This document contains implementation notes for the MATLAB class "mpoly" (Multivariate Polynomial), which represents a numeric array (of any nonempty size, any number of dimensions) as a polynomial function (any degree) of a set of independent parameters (any number), or as a truncated Taylor series approximation. The class supports most standard array operations (algebra, indexing, etc.), employing automatic differentiation to calculate series coefficients of function outputs.

Representation

The MATLAB class mpoly[1] represents a multivariate polynomial function of parameters x_1, x_2, … , e.g.,

$$f(x) = C_1 + C_2 \cdot x_1 + C_3 \cdot x_1{}^2 + C_4 \cdot x_2 + C_5 \cdot x_2 \cdot x_1 + C_6 \cdot x_2{}^2 \tag{1}$$

For notational convenience, f will be represented as a homogeneous polynomial by defining an auxiliary constant parameter $x_0 = 1$. The polynomial is written with all powers multiplied out and with the same number of x factors in each monomial,

$$x_0 = 1 \tag{2}$$

$$f(x) = C_1 \cdot x_0 \cdot x_0 + C_2 \cdot x_1 \cdot x_0 + C_3 \cdot x_1 \cdot x_1 + C_4 \cdot x_2 \cdot x_0 + C_5 \cdot x_2 \cdot x_1 + C_6 \cdot x_2 \cdot x_2 \tag{3}$$

The x subscripts in the monomial associated with coefficient C_j are listed in a corresponding "subscript vector" $s_{j,:}$, and the subscript vectors are collected in a matrix s. For example, the subscript matrix associated with Eq. (3) is

[1] posted on MathWorks File Exchange: https://www.mathworks.com/matlabcentral/fileexchange/182185-class-mpoly-multivariate-polynomial

$$s=\begin{pmatrix}0&0\\1&0\\1&1\\2&0\\2&1\\2&2\end{pmatrix} \tag{4}$$

The general form of a multivariate polynomial is

$$f(x)=\sum_j C_j \cdot x_{s_{j,1}} \cdot x_{s_{j,2}} \cdots \cdot x_{s_{j,deg}} \tag{5}$$

deg is the polynomial degree, equal to the number of s columns. The subscript rows are individually sorted in descending order and are in the range $0,\ldots,Nx$, where Nx is the number of parameters (excluding x_0),

$$Nx \ge s_{j,1} \ge s_{j,2} \ldots \ge s_{j,deg} \ge 0 \tag{6}$$

The s rows are collectively sorted, first by column 1, then by column 2, etc.;

$$s_{j,1} \le s_{j+1,1}. \quad \text{If } s_{j,1:k-1} = s_{j+1,1:k-1}, \text{ then } s_{j,k} \le s_{j+1,k}. \tag{7}$$

(MATLAB's colon notation is used here and elsewhere to denote an index range, e.g., $1:k-1=[1,2,\ldots k-1]$.)

The monomial factor in Eq. (5) will be written as $mon(s_{j,:},x)$. For a particular subscript vector $s=[s_1,s_2,\ldots]$,

$$mon(s,x)=x_{s_1} \cdot x_{s_2} \ldots \tag{8}$$

Eq. (5) is restated as

$$f(x)=\sum_j C_j \cdot mon(s_{j,:},x) \tag{9}$$

(An isolated colon represents a full index range, e.g., ":" is shorthand for $1:deg$ in Eq. (9).)

Typically, the x_j elements in Eq. (8) are scalar, but they can alternatively be arrays to represent multi-valued parameters. In this case, the multiplication operations in Eq's. (8) and (9) are elementwise multiplications, which commute. (Following MATLAB's convention, product factors and summation terms are automatically repmat-expanded in singleton dimensions to match sizes.[2]) If $f(x)$ is array-valued for scalar parameters x_i, then the C_j coefficients can be

[2] See MATLAB documentation: Array vs. Matrix Operations and Compatible Array Sizes for Basic Operations.

array-valued. The x_i parameters can also be array-valued in this case, and the products in Eq's. (8) and (9) are still elementwise, commutative multiplications. In general, the x_i and C_j factors must all be mutually "size-compatible": Two arrays are size-compatible if they are size-matched in all dimensions except for singleton (size-1) dimensions in either array.

Index, subscript association

The complete subscript matrix s associated with polynomial degree deg and parameter count Nx is denoted as $s^{(deg,Nx)}$. The rows of $s^{(deg,Nx)}$ comprise all length-deg, descent-sorted integer vectors with elements in the range $0:Nx$ (Eq. (6)). The matrix has deg columns and is row-partitioned into two submatrices: a first submatrix containing no occurrences of Nx in any of its rows, and a second submatrix containing at least one occurrence of Nx in every row. The first submatrix is $s^{(deg,Nx-1)}$. All rows in the second submatrix have Nx as their first element (due to the descent ordering, Eq. (6)); thus the second submatrix, with its first column omitted, is $s^{(deg-1,Nx)}$. $s^{(deg,Nx)}$ is recursively constructed as follows:

$$\text{If } Nx=0 \text{ or } deg=0, \text{ then } s^{(deg,Nx)}=\overbrace{[0,0,\ldots]}^{\text{size-}[1,deg]}; \text{ otherwise}$$

$$s^{(deg,Nx)}=\left(\begin{array}{c|c} \multicolumn{2}{c}{s^{(deg,Nx-1)}} \\ \hline \vdots & \\ Nx & s^{(deg-1,Nx)} \\ \vdots & \end{array}\right) \tag{10}$$

The number of subscript rows in $s^{(deg,Nx)}$, denoted as $B(deg,Nx)$, is defined by the recursion relation

$$\begin{aligned} &\text{If } Nx=0 \text{ or } deg=0, \text{ then } B(deg,Nx)=1; \text{ otherwise} \\ &B(deg,Nx)=B(deg,Nx-1)+B(deg-1,Nx) \end{aligned} \tag{11}$$

This equates to the binomial coefficient

$$B(deg,Nx)=\frac{(Nx+1)\cdot(Nx+2)\cdot\ldots\cdot(Nx+deg)}{1\cdot 2\cdot\ldots\cdot deg} \tag{12}$$

Since $s^{(deg,Nx)}$ is constructed by appending rows to $s^{(deg,Nx-1)}$, it can be conceptualized as the first $B(deg,Nx)$ rows of an extended subscript matrix $s^{(deg)}$, which has an infinite number of rows:

$$s^{(deg,Nx)}=s^{(deg)}_{1:B(\deg,Nx),:} \quad \left(\text{i.e., } s^{(deg,Nx)}_{j,:}=s^{(deg)}_{j,:} \text{ for } j\le B(deg,Nx)\right) \tag{13}$$

$s^{(deg)}$ lists all length-deg, descent-sorted subscript vectors $s = s^{(deg)}_{j,:}$ as a function of serialization index j. The inverse mapping to index j from subscript vector s, denoted as $Index(s)$, is defined as

$$j = Index(s) = \texttt{size}\,(s^{(deg)}_{1:j,:}\texttt{,1}) \quad \text{with } deg = \texttt{length}(s) \text{ and } s^{(deg)}_{1:j,:} = s \tag{14}$$

In this definition s is a row vector, which is assumed to be zero-padded to length deg and descent-sorted. $s^{(deg)}_{1:j,:}$ is equivalent to $s^{(deg,Nx)}_{1:j,:}$ with $Nx = s^{(deg)}_{j,1}$ (because no $s^{(deg)}_{1:j,:}$ element exceeds $s^{(deg)}_{j,1}$); and based on the partitioning of $s^{(deg,Nx)}$ in Eq. (10), the following condition is obtained,

$$\begin{aligned}&\text{With } deg = \texttt{length}(s) \text{ and } Nx = s_1 \text{ (or } Nx = \ \text{ if } deg = 0\text{):}\\ &\text{If } deg = 0 \text{ or } Nx = 0, \text{ then } Index(s) = 1;\\ &\text{otherwise } Index(s) = B(deg, Nx-1) + Index(s_{2:deg}).\end{aligned} \tag{15}$$

It follows from Eq. (15) that

$$\begin{aligned}&\text{With } deg = \texttt{length}(s) \text{ if } deg > 0\text{:}\\ &Index(s) = B(deg, s_1 - 1) + B(deg-1, s_2 - 1) + \ldots + B(1, s_{deg} - 1) + 1\end{aligned} \tag{16}$$

(If $s_k = 0$, then the term $B(k, s_k - 1)$ in Eq. (16) is zero.)

Denoting the polynomial degree of $f(x)$ in Eq. (9) as $f.deg$, and the number of parameters $x_1, x_2, \ldots$ in $f(x)$ as $f.Nx$, the equation is written canonically as

$$f(x) = \sum_{j \in f.Indices} C_j \cdot mon(s^{(f.deg)}_{j,:}, x) \tag{17}$$

where the index set $f.Indices$ is selected from $1 : B(f.deg, f.Nx)$,

$$f.Indices \subseteq 1 : B(f.deg, f.Nx) \tag{18}$$

<u>Derivatives</u>

Each nonzero subscript $i = s^{(deg)}_{j,k}$ corresponds to a first-order partial derivative operator $\partial / \partial x_i$, and each subscript vector $i = s^{(deg)}_{j,:}$ corresponds to a generalized mixed partial derivative operator Dop_i

$$Dop_i = \prod_{\{k | i_k > 0\}} \frac{\partial}{\partial_{i_k}} \tag{19}$$

(The "Π" notation, in this context, represents operator composition.) The operator Dop_i, applied to function f defined by Eq. (17) and evaluated at $x = 0$, is denoted as D_j and is proportional to C_j,

$$\text{With } i = s_{j,:}^{(deg)}, \quad D_j = Dop_i f(0) = \sigma_j^{(deg)} \cdot C_j \tag{20}$$

The proportionality factor $\sigma_j^{(deg)}$ is a product of factorials,

$$\text{With } i = s_{j,:}^{(deg)}, \quad \sigma_j^{(deg)} = \prod_{k=1}^{\max(i)} \left(\left(\sum_{j=1}^{deg} i_j = k \right)! \right) \tag{21}$$

(In this expression the logical summand "$i_j = k$" is implicitly cast to an integer, 0 if false or 1 if true, and the sum counts the number of occurrences of k in i.)

<u>Polynomial evaluation</u>

Eq. (17) takes the following form, with application of Eq. (8),

$$f(x) = \sum_{j \in f.Indices} \left(\text{With } i = s_{j,:}^{(f.deg)}, \quad C_j \cdot x_{i_1} \cdot x_{i_2} \ldots \right) \tag{22}$$

The monomial $mon(i,x) = x_{i_1} \cdot x_{i_2} \ldots$ has monomial degree

$$mon\_deg(i) = \sum_k (i_k \neq 0) \tag{23}$$

(The logical summand "$i_k \neq 0$" is implicitly cast to an integer, 0 if false or 1 if ture.) Each subscript vector i is descent-sorted, so the first $mon\_deg(i)$ elements of i are nonzero and all remaining elements are zero. The product $mon(i,x) = x_{i_1} \cdot x_{i_2} \ldots$ need only include the first $mon\_deg(i)$ factors; all remaining factors are $x_0 = 1$ (Eq. (2)).

Eq. (22) can be efficiently evaluated by maintaining a list of partial products associated with each summation term,

$$pprod_{j,:} = \left(\text{With } deg = mon\_deg(s_{j,:}^{(f.deg)}), \quad [x_{i_1}, x_{i_1} \cdot x_{i_2}, \ldots, x_{i_1} \cdot x_{i_2} \cdot \ldots \cdot x_{i_{deg}}] \right) \tag{24}$$

Based on the construction illustrated in Eq. (10), the subscript vectors satisfy the relation

$$\text{With } deg = mon\_deg(s_{j+1,:}^{(f.deg)}), \quad deg > 0 \text{ and } s_{j+1,1:deg-1}^{(f.deg)} = s_{j,1:deg-1}^{(f.deg)} \tag{25}$$

($s_{j+1,:}^{(f.deg)}$ is generated from $s_{j,:}^{(f.deg)}$ by a counting procedure in which deg is the minimum index for which elements $s_{j,deg:f.deg}^{(f.deg)}$ are all identical, and $s_{j+1,:}^{(f.deg)} = [s_{j,1:deg-1}^{(f.deg)}, s_{j,deg}^{(f.deg)} + 1, 0, 0, \ldots]$.) The partial products can thus be calculated as follows,

$$\text{With } deg = mon\_deg(s_{j+1,:}^{(f.deg)}),$$
$$pprod_{j+1,:} = \begin{cases} s_{j+1,1}^{(f.deg)} & \text{if } deg = 1 \\ [pprod_{j,1:deg-1},\ pprod_{j,deg-1} \cdot s_{j+1,deg}^{(f.deg)}] & \text{if } deg > 1 \end{cases} \tag{26}$$

<u>Polynomial shift transformation</u>

A shift transformation of polynomial $f(x)$ by vector c computes the polynomial $g(x) = f(c+x)$. The shift is applied to one x coordinate at a time, i.e., a function sequence g_0, g_1, g_2, … with

$$g_0 = f,$$
$$g_{j+1}(x) = g_j(c' + x) \quad \text{with } c'_k = \begin{cases} c_k & \text{if } k = j+1 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Each step of this process involves a computation of the form $g(x) = f(c+x)$, where c is all-zero except for c_n:

$$c_k = 0 \quad \text{if } k \neq n \tag{28}$$

f and g have expansions having the form of Eq. (17),

$$g(x) = f(c+x) = \sum_{j \in f.Indices} C_j \cdot mon(s_{j,:}^{(f.deg)}, c+x) = \sum_{j \in f.Indices} C'_j \cdot mon(s_{j,:}^{(f.deg)}, x) \tag{29}$$

The C'_j coefficients are initialized to zero, and then terms $C_j \cdot {c_n}^q$ are accumulated into C'. The subscript vector $i = s_{j,:}^{(f.deg)}$ contains p occurrences of n in $i_{k:k+p-1}$, for some k and p,

$$\text{With } i = s_{j,:}^{(f.deg)}: \quad i_{1:k-1} > n, \quad i_{k:k+p-1} = n, \quad i_{k+p:f.deg} < n \quad (0 \le p \le f.deg) \tag{30}$$

The monomial term in the first sum in Eq. (29) expands to

$$\text{With } deg = mon\_deg(i), \quad mon(i, c+x) = x_{i_1} \cdot \ldots \cdot x_{i_{k-1}} \cdot (c_n + x_n)^p \cdot x_{i_{k+p}} \cdot \ldots \cdot x_{i,deg}$$
$$= \sum_{q=0}^{p} B(q, p-q) \cdot {c_n}^q \cdot x_{i_1} \cdot \ldots \cdot x_{i_{k-1}} \cdot {x_n}^{p-q} \cdot x_{i_{k+p}} \cdot \ldots \cdot x_{i,deg} \tag{31}$$
$$= \sum_{q=0}^{p} B(q, p-q) \cdot {c_n}^q \cdot mon([i_{[1:k-1,\, k+q:deg]}], x)$$

The factor $C_j \cdot B(q, p-q) \cdot {c_n}^q$ is accumulated into $C'_{j'}$ with $j' = Index([i_{[1:k-1,\, k+q:deg]}, 0, \ldots])$. (The $Index$ argument is zero-padded to length $f.deg$.)

The C' coefficients are calculated by the following algorithm,

$$
\begin{aligned}
&\text{Initialize } C'_j = 0. \\
&\text{For } j \in f.Indices, \text{ with } i = s^{(f.deg)}_{j,:} \\
&\quad \text{if } i \text{ contains } p \text{ occurences of } n \text{ with } i_{1:k-1} > n,\ i_{k:k+p-1} = n,\ i_{k+p:f.deg} < n\ \ (0 \le p \le f.deg), \\
&\qquad \text{for } q = 0 : p, \\
&\qquad\quad \text{with } deg = mon\_deg(i) \text{ and } j' = Index([i_{[1:k-1,\, k+q:deg]}, 0, \ldots]), \\
&\qquad\qquad C'_{j'} \leftarrow C'_{j'} + C_j \cdot B(q, p-q) \cdot {c_n}^q
\end{aligned}
\tag{32}
$$

<u>Polynomial product</u>

The following product algorithm applies to scalar multiplication of two polynomials but can be generalized in a straightforward way to elementwise array multiplication, matrix multiplication, tensor products, and paged variants[3]. To accommodate these generalizations, the product operation (which need not be commutative) will be denoted as "$*$", while elementwise multiplication is denoted as "$\cdot$".

Polynomial functions $f(x)$ and $g(x)$ having respective degrees $f.deg$ and $g.deg$, and parameter counts $f.Nx$ and $g.Nx$, are defined as in Eq. (9),

$$
f(x) = \sum_{j=1}^{B(f.deg,\, f.Nx)} fC_j \cdot mon(s^{(f.deg)}_{j,:}, x), \quad g(x) = \sum_{k=1}^{B(g.deg,\, g.Nx)} gC_k \cdot mon(s^{(g.deg)}_{k,:}, x) \tag{33}
$$

The product $h(x) = f(x) * g(x)$, with degree $h.deg$ and parameter count $h.Nx$, has a form similar to Eq's. (33),

$$
\begin{aligned}
h(x) &= \sum_{m=1}^{B(h.deg,\, h.Nx)} hC_m \cdot mon(s^{(h.deg)}_{m,:}, x) \\
&= f(x) * g(x) = \sum_{j=1}^{B(f.deg,\, f.Nx)} \sum_{k=1}^{B(g.deg,\, g.Nx)} fC_j * gC_k \cdot mon(s^{(f.deg)}_{j,:}, x) \cdot mon(s^{(g.deg)}_{k,:}, x), \\
h.deg &= f.deg + g.deg, \quad h.Nx = \max(f.Nx, g.Nx)
\end{aligned}
\tag{34}
$$

[3] pagemtimes, pagetensorprod

The subscript vector $s_{m,:}^{(h.deg)}$ in $h(x)$ is determined from associated vectors $s_{j,:}^{(f.deg)}$ and $s_{k,:}^{(g.deg)}$ via merging and sorting ($sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])$, where the "$\geq$" superscript denotes descent sorting, Eq. (6)):

$$\begin{aligned} &mon(s_{m,:}^{(f.deg+g.deg)}, x) = mon(s_{j,:}^{(f.deg)}, x) \cdot mon(s_{k,:}^{(g.deg)}, x), \\ &s_{m,:}^{(f.deg+g.deg)} = sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}]) \end{aligned} \tag{35}$$

(The monomial equivalence in Eq. (35) is a consequence of commutativity of the $\cdot$ operation.)

The mapping from j,k to m in Eq. (35) is defined by a "$sortIndex$" matrix,

$$m = sortIndex_{j,k}^{(f.deg,g.deg)} = Index(sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])) \tag{36}$$

(The $sortIndex^{(f.deg,g.deg)}$ matrix is considered to be of infinite size, but Eq. (34) only uses elements $sortIndex_{j,k}^{(f.deg,g.deg)}$ in the range $j \leq B(f.deg, f.Nx)$, $k \leq B(g.deg, g.Nx)$.) The coefficients hC_m in Eq. (34) are defined as follows,

$$\begin{aligned} &\text{With } m \in 1 : B(h.deg, h.Nx), \\ &\quad hC_m = \sum_{\left\{ j,k \,\middle|\, \substack{j \in 1:B(f.deg,f.Nx),\, k \in 1:B(g.deg,g.Nx), \\ \text{and } m = sortIndex_{j,k}^{(f.deg,g.deg)}} \right\}} fC_j * gC_k \end{aligned} \tag{37}$$

Eq. (37) is implemented procedurally as

$$\begin{aligned} &\text{With } m \in 1 : B(h.deg, h.Nx), \text{ initialize } hC_m = 0. \\ &\text{For } j \in 1 : B(f.deg, f.Nx) \text{ and } k \in 1 : B(g.deg, g.Nx), \\ &\quad \text{With } m = sortIndex_{j,k}^{(f.deg,g.deg)}, \quad hC_m \leftarrow hC_m + fC_j * gC_k \end{aligned} \tag{38}$$

<u>Degree-truncated polynomial approximations</u>

Infinite Taylor series are typically approximated as low-degree polynomials via degree truncation, e.g.,

$$f(x) = C_1 + C_2 \cdot x + C_3 \cdot x^2 + O\,x^3 \tag{39}$$

The polynomial product algorithm is modified to work with truncated polynomials. The degree of a polynomial product $h(x) = f(x) * g(x)$, without truncation, is the sum of the f and g factors' degrees, but if f or g is truncated then the degree of the product is typically the minimum of the factors' degrees, e.g.,

$$\left(C_1+C_2\cdot x+C_3\cdot x^2+O\,x^3\right)*\left(C_1'+C_2'\cdot x+C_3'\cdot x^2+C_4'\cdot x^3+O\,x^4\right)= \\ C_1*C_1'+\left(C_1*C_2'+C_2*C_1'\right)\cdot x+\left(C_1*C_3'+C_2*C_2'+C_3*C_1'\right)\cdot x^2+O\,x^3 \tag{40}$$

The product's degree can be higher if either of the factors is missing low-degree terms, e.g.,

$$\begin{aligned}&\left(C_1+C_2\cdot x+C_3\cdot x^2+O\,x^3\right)*\left(C_2'\cdot x+C_3'\cdot x^2+C_4'\cdot x^3+O\,x^4\right)=\\ &\quad C_1*C_2'\cdot x+\left(C_1*C_3'+C_2*C_2'\right)\cdot x^2+\left(C_1*C_4'+C_2*C_3'+C_3*C_2'\right)\cdot x^3+O\,x^4,\\ &\left(C_2\cdot x+C_3\cdot x^2+O\,x^3\right)*\left(C_3'\cdot x^2+C_4'\cdot x^3+O\,x^4\right)=\\ &\quad C_2*C_3'\cdot x^3+\left(C_2*C_4'+C_3*C_3'\right)\cdot x^4+O\,x^5\end{aligned} \tag{41}$$

A multivariate monomial $mon(s,x)$ associated with subscript vector s (Eq. (8)) has a monomial degree $mon\_deg(s)$ defined by Eq. (23). The degree of polynomial function $f(x)$, denoted as $f.deg$, is the maximum of its specified monomials' degrees. This can include monomials with zero-valued coefficients, but typically $f.deg$ is reduced to the maxim degree of the monomials with nonzero coefficients. The minimum degree of the monomials with nonzero coefficients are denoted as $f.min\_deg$. The truncation degree of a truncated polynomial $f(x)$, denoted as $f.trunc\_deg$, is the minimum degree of its truncated monomials. The conventional notation for degree truncation is $f(x)=\ldots+O|\,x\,|^{f.trunc\_deg}$. All monomials of degree $f.trunc\_deg$ and higher in $f(x)$ are truncated, and no monomials of lesser degree are truncated. Typically, $f.deg=f.trunc\_deg-1$; if $f.deg<f.trunc\_deg-1$, then all monomials of degree greater than $f.deg$ and less than $f.trunc\_deg$ implicitly have zero-valued coefficients.

If polynomial $f(x)$ is not truncated, then $f.trunc\_deg=\infty$. If the specified polynomial coefficients are all zero (including the constant term), then $f.min\_deg=f.trunc\_deg$ and $f.deg$ is undefined. (In general, $f.min\_deg$ represents the minimum monomial degree of any polynomial term that is, or could *potentially* be, nonzero.) The case $f.min\_deg=\infty$ holds when $f(x)$ is not truncated and is identically zero. With these conventions, the following conditions apply to a product of polynomials,

$$\begin{aligned}&\text{With } h(x)=f(x)*g(x):\\ &h.min\_deg=f.min\_deg+g.min\_deg\\ &h.trunc\_deg=min(f.trunc\_deg+g.min\_deg,\,g.trunc\_deg+f.min\_deg)\\ &h.deg=min(f.deg+g.deg,\,h.trunc\_deg-1)\\ &\quad(\text{or } h.deg \text{ is undefined if either } f.deg \text{ or } g.deg \text{ is undefined})\end{aligned} \tag{42}$$

The "degree span" of polynomial f, denoted as $f.span\_deg$, is defined as

$$f.span\_deg=f.trunc\_deg-f.min\_deg \tag{43}$$

(In the case that $f.min\_deg = f.trunc\_deg = \infty$, $f.span\_deg$ is defined to be zero.) It follows from Eq's. (42) that

$$\begin{aligned}&\text{With } h(x) = f(x) * g(x):\\ &h.span\_deg = min(f.span\_deg, g.span\_deg)\end{aligned} \tag{44}$$

The truncation degree of either f or g can be reduced so that $f.span\_deg = g.span\_deg$ without changing the product $f(x) * g(x)$. The discarded terms have no effect on the truncated product, so it can be assumed that

$$h.span\_deg = f.span\_deg = g.span\_deg \tag{45}$$

With truncation, the polynomial product formula, Eq. (38), is modified as follows: The monomial degree of subscript vector $s_{j,:}^{(deg)}$ (Eq. (23)) is denoted as $mon\_deg_j^{(deg)}$,

$$mon\_deg_j^{(deg)} = mon\_deg(s_{j,:}^{(deg)}) = \sum_{k=1}^{deg} (s_{j,k}^{(deg)} \neq 0) \leq deg \tag{46}$$

The following result follows from the condition $s_{m,:}^{(f.deg+g.deg)} = sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])$ in Eq. (35),

$$\begin{aligned}&sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}]) = s_{m,:}^{(f.deg+g.deg)} \;\rightarrow\\ &mon\_deg_j^{(f.deg)} + mon\_deg_k^{(g.deg)} = mon\_deg_m^{(f.deg+g.deg)} \leq f.deg + g.deg\end{aligned} \tag{47}$$

Without degree truncation, the product $h(x) = f(x) * g(x)$ has degree $h.deg = f.deg + g.deg$, but with degree truncation, $h.deg \leq f.deg + g.deg$ and the summation in Eq. (37) is truncated to only include product terms $fC_j \cdot gC_k$ with $mon\_deg_j^{(f.deg)} + mon\_deg_k^{(g.deg)} \leq h.deg$. With this restriction, the subscript vector $sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])$ in Eq. (47) comprises a length-$h.deg$ subscript vector $s_{m,:}^{(h.deg)}$ followed by $f.deg + g.deg - h.deg$ trailing zeros;

$$\begin{aligned}&sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}]) = [s_{m,:}^{(h.deg)}, \overbrace{0,0,\ldots}^{f.deg+g.deg-h.deg\text{ zeros}}] \rightarrow\\ &mon\_deg_j^{(f.deg)} + mon\_deg_k^{(g.deg)} = mon\_deg_m^{(h.deg)} \leq h.deg\end{aligned} \tag{48}$$

Eq's. (37) and (38) are modified as follows to accommodate truncation,

$$\begin{aligned}&\text{With } m \in 1:B(h.deg, h.Nx),\\ &\quad hC_m = \sum_{\left\{ j,k \;\middle|\; \begin{array}{l} j \in j \in 1:B(f.deg, f.Nx), k \in 1:B(g.deg, g.Nx),\\ mon\_deg_j^{(f.deg)} + mon\_deg_k^{(g.deg)} \leq h.deg,\\ \text{and } m = Index(sort^{(\geq)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])_{1:h.deg}) \end{array} \right\}} fC_j * gC_k\end{aligned} \tag{49}$$

$$\begin{aligned}
&\text{With } m \in 1:B(h.deg, h.Nx), \text{ initialize } hC_m = 0.\\
&\text{For } j \in 1:B(f.deg, f.Nx),\ k \in 1:B(g.deg, g.Nx),\\
&\quad \text{and } mon_deg_j^{(f.deg)} + mon_deg_k^{(g.deg)} \le h.deg:\\
&\qquad \text{With } m = Index(sort^{(\ge)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])_{1:h.deg}), \quad hC_m \leftarrow hC_m + fC_j * gC_k
\end{aligned} \tag{50}$$

Eq. (50) is efficiently implemented by precomputing the index triplets $[j,k,m]$, and the associated monomial degrees deg, and collecting them into rows of a four-column index matrix $timesIndex^{(f.deg, g.deg, h.deg, f.Nx, g.Nx)}$:

$$\begin{aligned}
&\text{With } [j,k,m,deg] = timesIndex_{i,:}^{(f.deg, g.deg, h.deg, f.Nx, g.Nx)}:\\
&\quad j \in 1:B(f.deg, f.Nx),\ k \in 1:B(g.deg, g.Nx),\\
&\quad m = Index(sort^{(\ge)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])_{1:h.deg}),\\
&\quad deg = mon_deg_m^{(h.deg)} \le h.deg
\end{aligned} \tag{51}$$

(The deg entry in $[j,k,m,deg]$ is not used here but will be used for polynomial division.) Eq. (50) translates to

$$\begin{aligned}
&\text{With } m \in 1:B(h.deg, h.Nx), \text{ initialize } hC_m = 0.\\
&\text{For } i = 1,2,\ldots\\
&\quad \text{With } [j,k,m] = timesIndex_{i,1:3}^{(f.deg, g.deg, h.deg, f.Nx, g.Nx)}, \quad hC_m \leftarrow hC_m + fC_j * gC_k
\end{aligned} \tag{52}$$

If the coefficients in either f or g are all zero, then the following zero-product formulas apply:

$$f(x) \cdot Ox^n = Ox^{f.min\_deg+n} \tag{53}$$

$$Ox^m \cdot Ox^n = Ox^{m+n} \tag{54}$$

Polynomial division

The following division algorithm applies to scalar division but can be generalized in a straightforward way to elementwise array division and matrix left- or right-division (with a square, nonsingular divisor matrix).

The operation $g(x) = f(x) \backslash h(x)$ (left-division) yields the solution $g(x)$ to the relation $h(x) = f(x) * g(x)$. A truncated polynomial approximation to $g(x)$ can be determined by modifying Eq. (52) to solve for the coefficients gC_k. The denominator $f(x)$ must have a nonzero leading coefficient fC_1 (i.e., $f.min\_deg = 0$). Assuming that $f(x)$ is non-constant, at least one of $f.trunc\_deg$ or $h.trunc\_deg$ must be finite. The numerator $h(x)$ is assumed to be nonzero. Also, it is assumed that

$$f.span\_deg = h.span\_deg \tag{55}$$

$$h.trunc\_deg = h.deg + 1 \tag{56}$$

$$h.Nx \ge f.Nx \tag{57}$$

Eq. (55) is consistent with Eq. (45). (Either $f.trunc\_deg$ or $h.trunc\_deg$ can be reduced, if necessary, to satisfy Eq. (55).) Eq. (56), with the additional condition $g.deg = h.deg$, is consistent with the relation $h.deg = min(f.deg + g.deg, h.trunc\_deg - 1)$ (Eq. (42)). ($h.deg$ can be increased to satisfy Eq. (56).) Eq. (57), with the additional condition $h.Nx = g.Nx$, is consistent with the relation $h.Nx = \max(f.Nx, g.Nx)$ (Eq. (34)). ($h.Nx$ can be increased to satisfy Eq. (57).) The preceding conditions imply

$$f.min\_deg = 0, \quad h.deg = h.min\_deg + f.deg \tag{58}$$

A polynomial approximation to $g(x)$ will be generated with

$$\begin{aligned} &g.min\_deg = h.min\_deg, \quad g.deg = h.deg, \quad g.trunc\_deg = h.trunc\_deg, \\ &g.Nx = h.Nx \end{aligned} \tag{59}$$

Eq. (52) computes a sum of the form

$$\begin{aligned} &\text{With } m \in 1 : B(h.deg, h.Nx) \text{ and } mon\_deg_m^{(h.deg)} \ge h.min\_deg, \\ &hC_m = \sum_{\left\{ i \,\middle|\, timesIndex_{i,3}^{(f.deg, g.deg, h.deg, f.Nx, g.Nx)} = m \right\}} \left(\begin{aligned} &\text{With } [j,k] = \\ &\quad timesIndex_{i,1:2}^{(f.deg, g.deg, h.deg, f.Nx, g.Nx)}, \\ &\quad fC_j * gC_k \end{aligned} \right) \end{aligned} \tag{60}$$

The sum includes a term with $j = 1$ because fC_1 is nonzero. For this case $s_{j,:}^{(f.deg)}$ is all zeros and $s_{m,:}^{(h.deg)} = s_{k,:}^{(g.deg)}$ (because $g.deg = h.deg$ and $s_{m,:}^{(h.deg)} = sort^{(\ge)}([s_{j,:}^{(f.deg)}, s_{k,:}^{(g.deg)}])_{1:h.deg}$), implying that $k = m$. Eq. (60) is applied in order of increasing degree $mon\_deg_m^{(h.\mathrm{deg})}$, with the $fC_1 * gC_m$ term separated out of the sum:

For $deg = h.min\_deg : h.deg$,
with $m \in 1 : B(h.deg, h.Nx)$ and $mon\_deg_m^{(h.\deg)} = deg$,

$$hC_m = fC_1 * gC_m + \sum_{\left\{ i \,\middle|\, \begin{array}{l} timesIndex_{i,3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = m \\ \text{and with } timesIndex_{i,1}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} > 1 \end{array} \right\}} \left(\begin{array}{l} \text{With } [j,k] = \\ \quad timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\ \quad fC_j * gC_k \end{array} \right) \tag{61}$$

For each $[j,k]$ index pair in the sum, $mon\_deg_j^{(f.\deg)} + mon\_deg_k^{(g.\deg)} = mon\_deg_m^{(h.\deg)} = deg$ and $mon\_deg_j^{(f.\deg)} > 0$ (because $j > 1$); hence, $mon\_deg_k^{(g.\deg)} < deg$ and the gC_k factors in the sum will have already been determined for a smaller deg. Thus, Eq. (61) can be solved for gC_m,

For $deg = h.min\_deg : h.deg$,
with $m \in 1 : B(h.deg, h.Nx)$ and $mon\_deg_m^{(h.\deg)} = deg$,

$$gC_m = fC_1 \backslash \left(\begin{array}{l} hC_m - \\ \displaystyle\sum_{\left\{ i \,\middle|\, \begin{array}{l} timesIndex_{i,3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = m \\ \text{and with } timesIndex_{i,1}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} > 1 \end{array} \right\}} \left(\begin{array}{l} \text{With } [j,k] = \\ \quad timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\ \quad fC_j * gC_k \end{array} \right) \end{array} \right) \tag{62}$$

Eq. (62) is formulated procedurally as follows,

Initialize $gC = hC$.
For $deg = h.min\_deg : h.deg$,
For $i = 1, 2, \ldots$: If $timesIndex_{i,4}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = deg$,
with $[j,k,m] = timesIndex_{i,1:3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}$, (63)
if $j > 1$, $gC_m \leftarrow gC_m - fC_j * gC_k$
For $m = 1 : B(h.deg, h.Nx)$,
if $mon\_deg_m^{(h.\deg)} = deg$, $gC_m \leftarrow fC_1 \backslash gC_m$

Polynomial square root

A truncated polynomial approximation to $f(x) = \sqrt{h(x)}$ can be determined by modifying Eq. (52), with $gC = fC$, to solve for the coefficients fC_j. Assuming that $h(x)$ is non-constant, it must be truncated to finite degree ($h.trunc\_deg$ finite), and its leading coefficient hC_1 must be nonzero (implying $h.min\_deg = 0$). A polynomial approximation to $f(x)$ will be generated with

$$f.min\_deg = h.min\_deg = 0, \quad f.trunc\_deg = h.trunc\_deg \tag{64}$$

$f(x)$ is the solution of $f(x) \cdot f(x) = h(x)$, using pointwise multiplication. The square root algorithm could be generalized to solve the equation $f(x) * f(x) = h(x)$, e.g., using matrix multiplication, but the following algorithm is applied using pointwise multiplication.

Eq. (61) is modified as follows for this case: gC is replaced by fC, the multiplication operator "$*$" is pointwise multiplication "$\cdot$", and the summation terms $[j,k] = [m,1]$ and $[j,k] = [1,m]$ are both separated out of the sum,

$$\begin{aligned}
& fC_1 = \sqrt{hC_1}, \\
& \text{For } deg = 1 : h.deg, \\
& \quad \text{with } m \in 1 : B(h.deg, h.Nx) \text{ and } mon_deg_m^{(h.\mathrm{deg})} = deg, \\
& \quad\quad hC_m = 2 \cdot fC_1 \cdot fC_m + \\
& \quad\quad \sum_{\left\{ i \,\middle|\, \begin{array}{l} timesIndex_{i,3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = m \\ \text{and with } timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} > 1 \end{array} \right\}} \left(\begin{array}{l} \text{With } [j,k] = \\ \quad timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\ \quad fC_j \cdot fC_k \end{array} \right)
\end{aligned} \tag{65}$$

This is solved for fC_m,

$$\begin{aligned}
& fC_1 = \sqrt{hC_1}, \\
& \text{For } deg = 1 : h.deg, \\
& \quad \text{with } m \in 1 : B(h.deg, h.Nx) \text{ and } mon_deg_m^{(h.\mathrm{deg})} = deg, \\
& \quad\quad fC_m = \\
& \quad\quad (2 \cdot fC_1) \backslash \left(\begin{array}{l} hC_m - \\ \displaystyle\sum_{\left\{ i \,\middle|\, \begin{array}{l} timesIndex_{i,3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = m \\ \text{and with } timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} > 1 \end{array} \right\}} \left(\begin{array}{l} \text{With } [j,k] = \\ \quad timesIndex_{i,1:2}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\ \quad fC_j \cdot fC_k \end{array} \right) \end{array} \right)
\end{aligned} \tag{66}$$

(Note: For the more general case $f(x) * f(x) = h(x)$ with a non-commutative multiplication operator, Eq. (65) would take the form of a Sylvester equation $hC_m = fC_1 * fC_m + fC_m * fC_1 + \ldots$, which can be solved for fC_m in the fC_1 diagonal space.)

Eq. (66) is implemented procedurally as in Eq. (63),

$$
\begin{aligned}
&\text{Initialize } fC = hC,\;\; fC_1 = \sqrt{hC_1}\,. \\
&\text{For } deg = 1 : h.deg, \\
&\quad \text{For } i = 1,2,\ldots:\; \text{If } timesIndex_{i,4}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = deg, \\
&\qquad \text{with } [j,k,m] = timesIndex_{i,1:3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\
&\qquad\quad \text{if } j > 1 \text{ and } k > 1,\;\; fC_m \leftarrow fC_m - fC_j \cdot fC_k \\
&\quad \text{For } m = 1 : B(h.deg, h.Nx), \\
&\qquad \text{if } mon\_deg_m^{(h.\mathrm{deg})} = deg,\;\; fC_m \leftarrow (2 \cdot fC_1) \setminus fC_m
\end{aligned}
\tag{67}
$$

The duplicate operations $fC_j \cdot fC_k$ and $fC_k \cdot fC_j$ can be avoided with the following modified procedure in which the condition $k > 1$ is replaced by $k \ge j$ and the product $fC_j \cdot fC_k$ is doubled in the case $k > j$,

$$
\begin{aligned}
&\text{Initialize } fC = hC,\;\; fC_1 = \sqrt{hC_1}\,. \\
&\text{For } deg = 1 : h.deg, \\
&\quad \text{For } i = 1,2,\ldots:\; \text{If } timesIndex_{i,4}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)} = deg, \\
&\qquad \text{with } [j,k,m] = timesIndex_{i,1:3}^{(f.deg,g.deg,h.deg,f.Nx,g.Nx)}, \\
&\qquad\quad \text{if } j > 1 \text{ and } k \ge j,\;\; fC_m \leftarrow fC_m - fC_j \cdot fC_k \cdot (1 + (k > j)) \\
&\quad \text{For } m = 1 : B(h.deg, h.Nx), \\
&\qquad \text{if } mon\_deg_m^{(h.\mathrm{deg})} = deg,\;\; fC_m \leftarrow (2 \cdot fC_1) \setminus fC_m
\end{aligned}
\tag{68}
$$

(The logical expression “ $k > j$ ” is implicitly cast to an integer, 0 if false or 1 if ture.)

<u>Multivariate composition</u>

The composition (or “chaining”) of multivariate polynomial f with polynomials g_1, g_2, … calculates

$$h(x) = f(g(x)) \tag{69}$$

f and h are assumed here to be scalars while g and x are vectors of length Ng and Nx, respectively,

$$Ng = f.Nx, \quad Nx = g_1.Nx = g_2.Nx = \ldots \tag{70}$$

The polynomials’ leading constant terms are separated out, leaving residual primed terms:

$$h'(x) = f'(g'(x)) \tag{71}$$

where

$$\begin{aligned} &g'(x) = g(x) - g(0), \quad g'(0) = 0 \\ &f'(y) = f(g(0) + y) - f(g(0)), \quad f'(0) = 0 \\ &h'(x) = h(x) - h(0), \quad h'(0) = 0 \end{aligned} \tag{72}$$

If f is truncated ($f.trunc\_deg \neq \infty$), then $g(0)$ must be zero. Otherwise, a shift transformation is applied to f to obtain f'. The primes on f', g', and h' are henceforth omitted and it is assumed that

$$g(0) = 0, \quad f(0) = 0, \quad h(0) = 0 \tag{73}$$

The polynomial degree of f is denoted as J,

$$J = f.deg \tag{74}$$

$h(x)$ is constructed by substituting $g_i(x)$ for y_i in $f(y)$,

$$\begin{aligned} f(y) &= \sum_n fC_n \cdot mon(s_{n,:}^{(J)}, y) + O\, y^{f.trunc\_deg} \\ &= \sum_n \Big(\text{With } i = s_{n,:}^{(J)} \text{ and } y_0 = 1,\ fC_n \cdot y_{i_1} \cdot y_{i_2} \cdot \ldots \cdot y_{i_J} \Big) + O\, y^{f.trunc\_deg} \\ h(x) &= f(g(x)) \\ &= \sum_n \Big(\text{With } i = s_{n,:}^{(J)} \text{ and } g_0(x) = 1,\ fC_n \cdot g_{i_1}(x) \cdot g_{i_2}(x) \cdot \ldots \cdot g_{i_J}(x) \Big) + O\, x^{h.trunc\_deg} \end{aligned} \tag{75}$$

f has minimum, maximum, and truncation degrees $f.min\_deg$, $f.deg$, and $f.trunc\_deg$. g_i has minimum, maximum, and truncation degrees $g_i.min\_deg$, $g_i.deg$, and $g_i.trunc\_deg$. Eq's. (73) imply that all of the min_deg values are positive,

$$f.min\_deg > 0, \quad g_i.min\_deg > 0 \ \ (i > 0) \tag{76}$$

The summation index n ranges over a set $f.Indices$ corresponding to nonzero coefficients fC_n,

$$n \in f.Indices \subseteq 1 : B(J, Ng) \tag{77}$$

(Each vector $s_{n,:}^{(J)}$ contains at least one nonzero element because $f.min\_deg > 0$.) In addition, if any $g_i(x)$ is identically zero $(g_i.min\_deg = \infty)$, then $f.Indices$ excludes all indices n for which subscript vector $s_{n,:}^{(J)}$ contains i. After making these adjustments, none of the summation terms in Eq. (75) will be identically zero.

Summation term n in Eq. (75) has minimum degree

$$min\_deg\_term_n = \left(\text{With } i = s^{(J)}_{n,:},\ \sum_{j \in 1:J} g_{i_j}.min\_deg \right) > 0 \tag{78}$$

The term's truncation degree $trunc\_deg\_term_n$ is determined by the factors' truncation degrees $g_{i_j}.trunc\_deg$. For example, the first factor $g_{i_1}(x)$ includes truncated terms $O|x|^{g_{i_1}.trunc\_deg}$ while the remaining factors $g_{i_2}(x) \cdot g_{i_3}(x) \cdot \ldots$ have combined minimum degree $g_{i_2}.min\_deg + g_{i_3}.min\_deg + \ldots$; thus, $trunc\_deg\_term_n$ is at most $g_{i_1}.trunc\_deg + g_{i_2}.min\_deg + g_{i_3}.min\_deg + \ldots$, or equivalently, $g_{i_1}.span\_deg + g_{i_1}.min\_deg + g_{i_2}.min\_deg + g_{i_3}.min\_deg + \ldots$. Considering similar truncation degrees of g_{i_2}, g_{i_3}, …, $trunc\_deg\_term_n$ is

$$trunc\_deg\_term_n = span\_deg\_term_n + min\_deg\_term_n > 0 \tag{79}$$

where

$$span\_deg\_term_n = \left(\text{With } i = s^{(J)}_{n,:},\ \min_{j \in 1:J}(g_{i_j}.span\_deg) \right) \tag{80}$$

(Note: With $g_0(x) = 1$, $g_0.min\_deg = 0$ in Eq. (78) and $g_0.span\_deg = \infty$ in Eq. (80).)

The h truncation degree is limited by $trunc\_deg\_term_n$, and it is also limited by the truncation terms $O|y|^{f.trunc\_deg}$ in $f(y)$. With the substitution $y = g(x)$, the truncated terms in $O|y|^{f.trunc\_deg}$ comprise products of $g_{i_1}(x)$, $g_{i_2}(x)$, … with at least $f.trunc\_deg$ factors (excluding $g_0(x) = 1$) in each term. The minimum degree of the truncated terms is $f.trunc\_deg \cdot \min_{i>0}(g_i.min\_deg)$. Hence, the truncation degree of h is limited by

$$h.trunc\_deg \le \min\left(\min_{n \in f.Indices}(trunc\_deg\_term_n), f.trunc\_deg \cdot \min_{i>0}(g_i.min\_deg) \right) \tag{81}$$

By default, $h.trunc\_deg$ is equal to the right side of this expression, but $h.trunc\_deg$ can be further reduced to a predetermined truncation limit. The minimum degree and degree span of $h(x)$ are

$$h.min\_deg = \min\left(\min_{n \in f.Indices}(min\_deg\_term_n),\ h.trunc\_deg \right) > 0 \tag{82}$$

$$h.span\_deg = h.trunc\_deg - h.min\_deg \ge 0 \tag{83}$$

If $h.span\_deg = 0$, then $h(x) = O|x|^{h.trunc\_deg}$. Assuming that $h.span\_deg > 0$, the degree of summation term n in Eq. (75) (deg_term_n) and of h $(h.deg)$ are

$$deg\_term_n = \min\left(\left(\text{With } i = s_{n,:}^{(J)},\ \sum_{j=1}^{J} g_{i_j}.deg \right), trunc\_deg\_term_n - 1 \right) \tag{84}$$

$$h.deg = \min\left(\max_{n \in f.Indices} (deg\_term_n), h.trunc\_deg - 1 \right) \tag{85}$$

If any $g_{i_j}.deg$ is undefined in Eq. (84) (i.e., $g_{i_j}(x) = O x^{g_{i_j}.trunc\_deg}$), then deg_term_n is undefined, and the set $f.Indices$ in Eq. (85) will be restricted to exclude n values for which deg_term_n is undefined. $g_{i_j}.span\_deg$ and $span\_deg\_term_n$ in Eq. (80) are then positive,

$$\text{For } n \in f.Indices \text{ and } i = s_{n,:}^{(J)},\ g_{i_j}.span\_deg > 0 \text{ and } span\_deg\_term_n > 0. \tag{86}$$

deg_term_n is defined for at least one n because the condition $h.span\_deg > 0$ and Eq's. (82) and (83) imply that $h.trunc\_deg > h.min\_deg = \min\left(\min_{n \in f.Indices} (min\_deg\_term_n), h.trunc\_deg \right)$; hence

$$min\_deg\_term_n < h.trunc\_deg \quad \text{for some } n \in f.Indices. \tag{87}$$

Eq. (81) implies

$$h.trunc\_deg \le trunc\_deg\_term_n \quad \text{for any } n \in f.Indices. \tag{88}$$

Hence $min\_deg\_term_n < trunc\_deg\_term_n$, $span\_deg\_term_n > 0$ (Eq. (79)), and deg_term_n is defined for some $n \in f.Indices$.

If any $h(x)$ summation term in Eq. (75) has truncation degree $trunc\_deg\_term_n$ greater than $h.trunc\_deg$, it can be truncated to degree $h.trunc\_deg$ without affecting the result. If $min\_deg\_term_n >= h.trunc\_deg$, then n can be omitted from $f.Indices$. Otherwise, the degree span of $g_{i_j}(x)$ can be limited to $h.trunc\_deg - min\_deg\_term_n$ in the context of term n (cf. Eq. (80)), i.e., the truncation degree $g_{i_j}.trunc\_deg$ is temporarily (in the context of term n) limited to $g_{i_j}.trunc\_deg \le g_{i_j}.min\_deg + h.trunc\_deg - min\_deg\_term_n$.

The polynomial degrees of g_i ($i \in 1:Ng$) and h are denoted as K_i and M,

$$K_i = g_i.deg \text{ (or 0 if } g_i.deg \text{ is undefined)}, \quad M = h.deg \tag{89}$$

$g_i(x)$ has the polynomial expansion

$$g_i(x) = \sum_{k \in 1:B(K_i, Nx)} gC_{i,k} \cdot mon(s_{k,:}^{(K_i)}, x) + O|x|^{g_i.trunc\_deg} \tag{90}$$

This is substituted in Eq. (75) (initially with n ranging over the full index range $1:B(J,Ng)$),

$$h(x)=\sum_{n\in 1:B(J,Ng)}\left(\begin{array}{l}\text{With } i=s_{n,:}^{(J)} \text{ and } gC_{0,k_j}=(k_j=1),\\ \displaystyle\sum_{\substack{k_1\in 1:B(K_{i_1},Nx)\\ k_2\in 1:B(K_{i_2},Nx)\\ \dots\\ k_J\in 1:B(K_{i_J},Nx)\\ \sum_{j=1}^{J} mon\_deg([s_{k_j,:}^{(K_{i_j})}])\le M}}\left(\begin{array}{l} fC_n\cdot gC_{i_1,k_1}\cdot gC_{i_2,k_2}\cdot\ldots\cdot gC_{i_J,k_J}\\ \quad\cdot mon(s_{k_1,:}^{(K_{i_1})},x)\cdot mon(s_{k_2,:}^{(K_{i_2})},x)\cdot\ldots\cdot mon(s_{k_J,:}^{(K_{i_J})},x)\end{array}\right)\end{array}\right)$$
$$+O\,x^{h.trunc\_deg} \tag{91}$$
$$=\left(\sum_{m\in 1:B(M,Nx)} hC_m\cdot mon(s_{m,:}^{(M)},x)\right)+O\,x^{h.trunc\_deg}$$

(The logical expression "$k_j=1$" is implicitly cast to an integer, 0 if false or 1 if true.)

Subscript vectors $s_{k_1,:}^{(K_{i_1})}$, $s_{k_2,:}^{(K_{i_2})}$, … in Eq. (91) are merged and sorted to determine a corresponding subscript vector $s_{m,:}^{(M)}$ in $h(x)$,

$$\begin{array}{l} mon(s_{k_1,:}^{(K_{i_1})},x)\cdot mon(s_{k_2,:}^{(K_{i_2})},x)\cdot\ldots\cdot mon(s_{k_J,:}^{(K_{i_J})},x)=mon([s_{k_1,:}^{(K_{i_1})},s_{k_2,:}^{(K_{i_2})},\ldots,s_{k_J,:}^{(K_{i_J})}],x)\\ \quad =mon(sort^{(\ge)}([s_{k_1,:}^{(K_{i_1})},s_{k_2,:}^{(K_{i_2})},\ldots,s_{k_J,:}^{(K_{i_J})}]),x)=mon(s_{m,:}^{(M)},x);\\ s_{m,:}^{(M)}=sort^{(\ge)}([s_{k_1,:}^{(K_{i_1})},s_{k_2,:}^{(K_{i_2})},\ldots,s_{k_J,:}^{(K_{i_J})}])_{1:M}\end{array} \tag{92}$$

Eq. (91) is implemented by precomputing the index combinations $[m,n,i_1,k_1,i_2,k_2\ldots,i_J,k_J]$ and collecting them into a matrix $chainIndex^{(Nx,M,K_1,K_2,\ldots,K_J)}$:

$$\begin{array}{l} \text{With } [m,n,i_1,k_1,i_2,k_2\ldots,i_J,k_J]=chainIndex_{p,:}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}:\\ n\in 1:B(J,Ng)\quad (fC_n \text{ index})\\ i_j=s_{n,j}^{(J)},\quad 0<i_1\ge i_2\ge\ldots\ge i_J\ge 0\quad \left(gC_{i_j,k_j}\text{ 1st subscript}\right)\\ k_j\in 1:B(K_{i_j},Nx)\quad \left(gC_{i_j,k_j}\text{ 2nd subscript}\right)\quad \text{If } i_j>0, \text{ then } k_j>1; \text{ if } i_j=0, \text{ then } k_j=1.\\ mon\_deg([s_{k_1,:}^{(K_{i_1})},s_{k_2,:}^{(K_{i_2})},\ldots,s_{k_J,:}^{(K_{i_J})}])\le M\quad \left(mon(s_{m,:}^{(M)},x)\right)\\ s_{m,:}^{(M)}=sort^{(\ge)}([s_{k_1,:}^{(K_{i_1})},s_{k_2,:}^{(K_{i_2})},\ldots,s_{k_J,:}^{(K_{i_J})}])_{1:M}\end{array} \tag{93}$$

The condition $0<i_1$ is imposed because $0=i_1$ implies $i_1=i_2=\ldots=i_J=0$ and $n=1$ ($i=s_{n,:}^{(J)}$, all-zero); but $fC_1=0$ in Eq. (75) because $f(0)=0$ (Eq. (73)). If $i_j>0$, then $k_j>1$ because $g_i(0)=0$ for $i>0$ (Eq's. (73), (90)). If $i_j=0$, then $k_j=1$ because $g_0(x)=1$. $chainIndex^{(Nx,M,K_1,K_2,\ldots,K_J)}$ is defined under the premise that $g_1.Nx=g_2.Nx=\ldots=Nx$, but if any

$g_i.Nx < Nx$, then $chainIndex^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}$ rows can be omitted to include only $k_j \in 1:B(K_{i_j}, g_{i_j}.Nx)$.

Eq. (91) is reformulated as a sum over $chainIndex^{(Nx,M,K_1,K_2,\ldots,K_J)}$ rows,

$$\begin{aligned} h(x) &= \sum_p \left(\begin{array}{l} \text{With } [m,n,i_1,k_1,i_2,k_2\ldots,i_J,k_J] = chainIndex_{p,:}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}, \\ fC_n \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_J,k_J} \cdot mon(s_{m,:}^{(M)}, x) \end{array} \right) \\ &\quad + O\, x^{h.trunc\_deg} \\ &= \left(\sum_{m \in 1:B(M,Nx)} hC_m \cdot mon(s_{m,:}^{(M)}, x) \right) + O\, x^{h.trunc\_deg} \end{aligned} \tag{94}$$

(This sum can include zero factors fC_n or gC_{i_j,k_j}. $chainIndex^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}$ rows corresponding to zero-valued summation terms can be eliminated.) Corresponding monomials on both sides of Eq. (94) are matched to obtain hC_m,

$$\begin{array}{l} \text{With } m \in chainIndex_{:,1}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}, \\ hC_m = \displaystyle\sum_{\left\{ p \,\middle|\, chainIndex_{p,1}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})} = m \right\}} \left(\begin{array}{l} \text{With } [n,i_1,k_1,i_2,k_2\ldots,i_J,k_J] = \\ \quad chainIndex_{p,2:end}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}, \\ fC_n \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_J,k_J} \end{array} \right) \end{array} \tag{95}$$

Eq. (95) is implemented procedurally as

$$\begin{array}{l} \text{With } m \in 1:B(M,Nx), \text{ initialize } hC_m = 0. \\ \text{For } p = 1,2,\ldots \\ \quad \text{With } [m,n,i_1,k_1,i_2,k_2\ldots,i_J,k_J] = chainIndex_{p,:}^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}, \\ \quad\quad hC_m \leftarrow hC_m + fC_n \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_J,k_J} \end{array} \tag{96}$$

The product in Eq. (96) is efficiently calculated by maintaining a list of partial products $pprod = [fC_n, fC_n \cdot gC_{i_1,k_1}, fC_n \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2}, \ldots]$ and only calculating list terms that have changed on each iteration in Eq. (96). The $chainIndex^{(Nx,M,J,K_1,K_2,\ldots,K_{Ng})}$ rows should be collectively sorted to make this process efficient. The subscript vector $i = [i_1, i_2, \ldots, i_J]$ in each row Eq. (95) is descent-sorted, $i_1 \ge i_2 \ge \ldots \ge i_J \ge 0$. The rows are sorted by n; for each n they are sorted by i_1; for each $[n,i_1]$ they are sorted by k_1; for each $[n,i_1,k_1]$ they are sorted by i_2, etc.

<u>Multivariate solve</u>

The multivariate solve algorithm finds a truncated series representation of the solution of a set of equations similar to Eq. (69),

$$\begin{aligned} & f_q(g(x)) = c_q, \quad q = 1:Nf \\ & \text{Solve for } g_{Ng-Nf+1}(x), g_{Ng-Nf+2}(x), \ldots, g_{Ng}(x) \end{aligned} \tag{97}$$

f and c are a length- Nf vectors (with c constant), and g and x are vectors of length Ng and Nx, respectively, with Ng greater than Nf ;

$$Nf < Ng = f_q.Nx, \quad Nx = g_i.Nx \tag{98}$$

$g_1(x)$, $g_2(x)$, …, $g_{Ng-Nf}(x)$ are predetermined, while $g_{Ng-Nf+1}(x)$, …, $g_{Ng}(x)$ constitute Nf unknowns to be determined. $g_{Ng-Nf+1}(0)$, …, $g_{Ng}(0)$ are predetermined; these determine the constants c_1, …, c_{Nf} :

$$c_q = f_q(g(0)) \tag{99}$$

The polynomials' leading constant terms are separated out as in Eq's. (71) and (72),

$$f'_q(g'(x)) = 0 \tag{100}$$

where

$$\begin{aligned} & g'_i(x) = g_i(x) - g_i(0), \quad g'_i(0) = 0 \\ & f'_q(y) = f_q(g(0) + y) - f_q(g(0)), \quad f'_q(0) = 0 \end{aligned} \tag{101}$$

If any f_q is truncated ($f_q.trunc\_deg \neq \infty$), then $g(0)$ must be zero. The primes on f' and g' are henceforth omitted and it is assumed that

$$g_i(0) = 0, \quad f_q(0) = 0, \quad c_q = 0 \tag{102}$$

The min_deg values for f_1, …, f_{Nf} , and $g_{Ng-Nf+1}$, …, g_{Ng} are 1, and are positive for g_1 , …, g_{Ng-Nf} .

$$\begin{aligned} & f_1.min\_deg = \ldots = f_{Nf}.min\_deg = 1 \\ & g_{Ng-Nf+1}.min\_deg = \ldots = g_{Ng}.min\_deg = 1 \\ & g_1.min\_deg \geq 1, \ldots, g_{Ng-Nf}.min\_deg \geq 1 \end{aligned} \tag{103}$$

It is assumed that none of the $g_i(x)$ functions is identically zero $(g_i.min\_deg \neq \infty)$. The polynomial degree of f_q is denoted as J_q,

$$J_q = f_q.deg \tag{104}$$

f_q has a polynomial expansion similar to Eq. (75),

$$\begin{aligned}
f_q(y) &= \sum_n fC_{q,n} \cdot mon(s_{n,:}^{(J_q)}, y) + O\, y^{f_q.trunc_deg} \\
&= \sum_n \left(\text{With } i = s_{n,:}^{(J_q)} \text{ and } y_0 = 1,\ fC_{q,n} \cdot y_{i_1} \cdot y_{i_2} \cdot \ldots \cdot y_{i_{J_q}} \right) + O\, y^{f_q.trunc_deg} \\
h_q(x) &= f_q(g(x)) = 0 \\
&= \sum_n \left(\text{With } i = s_{n,:}^{(J_q)} \text{ and } g_0(x) = 1,\ fC_{q,n} \cdot g_{i_1}(x) \cdot g_{i_2}(x) \cdot \ldots \cdot g_{i_{J_q}}(x) \right) + O\, x^{h_q.trunc_deg}
\end{aligned} \tag{105}$$

The truncation degree $h_q.trunc\_deg$ is defined as in Eq's. (78)-(81). The n-th summand in Eq. (105) has minimum degree $min\_deg\_term_{q,n}$ (from Eq. (78)) and truncation degree $trunc\_deg\_term_{q,n}$ (from Eq's. (80), (79)),

$$min\_deg\_term_{q,n} = \left(\text{With } i = s_{n,:}^{(J_q)},\ \sum_{j \in 1:J_q} g_{i_j}.min\_deg \right) \tag{106}$$

$$span\_deg\_term_{q,n} = \left(\text{With } i = s_{n,:}^{(J_q)},\ \min_{j \in 1:J_q} (g_{i_j}.span\_deg) \right) \tag{107}$$

$$trunc\_deg\_term_{q,n} = span\_deg\_term_{q,n} + min\_deg\_term_{q,n} \tag{108}$$

With $g_0(x) = 1$, $g_0.min\_deg = 0$ in Eq. (106) and $g_0.span\_deg = \infty$ in Eq. (107). j can be limited to the range $1: mon\_deg_n^{(J_q)}$ in Eq's. (106) and (107) because, if $j > mon\_deg_n^{(J_q)}$, then $i_j = 0$, $g_{i_j}(x) = 1$, $g_{i_j}.min\_deg = 0$, and $g_{i_j}.span\_deg = \infty$. $h_q.trunc\_deg$ is defined by Eq. (81),

$$h_q.trunc\_deg \le \min\left(\min_{n \in f_q.Indices} (trunc\_deg\_term_{q,n}), f_q.trunc\_deg \right) \tag{109}$$

(The factor $\min_{i>0}(g_i.min\_deg)$ in Eq. (81) is 1, due to Eq's. (103).) By default, $h_q.trunc\_deg$ is equal to the right side of Eq. (109), but $h_q.trunc\_deg$ can be further reduced to a predetermined truncation limit.

Eq. (109) implies that $h_q.trunc\_deg \le f_q.trunc\_deg$. If $h_q.trunc\_deg < f_q.trunc\_deg$, then $f_q.trunc\_deg$ can be reduced to $h_q.trunc\_deg$ because any $f_q(y)$ term $fC_{q,n} \cdot mon(s_{n,:}^{(J_q)}, y)$ in Eq. (105) of order $f_q.trunc\_deg$ or higher in y will translate to a $h_q(x)$ term

$fC_{q,n} \cdot mon(s_{n,:}^{(J_q)}, g(x))$ of the same or higher order in x. Thus, it can be assumed without loss of generality that $f_q.trunc\_deg \le h_q.trunc\_deg$, and Eq. (109) implies

$$f_q.trunc\_deg \le \min_{n \in f_q.Indices}(trunc\_deg\_term_{q,n}) \tag{110}$$

The following conditions are imposed, for a common truncation degree $trunc\_deg$, to satisfy Eq. (110),

$$f_q.trunc\_deg = g_{Ng-Nf+1}.trunc\_deg = \ldots = g_{Ng}.trunc\_deg = trunc\_deg \tag{111}$$

$$trunc\_deg \le \min_q\left(\min_{n \in f_q.Indices}\left(\text{With } i = s_{n,:}^{(J_q)},\ \min_{j \in 1:J_q}(g_{i_j}.span\_deg) + \sum_{j \in 1:J_q} g_{i_j}.min\_deg\right)\right) \tag{112}$$

The $f_q.trunc\_deg$ and $g_i.trunc\_deg$ terms in Eq. (111) are reduced to a common value to satisfy the equation. ($trunc\_deg$ can be further reduced to a predetermined truncation limit.) If Eq. (112) is not satisfied, then $trunc\_deg$ is reduced to satisfy Eq. (112) and the left-hand terms in Eq. (111) are further reduced to match $trunc\_deg$. (The terms in Eq. (112) with $i_j > Ng - Nf$ can be neglected because, in this case, $g_{i_j}.min\_deg = 1$, and $g_{i_j}.span\_deg = trunc\_deg - 1$. Also, $mon\_deg_n^{(J_q)} \ge 1$ when $n \in f_q.Indices$.)

$trunc\_deg$ is assumed to be finite. (Some special cases do not require truncation, e.g., if f and g are linear functions, but those cases are not considered here.) The polynomial degrees of h_q and $g_{Ng-Nf+1}$, …, g_{Ng} are set to $trunc\_deg - 1$,

$$h_q.deg = M = trunc\_deg - 1 \tag{113}$$

$$K_i = g_i.deg, \quad K_{Ng-Nf+1} = \ldots = K_{Ng} = M \tag{114}$$

$g_i(x)$ has the polynomial expansion in Eq. (90). Eq. (105) is expanded as in Eq's. (94) and (95) (with the left sides of the equations set to zero). Eq. (95) translates to

With $q \in 1:Nf$ and $m \in chainIndex_{:,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}$,

$$0 = \sum_{\left\{p \,\middle|\, chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})} = m\right\}} \left(\begin{array}{c}\text{With } [n, i_1, k_1, i_2, k_2 \ldots, i_{J_q}, k_{J_q}] = \\ chainIndex_{p,2:end}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n} \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_{J_q},k_{J_q}}\end{array}\right) \tag{115}$$

This equation is applied in order of increasing monomial degree, $deg = mon\_deg_m^{(M)}$:

For $deg = 1:M$,

with $q \in 1:Nf$, $m \in chainIndex_{:,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}$, and $mon\_deg_m^{(M)}=deg$, (116)

$$0 = \sum_{\left\{p \,\middle|\, chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}=m\right\}} \begin{pmatrix} \text{With } [n, i_1, k_1, i_2, k_2 \ldots, i_{J_q}, k_{J_q}] = \\ chainIndex_{p,2:end}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n} \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_{J_q},k_{J_q}} \end{pmatrix}$$

The index list $i = [i_1, i_2, \ldots]$ in Eq. (116) corresponds to subscript vector $i = s_{n,:}^{(J_q)}$, and index k_j corresponds to subscript vector $s_{k_j,:}^{(K_{i_j})}$ (Eq. (93)). List i is descent-sorted ($i_1 \geq i_2 \geq \ldots \geq 0$, Eq. (6)). i_1 is nonzero because $f_q.min\_deg > 0$ (Eq's. (103)), implying that $mon\_deg(i) > 0$. The summation terms for which $mon\_deg(i) = 1$ (i.e., $i_2 = i_3 = \ldots = 0$) and $i_1 > Ng - Nf$ are separated out of the sum. (The factors gC_{i_2,k_2}, gC_{i_3,k_3}, … are all 1 for these terms, and the subscript vectors $s_{k_2,:}^{(K_{i_2})}$, $s_{k_3,:}^{(K_{i_3})}$, … are all zero.)

For $deg = 1:M$,

with $q \in 1:Nf$, $m \in chainIndex_{:,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}$, and $mon\_deg_m^{(M)}=deg$,

$$\begin{aligned} 0 = & \sum_{\left\{p \,\middle|\, \begin{array}{l} chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}=m \\ \text{and with } [i_1, i_2]=chainIndex_{p,[3,5]}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ i_1 > Ng-Nf \text{ and } i_2=0 \end{array}\right\}} \begin{pmatrix} \text{With } [n, i_1, k_1] = \\ chainIndex_{p,2:4}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n} \cdot gC_{i_1,k_1} \end{pmatrix} \\ & + \sum_{\left\{p \,\middle|\, \begin{array}{l} chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}=m \\ \text{and with } [i_1, i_2]=chainIndex_{p,[3,5]}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ 0 < i_1 \leq Ng-Nf \text{ or } i_2 > 0 \end{array}\right\}} \begin{pmatrix} \text{With } [n, i_1, k_1, i_2, k_2 \ldots, i_{J_q}, k_{J_q}] = \\ chainIndex_{p,2:end}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n} \cdot gC_{i_1,k_1} \cdot gC_{i_2,k_2} \cdot \ldots \cdot gC_{i_{Jq},k_{J_q}} \end{pmatrix} \end{aligned} \quad (117)$$

(This equation assumes that $J_q > 1$; if $J_q = 1$ then i_2 is implicitly zero and the $i_2 = 0$ and $i_2 > 0$ tests are omitted.)

In the first sum of Eq. (117) the index n for which $s_{n,:}^{(J_q)} = [i_1, 0, 0, \ldots]$ is

$$n = Index([i_1, \overbrace{0, 0, \ldots}^{J_q-1 \text{ zeros}}]) = B(J_q, i_1 - 1) + 1 \quad (i_1 > Ng - Nf) \quad (118)$$

(cf. Eq. (16)). The relationship $s_{m,:}^{(M)} = sort^{(\geq)}([s_{k_1,:}^{(K_{i_1})}, s_{k_2,:}^{(K_{i_2})}, \ldots, s_{k_J,:}^{(K_{i_J})}])_{1:M}$ (Eq. (93)), with $s_{k_2,:}^{(K_{i_2})}$, $s_{k_3,:}^{(K_{i_3})}$, … all zero and $K_{i_1} = M$ (Eq. (114)), implies that $k_1 = m$. The sum simplifies to

$$\sum_{\left\{p \,\middle|\, \begin{array}{l} chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}=m \\ \text{and with } [i_1,i_2]=chainIndex_{p,[3,5]}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ i_1>Ng-Nf \text{ and } i_2=0 \end{array}\right\}} \left(\begin{array}{l} \text{With } [n,i_1,k_1]= \\ \quad chainIndex_{p,2:4}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n}\cdot gC_{i_1,k_1} \end{array} \right)$$

$$= \sum_{i_1=Ng-Nf+1}^{Ng} \left(\text{With } n=B(J_q,i_1-1)+1, \quad fC_{q,n}\cdot gC_{i_1,m}\right) \tag{119}$$

$$= \sum_{i_1=1}^{i_1=Nf} \left(\text{With } n=B(J_q,Ng-Nf+i_1-1)+1, \quad fC_{q,n}\cdot gC_{Ng-Nf+i_1,m}\right) = L_{q,:}\cdot gC_{Ng-Nf+1:Ng,m}$$

where L is a square matrix defined by

$$\text{For } q,i_1=1:Nf, \quad L_{q,i_1}=\left(\text{With } n=B(J_q,Ng-Nf+i_1-1)+1, \quad fC_{q,n}\right) \tag{120}$$

L is a Jacobian matrix, $L_{q,i_1}=\partial f_q(y_1,y_2,\ldots)/\partial y_{Ng-Nf+i_1}\big|_{y=0}$. The L matrix must be nonsingular. Its inverse is denoted $invL$,

$$invL=L^{-1} \tag{121}$$

In the second sum of Eq. (117), if $i_2=0$, then $i_1\le Ng-Nf$ and the summand is $fC_{q,n}\cdot gC_{i_1,k_1}$. (The factors gC_{i_2,k_2}, gC_{i_3,k_3}, … are all 1.) gC_{i_1,k_1} is in this case a coefficient of one of the predetermined functions $g_1(x)$, …, $g_{Ng-Nf}(x)$. In the case $i_2>0$, i_1 and i_2 are both positive; hence $mon\_deg_{k_1}^{(K_{i_1})}$ and $mon\_deg_{k_2}^{(K_{i_2})}$ are both positive. It follows from Eq. (92) that $mon\_deg_{k_1}^{(K_{i_1})}+mon\_deg_{k_2}^{(K_{i_2})}+\ldots=mon\_deg_m^{(M)}$; hence $mon\_deg_{k_1}^{(K_{i_1})}$, $mon\_deg_{k_2}^{(K_{i_2})}$, … are all strictly less than $mon\_deg_m^{(M)}$ in the second sum and the summand factors gC_{i_1,k_1}, gC_{i_2,k_2}, … will have been determined from a previous iteration of Eq. (117) (for smaller deg). Thus, Eq. (117) can be solved for the unknowns $gC_{:,m}$ in the first sum,

For $deg=1:M$,

with $m\in chainIndex_{:,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}$, and $mon\_deg_m^{(M)}=deg$,

$$gC_{Ng-Nf+i_1,m}=$$

$$-\sum_{q\in 1:Nf} invL_{i_1,q}\cdot\left(\sum_{\left\{p \,\middle|\, \begin{array}{l} chainIndex_{p,1}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}=m \\ \text{and with } [i_1,i_2]=chainIndex_{p,[3,5]}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ i_1\le Ng-Nf \text{ or } i_2>0 \end{array}\right\}} \left(\begin{array}{l} \text{With } [n,i_1,k_1,i_2,k_2\ldots,i_{J_q},k_{J_q}]= \\ \quad chainIndex_{p,2:end}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}, \\ fC_{q,n}\cdot gC_{i_1,k_1}\cdot gC_{i_2,k_2}\cdot\ldots\cdot gC_{i_{J_q},k_{J_q}} \end{array} \right)\right)$$

(122)

Eq. (122) is implemented procedurally as

Initialize $gC_{Ng-Nf+1:Ng,:}=0$.
For $deg=1:M$,
For $q=1:Nf$, $p=1,2,\ldots$
with $[m,n,i_1,k_1,i_2,k_2\ldots,i_{J_q},k_{J_q}]=chainIndex_{p,:}^{(Nx,M,J_q,K_1,K_2,\ldots,K_{Ng})}$,
if $mon\_deg_m^{(M)}$=deg and ($i_1 \le Ng-Nf$ or $i_2>0$), (123)
$gC_{Ng-Nf+q,m} \leftarrow gC_{Ng-Nf+q,m} - fC_{q,n}\cdot gC_{i_1,k_1}\cdot gC_{i_2,k_2}\cdot\ldots\cdot gC_{i_{J_q},k_{J_q}}$
For $i_1=1:Nf$, $m=1:B(M,Nx)$
if $mon\_deg_m^{(M)}=deg$
$gC_{Ng-Nf+1:Ng,m} \leftarrow invL\cdot gC_{Ng-Nf+1:Ng,m}$

<u>ODE integration</u>

The ODE integration algorithm integrates a set of ordinary differential equations of the form

$$Dg_{Ng-Nf+q}(x)=f_q(g(x)) \quad (q\in 1:Nf) \tag{124}$$

x is scalar; f and g are vectors of length Nf and Ng, respectively,

$$Nf \le Ng = \max_q(f_q.Nx), \quad g_i.Nx=1 \tag{125}$$

Dg_q is the derivative of g_q. The functions $g_1(x)$, $g_2(x)$, …, $g_{Ng-Nf}(x)$ are predetermined, while $g_{Ng-Nf+1}(x)$, …, $g_{Ng}(x)$ are to be determined. ($g_{Ng-Nf+1}(0)$, …, $g_{Ng}(0)$ are predetermined initial values.) The g polynomials' leading constant terms are separated out,

$$Dg'_{Ng-Nf+q}(x)=f'_q(g'(x)) \quad (q\in 1:Nf) \tag{126}$$

where

$$\begin{aligned} g'(x)&=g(x)-g(0), \quad g'(0)=0 \\ f'_q(y)&=f_q(g(0)+y) \end{aligned} \tag{127}$$

If any f_q is truncated ($f_q.trunc\_deg\ne\infty$), then $g(0)$ must be zero. The primes on f' and g' are henceforth omitted and it is assumed that

$$g(0)=0 \tag{128}$$

f_q has minimum, maximum, and truncation degrees $f_q.min\_deg$, $f_q.deg$, and $f_q.trunc\_deg$. g_i has minimum, maximum, and truncation degrees $g_i.min\_deg$, $g_i.deg$, and $g_i.trunc\_deg$. Eq. (128) implies that

$$g_i.min\_deg > 0, \quad i \in 1:Ng \tag{129}$$

If $f(0)=0$, then Eq. (124) with initial condition $g(0)=0$ has the trivial solution $g(x)=0$. It is assumed that $f(0)$ is nonzero,

$$Dg_{Ng-Nf+q}(0) = f_q(0) \neq 0 \text{ for at least one } q \in 1:Nf \tag{130}$$

The condition $Dg_{Ng-Nf+q}(0) \neq 0$ implies that $g_i.min\_deg \le 1$ for at least one i in $Ng-Nf+1:Ng$. Hence, it follows from Eq's. (129) and (130) that

$$\min_i(g_i.min\_deg) = 1 \tag{131}$$

The polynomial degree of f_q is denoted as J_q,

$$J_q = f_q.deg \tag{132}$$

f_q has a polynomial expansion,

$$\begin{aligned}
f_q(y) &= \sum_{n \in f_q.Indices} fC_{q,n} \cdot mon(s_{n,:}^{(J_q)}, y) + O\, y^{f_q.trunc_deg} \\
&= \sum_{n \in f_q.Indices} \left(\text{With } i = s_{n,:}^{(J_q)} \text{ and } y_0{=}1,\ \ fC_{q,n} \cdot y_{i_1} \cdot y_{i_2} \cdot \ldots \cdot y_{i_{J_q}} \right) + O\, y^{f_q.trunc_deg} \\
Dg_{Ng-Nf+q}(x) &= f_q(g(x)) \\
&= \sum_{n \in f_q.Indices} \left(\text{With } i = s_{n,:}^{(J_q)} \text{ and } g_0(x){=}1,\ \ fC_{q,n} \cdot g_{i_1}(x) \cdot g_{i_2}(x) \cdot \ldots \cdot g_{i_{J_q}}(x) \right) \\
&\qquad + O\, x^{g_{Ng-Nf+q}.trunc_deg-1} \\
&\quad (q \in 1:Nf)
\end{aligned} \tag{133}$$

The derivative $Dg_{Ng-Nf+q}(x)$ in Eq. (133) has truncation degree $g_{Ng-Nf+q}.trunc\_deg-1$. This is equal to the truncation degree of $f_q(g(x))$, which is limited by $f_q.trunc\_deg \cdot \min_i(g_i.min\_deg)$. It thus follows from Eq. (131) that

$$g_{Ng-Nf+q}.trunc\_deg - 1 \le f_q.trunc\_deg \quad (q \in 1:Nf) \tag{134}$$

If $g_{Ng-Nf+q}.trunc\_deg$ is limited to a predetermined upper bound, then $f_q.trunc\_deg$ can be reduced so that $f_q.trunc\_deg+1$ is equal to the bound, consistent with Eq (134).

The n-th summand in Eq. (133) has minimum degree $min\_deg\_term_{q,n}$ and truncation degree $trunc\_deg\_term_{q,n}$, as in Eq's. (106)-(108),

$$min\_deg\_term_{q,n} = \left(\text{With } i = s^{(J_q)}_{n,:},\ \sum_{j \in 1:J_q} g_{i_j}.min\_deg \right) \quad (135)$$

$$span\_deg\_term_{q,n} = \left(\text{With } i = s^{(J_q)}_{n,:},\ \min_{j \in 1:J_q} (g_{i_j}.span\_deg) \right) \quad (136)$$

$$trunc\_deg\_term_{q,n} = span\_deg\_term_{q,n} + min\_deg\_term_{q,n} \quad (137)$$

With $g_0(x) = 1$, $g_0.min\_deg = 0$ in Eq. (135) and $g_0.span\_deg = \infty$ in Eq. (136). The truncation degree of $Dg_{Ng-Nf+q}(x)$ (i.e., $g_{Ng-Nf+q}.trunc\_deg - 1$) cannot exceed $trunc\_deg\_term_{q,n}$, and this condition is combined with Eq. (134),

$$g_{Ng-Nf+q}.trunc\_deg - 1 \le \min(\min_{n \in f_q.Indices} (trunc\_deg\_term_{q,n}), f_q.trunc\_deg) \quad (138)$$

$g_{Ng-Nf+q}.trunc\_deg$ can be further limited to a predetermined upper bound, and $g_{Ng-Nf+q}.trunc\_deg$ is maximized subject to these limits. (If Eq. (138) is insufficient to limit $g_{Ng-Nf+q}.trunc\_deg$, then a predetermined limit must be imposed.)

$g_i(x)$ has the polynomial expansion

$$g_i(x) = \sum_{k \in g_i.Indices} gC_{i,k} \cdot mon(s^{(K_i)}_{k,:}, x) + O\, x^{g_i.trunc\_deg}, \quad K_i = g_i.deg \quad (i \in 1:Ng) \quad (139)$$

For $i \in Ng - Nf + 1 : Ng$, g_i has polynomial degree $g_i.trunc\_deg - 1$,

$$K_i = g_i.trunc\_deg - 1 \text{ for } i \in Ng - Nf + 1 : Ng \quad (140)$$

In the context of Eq. (139), x is scalar ($g_i.Nx = 1$) and $s^{(K_i)}_{k,:} = s^{(K_i,1)}_{k,:}$, where $s^{(K_i,1)}$ has the form

$$s^{(K_i,1)} = \overbrace{\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}}^{K_i \text{ columns}} \left. \vphantom{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \right\} K_i + 1 \text{ rows}, \quad s^{(K_i,1)}_{j,k} = (j > k), \quad g_i.Indices \subset 2 : K_i + 1 \quad (141)$$

(The logical expression "$j > k$" is implicitly cast to an integer, 0 if false or 1 if true.) The summation set $g_i.Indices$ in Eq. (139), excludes 1 due to Eq. (128).) The monomial functions defined in Eq. (139) are powers of x,

$$mon(s_{k,:}^{(K_i,1)}, x) = x^{k-1} \quad (k > 1) \tag{142}$$

$$g_i(x) = \sum_{k+1 \in g_i.Indices} gC_{i,k+1} \cdot x^k + O\, x^{g_i.trunc\_deg} \quad (i \in 1:Ng) \tag{143}$$

Eq's. (140) and (143) are substituted in Eq. (133),

$$\begin{aligned} f_q(g(x)) &= \sum_{n \in f_q.Indices} \left(\text{With } i = s_{n,:}^{(J_q)} \text{ and } g_0(x) = 1, \quad fC_{q,n} \cdot g_{i_1}(x) \cdot g_{i_2}(x) \cdot \ldots \cdot g_{i_{J_q}}(x) \right) \\ &\quad + O\, x^{K_{Ng-Nf+q}} \\ &= \sum_{n \in f_q.Indices} fC_{q,n} \cdot \left(\begin{array}{l} \text{With } i = s_{n,:}^{(J_q)}.\ g_0(x) = 1, \text{ and } g_0.Indices = \{1\}, \\ \displaystyle\sum_{\substack{k_1+1 \in g_{i_1}.Indices \\ k_2+1 \in g_{i_2}.Indices \\ \ldots \\ k_{J_q}+1 \in g_{i_{J_q}}.Indices}} gC_{i_1,k_1+1} \cdot gC_{i_2,k_2+1} \cdot \ldots \cdot gC_{i_{J_q},k_{J_q}+1} \cdot x^{k_1+k_2+\ldots+k_{J_q}} \end{array} \right) \\ &\quad + O\, x^{K_{Ng-Nf+q}} \\ &= Dg_{Ng-Nf+q}(x) = \sum_{k+1 \in g_{Ng-Nf+q}.Indices} gC_{Ng-Nf+q,k+1} \cdot k \cdot x^{k-1} + O\, x^{K_{Ng-Nf+q}} \\ &\quad (q \in 1:Nf) \end{aligned} \tag{144}$$

Corresponding powers of x are matched to solve Eq. (144) for $gC_{Ng-Nf+q,k+1}$,

$$gC_{Ng-Nf+q,k+1} = \frac{1}{k} \cdot \sum_{n \in f_q.Indices} fC_{q,n} \cdot \left(\begin{array}{l} \text{With } i = s_{n,:}^{(J_q)},\ gC_{0,k_j+1} = (k_j = 0), \text{ and } g_0.Indices = \{1\}, \\ \displaystyle\sum_{\substack{k_1+1 \in g_{i_1}.Indices \\ k_2+1 \in g_{i_2}.Indices \\ \ldots \\ k_{J_q}+1 \in g_{i_{J_q}}.Indices \\ k_1+k_2+\ldots+k_{J_q} = k-1}} gC_{i_1,k_1+1} \cdot gC_{i_2,k_2+1} \cdot \ldots \cdot gC_{i_{J_q},k_{J_q}+1} \end{array} \right)$$

$$(q \in 1:Nf,\ k+1 \in g_{Ng-Nf+q}.Indices)$$

(145)

(The logical expression "$k_j = 0$" is implicitly cast to an integer, 0 if false or 1 if true.) The condition $k_1 + k_2 + \ldots + k_{deg} = k - 1$ in the sum implies that k_1, k_2, … are all less than k. Thus, iteration of the above formula in order of increasing k will determine $gC_{Ng-Nf+q,k+1}$ from previously determined coefficients gC_{i_1,k_1+1}, gC_{i_2,k_2+1}, …, $gC_{i_{deg},k_{deg}+1}$. The index sets

$g_{Ng-Nf+1}.Indices$, $g_{Ng-Nf+2}.Indices$, … $g_{Ng}.Indices$ can initially be empty, and are then augmented as the coefficients are calculated.

The conditions $k_1 + k_2 + \ldots + k_{deg} = k - 1$ and $k_j > 0$ (from Eq. (128)) further imply that $k_j + deg \leq k$. The index set $g_{i_j}.Indices$ is not known in advance for $i_j > Ng - Nf$, but the condition $k_j + 1 \in 2 : k + 1 - deg$ implies that the condition $k_j + 1 \in g_{i_j}.Indices$ can be replaced by $k_j + 1 \in g_{i_j}.Indices \cap 2 : k + 1 - deg$ in the sum. If $k < 1 + deg$ or any set $g_{i_j}.Indices \cap 2 : k + 1 - deg$ is empty, then the sum is zero.