Polynomial Inclusion Functions*

E. de Weerdt, E. van Kampen, Q. P. Chu, and J. A. Mulder
Delft University of Technology, Faculty of Aerospace
Engineering, Control and Simulation Division
E.deWeerdt@TUDelft.nl

Abstract

When using interval analysis, the bounds of an inclusion function are often non-tight due to dependency effects. The benefit of Taylor Models (TMs) or Verified Taylor Series (VTSs) is the use of higher order derivatives terms, significantly reducing the dependency effect. In this paper, it is assumed that the required information to derive these inclusion functions is obtained using automatic differentiation. The drawback of TMs and VTSs is that not all available information is used, resulting in non-optimal inclusion functions. In this paper the Polynomial Inclusion Function (PIF) is presented, which is guaranteed to form equal or shaper enclosures than any (combination of) Taylor Model(s) defined *using the same set of information*. The PIF is derived for the one dimensional case. Extension to n-dimensional functions is performed via application of the PIF to every dimension independently. The performance of the PIF is compared to that of Verified Taylor Series for multiple (non-linear) functions and is shown to yield to superior inclusions. Moreover, unlike with TMs or VTSs, increasing the order of the PIF will always sharpen its bounds.

Keywords: Polynomial inclusion function, interval analysis, Taylor Models, Verifies Taylor Series, inclusion function
AMS subject classifications: 65-00

1 Introduction

For any global optimization problem it is crucial to find guaranteed bounds, i.e. an inclusion function, on a given (usually non-linear) function f whose output depends on the variable parameters for which it is optimized. With these bounds one can find *all* global/local solutions of the given problem using simple branch and bound methods. For all optimization problems, it holds that the computational load needed to solve the problem highly depends on the sharpness of the obtained bounds. The sharper the bounds, the faster regions of the search space can be discarded, leading to less computational effort (unless the cost of obtaining sharper bounds is too high). Finding

*Submitted: April 16, 2010; Revised: July 16, 2010, October 29, 2010, September 18, 2012, and November 8, 2012; Accepted: November 13, 2012.

sharper bounds at low cost is therefore a crucial part of solving optimization problems.

Most inclusion functions describe the bounds on a function as a polynomial. When using standard interval arithmetic one obtains a zero-order polynomial describing the upper bound and one for the lower bound. In the centered forms [6, 7, 8] the obtained inclusion is a set of two first order polynomials for the upper bound and another set for the lower bound. Higher order polynomial inclusion functions can be obtained using Taylor series theory. One of such inclusion function is the Taylor Model. In this paper it is assumed that the set of information required to construct these type of inclusion functions is obtained via automatic differentiation in combination with interval arithmetic. All claims in this paper are based on this assumption.

Taylor models have been developed by Lanford around 1980, subsequently studied by Eckmann, Koch, Wittwer, Berz, Makino and Hoefkens [17, 12, 13, 10], and have been applied successfully to many problems [11, 2]. Taylor Models are shown to have excellent performance in situations where the domain is small[1] and are often used to remove the wrapping effect encountered when performing guaranteed integration. For larger domains the order of the polynomial must increase to prevent a remainder blow-up. For optimization problems in which the dependency effects increase when taking a higher order derivative of the function [5] the remainder blow-up can only be prevented by reducing the domain width (e.g., bisecting the domain). Although widely applied in the reliable computing community and applicable in n dimensional problems, the Taylor Models are not always optimal, and other inclusion functions can result in tighter bounds [5]. Generally speaking the optimal form of the inclusion function is problem dependent.

The main focus of this paper is on deriving a generic inclusion function which uses *all available* information as efficiently as possible to form the sharpest possible bounds. In case of Taylor Models one needs to derive, over the entire domain $[x]$, the natural inclusion function of the n^{th} order derivative ($[\nabla^n f([x])$) and the 0 to $(n-1)^{th}$ order derivative value at the expansion point. As previously stated, the assumption is that the derivative information is derived using automatic differentiation techniques. The key aspect is that when the n^{th} order derivative information is derived one automatically obtains information regarding the derivatives of lower order (0 to $(n-1)$)[2]. This means that the entire information set $\mathcal{S}$, for a one-dimensional function f, is given by:

$$\mathcal{S} : \left\{ \left.\frac{d^i f}{dx^i}\right|_{x=x_0}, \left[\left.\frac{d^j f}{dx^j}\right|_{x=[x]} \right] \right\}, i = 0, 1, ..., n \ \ , j = 0, 1, ..., n+1. \tag{1}$$

In the current application, the Taylor Models (or Taylor series) only use the information obtained from $[\nabla^n f([x])]$ leaving a lot of available information unused.

The first contribution of this paper is the derivation of a new inclusion function called the polynomial inclusion function (PIF) which provides a guaranteed (equal or) sharper inclusion than any (combination of) Taylor Model(s) derived using the same set of information. The PIF is an inclusion function based on a set of piecewise polynomials describing the guaranteed upper ($P_{\bar{f}}$) and lower bound ($P_{\underline{f}}$) on function f for domain

[1] The definition of 'small' is problem dependent.

[2] Automatic differentiation packages 'record' a directed acyclic graph (DAG) while computing the function value [18]. When the n^{th} order derivative is computed, the DAG automatically contains the information of the 0 to $(n-1)^{th}$ order derivative.

$[\mathbf{x}]$:

$$f(\mathbf{x}) \subseteq \left[P_{\underline{f}}(\mathbf{x}), P_{\bar{f}}(\mathbf{x})\right] \ \forall \mathbf{x} \in [\mathbf{x}]. \tag{2}$$

This paper focuses on the theoretical development of the new inclusion function. The goal is to show information can be used more efficiently than for Taylor Models. Since the computational load is an important aspect of any optimization algorithm, a cost analysis is performed. The second contribution of this paper is the introduction of a computational 'light' version of the proposed PIF to show that faster optimization is possible than when using Taylor Models.

In this paper the one-dimensional case is described. The PIF can be extended to multiple dimensions by applying it to each dimension separately as will be demonstrated in section 6. The one-dimensional case frequently is encountered in dynamic optimization problems, e.g., trajectory optimization, where the cost function commonly includes an integral [4] over time:

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t)\, dt, \tag{3}$$

where Φ is a function denoting the penalty for end-point constraints, and $\mathcal{L}$ is the Lagrangian. By having a sharp inclusion of the Lagrangian (a generally non-linear function) in polynomial form, the integral can be computed easily. Another large field of research is that of bounding the solutions of ODEs (required in for instance initial value problems) for which Taylor Models frequently are used [9][16](Packages COSY-INFINITY [3], VSPODE [9], and VNODE-LP [15] all incorporate some form of Taylor Models). The proposed inclusion function can be applied in this field as well.

In section 2, a brief review of guaranteed Taylor series expansions is given. The effect of Taylor series order on the sharpness of the inclusion function will be clearly demonstrated. In section 3 the derivation of the PIF will be performed in steps. With each step the information content of the PIF will be increased resulting in guaranteed sharper bounds (over the entire domain) compared to the previous step. The derived PIF will be validated and compared to Verified Taylor Series (introduced in the next section) for several examples in section 4. A cost analysis is performed in section 5, and the extension of the proposed method to higher dimensional function is given in section 6. Finally, in section 7, conclusions and recommendations for further work are given.

2 Verified Taylor Series (VTS)

Any (non-linear) function f can be approximated by a Taylor series. By using interval analysis one can define an inclusion function based on a Taylor series expansion which provides guaranteed bounds on f for a given domain. Taylor's formula with remainder for a one dimensional function is given by:

$$f(x) = \sum_{v=0}^{n} \frac{f^{(v)}(x_0)}{v!}(x - x_0)^v + \frac{g(x) - g(x_0)}{g^{(1)}(\xi)} \frac{f^{(n+1)}(\xi)}{n!}(x - \xi)^n, \tag{4}$$

where g is an arbitrary function with non vanishing derivative strictly between x_0 and x, ξ lie strictly between x_0 and x, and $f^{(v)} = d^v f(x)/dx^v$ [3]. The form of the remainder depends on the choice of the function g. When choosing

$$\begin{array}{l} g(y) = (x-y)^{n+1} \\ g^{(1)}(y) = -(n+1)(x-y)^n \end{array} , \tag{5}$$

where y is the independent variable, one obtains the Lagrange's Remainder:

$$L^n_{f,x_0}(\xi, x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}, \tag{6}$$

where again ξ lies strictly between x_0 and x. For the multivariate case we can use the following notation for the n-th order Taylor polynomial and corresponding remainder:

$$\begin{array}{lcl} T^n_{f,\mathbf{x}_0}(\mathbf{x}) & = & \sum\limits_{0\le i_1+\ldots+i_d\le n} \left\{ \frac{\partial^{i_1+\cdots+i_d} f(\mathbf{x})}{\partial x_1^{i_1}\ldots\partial x_1^{i_d}} \cdot \frac{(x_1-x_{0,1})^{i_1}\cdot\ldots\cdot(x_d-x_{0,d})^{i_d}}{i_1!\cdot\ldots\cdot i_d!} \right\} \\ L^n_{f,\mathbf{x}_0}(\xi,\mathbf{x}) & = & \sum\limits_{i_1+\ldots+i_d=n+1} \left\{ \frac{\partial^{n+1} f(\xi)}{\partial x_1^{i_1}\ldots\partial x_1^{i_d}} \cdot \frac{(x_1-x_{0,1})^{i_1}\cdot\ldots\cdot(x_d-x_{0,d})^{i_1}}{i_1!\cdot\ldots\cdot i_d!} \right\} \end{array} . \tag{7}$$

2.1 Forming a Guaranteed Inclusion Function

The bounds on the remainder can be determined by substituting the interval $[\mathbf{x}]$ for all $\mathbf{x}$ and ξ in the expression for $L^n_{f,\mathbf{x}_0}$. By using interval arithmetic rigorous bounds are obtained. This approach results in a single interval for the remainder I:

$$\begin{array}{lcl} I = L^n_{f,\mathbf{x}_0}([\mathbf{x}],[\mathbf{x}]) & = & \sum\limits_{i_1+\ldots+i_d=n+1} \left\{ \frac{\partial^{n+1} f([\mathbf{x}])}{\partial x_1^{i_1}\ldots\partial x_1^{i_d}} \cdot \right. \\ & & \left. \frac{([x_1]-x_{0,1})^{i_1}\cdot\ldots\cdot([x_d]-x_{0,d})^{i_1}}{i_1!\cdot\ldots\cdot i_d!} \right\} \end{array} . \tag{8}$$

The combination of (T, I) is called a Taylor Model [2]. The width of I depends on the width of $[\mathbf{x}]$ and on the function $f(\mathbf{x})$. If the domain $[\mathbf{x}]$ is large then the multiplication $\frac{(x_1-x_{0,1})^{i_1}\cdot\ldots\cdot(x_d-x_{0,d})^{i_1}}{i_1!\cdot\ldots\cdot i_d!}$ causes a blow-up of the remainder if the $(n+1)$-th order derivative of f is non-zero. Another difficulty is the number of occurrences of $\mathbf{x}$ in the $(n+1)$-th order derivative of f. As for any function multiple occurrences of $\mathbf{x}$ can cause overestimation, which yields large remainder bounds [5]. As an alternative method one can use Taylor Model arithmetic from Makino and Berz to derive the remainder. A trade-off between accuracy and speed must be made to decide which method to use. In this paper the method of deriving the Taylor coefficients is used.

What is important to realize is that the remainder also can be kept as a function of $\mathbf{x}$. Taylor series theory states that a guaranteed inclusion can be formed when inserting $[\mathbf{x}]$ for only ξ:

$$\begin{array}{lcl} I(\mathbf{x}) = L^n_{f,\mathbf{x}_0}([\mathbf{x}],\mathbf{x}) & = & \sum\limits_{i_1+\ldots+i_d=n+1} \left\{ \frac{\partial^{n+1} f([\mathbf{x}])}{\partial x_1^{i_1}\ldots\partial x_1^{i_d}} \cdot \right. \\ & & \left. \frac{(x_1-x_{0,1})^{i_1}\cdot\ldots\cdot(x_d-x_{0,d})^{i_1}}{i_1!\cdot\ldots\cdot i_d!} \right\} \end{array} . \tag{9}$$

The resulting inclusion function, i.e. $(T, I(\mathbf{x}))$, is called a Verified Taylor Series (VTS) in this paper. From the definition of the VTS and TM one can derive the following:

[3] For the work in this paper the function f may be analytically or numerically differentiable.

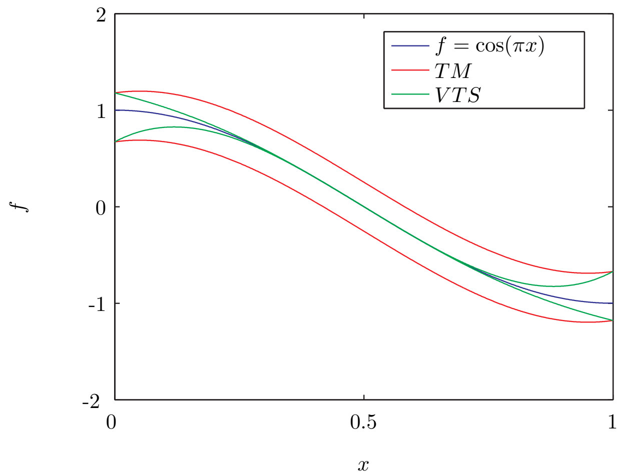

Figure 1: Inclusion of $f = \cos(x\pi), x \in [0, 1]$ using the Taylor Model and Verified Taylor Series. The expansion point for both inclusion functions is $x_0 = 0.5$, and the order is 3.

there exists a parameter $\delta_1 \le \mathbf{x}_0$ and a parameter $\delta_2 \ge \mathbf{x}_0$ such that the following holds[4]:

$$VTS(\mathbf{x}) \subseteq TM(\mathbf{x}), \forall \mathbf{x} \in [\delta_1, \delta_2]. \tag{10}$$

Under the assumption that automatic differentiation is used to derive the required information for construction of the VTS and TM the values of δ_1 and δ_2 can be set to $\inf[\mathbf{x}]$ and $\sup[\mathbf{x}]$ respectively. This means that the following holds: $VTS(\mathbf{x}) \subseteq TM(\mathbf{x}), \forall \mathbf{x} \in [\mathbf{x}]$. The difference between the TM and VTS is demonstrated in Figure 1. Since the VTS yields sharper inclusions it is used as a reference in the remainder of this paper.

2.2 Use of Derivative Information

The n^{th} order VTS is formed using the natural inclusion function of the $(n+1)^{th}$ order derivatives. As stated in the introduction, it is assumed that the derivative natural inclusion function $[f^{(n+1)}([x])]$ is derived using automatic differentiation techniques such that all other derivatives up to order $(n+1)$ are readily available. Since only $[f^{(n+1)}([x])]$ is used for the construction of the TM/VTS, a lot of valuable information is disregarded. In Figure 2 VTSs up to order 4 for function $f = \cos(\pi x), x \in [0, 3]$ are given. As one can clearly see, the VTSs of order > -1 all violate the bounds of the $[f([x])]$ (equal to $VTS_f^{-1}(x)$) at some point $x \in [x]$. Moreover, the bounds of other derivative inclusion functions may also be violated, i.e., upper bound violation:

$$\sup\left[\left.\frac{\partial^i VTS}{\partial x^i}\right|_{x^*}\right] > \sup\left[\left.\frac{\partial^i f}{\partial x^i}\right|_{[x]}\right], x^* \in [x], \tag{11}$$

[4]This results is independent of the way the remainder of the TM is derived, e.g. automatic differentiation or TM arithmetic.

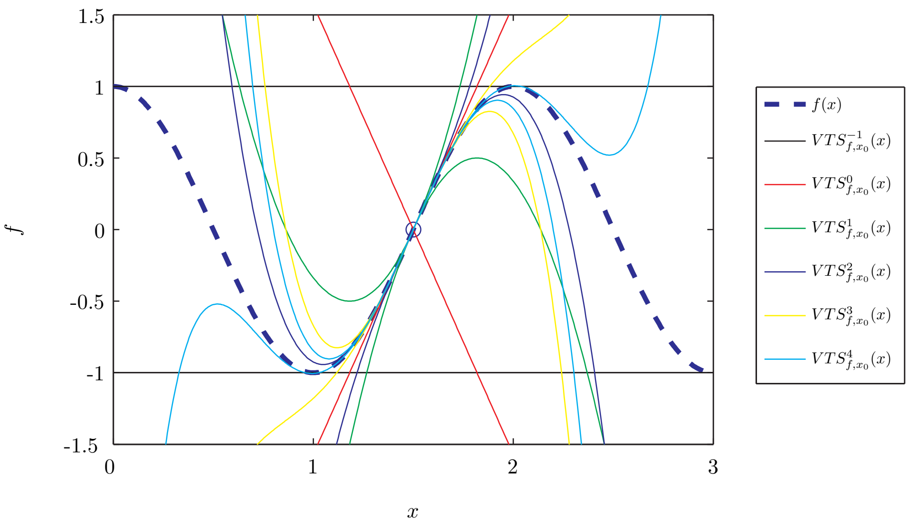

Figure 2: Taylor Models for function $f = cos(x\pi)$ for domain $X = [0, 3]$ and expansion point $x_0 = 1.5$.

(same holds for lower bound violations). Since the bounds of $[f^{(i)}([x])]$ are guaranteed, one can use this information to improve the inclusion. In the following section a method is given that makes sure that the VTS will not violate any of the bounds set by the derivative inclusion functions.

3 Polynomial Inclusion Function (PIF)

The discussion on the Verified Taylor Series showed that not all available information regarding derivative inclusions is used. In this section the polynomial inclusion function (PIF) is derived that uses the available derivative bounds more efficiently. The resulting PIF provides a guaranteed equal or sharper enclosure than any VTS constructed based on the same set of available information. The PIF consists of two piecewise polynomials, one bounding the function from above and one from below:

$$f\left(x\right) \subseteq PIF^d_f\left(x\right) = \left[P^d_{\underline{f}}\left(x\right), P^d_{\bar{f}}\left(x\right)\right] \ \forall x \in [x], \tag{12}$$

where f denotes the function and d the maximal derivative order of which information is used in the construction of the PIF.

The PIF derived in this paper is based on Taylor series expansion theory. To simplify the discussion the derivation of the PIF starts by looking at the upper bound on function f for $x \geq x_0$. All other remaining bounds (upper bound and lower bound for $x \leq x_0$, and lower bound for $x \geq x_0$) can be derived using the same procedure after having performed a simple coordinate mapping. The results given in this section show the PIF for the entire domain to demonstrate the overall performance. The example of $f = \cos(x\pi), x_0 = 1.5, x \in [0, 3]$ given in Figure 2 is used to explain the consequences of each step in the derivation. With each step, the bounds of the PIF become more tight or remain equal over the entire domain, i.e., for every $x \in [x]$ the bounds of the new PIF are equal to or within the bounds of the PIF of the previous step.

3.1 Combining VTS

Proposition 3.1. *Given a certain set $\mathcal{S}$ containing derivative inclusion functions for domain $[x]$ and derivative evaluations at location x_0 for function f,*

$$\mathcal{S} : \left\{ \left. \frac{d^i f}{dx^i} \right|_{x=x_0}, \left[\left. \frac{d^j f}{dx^j} \right|_{x=[x]} \right] \right\}, i = 0, 1, ..., n \;\; , j = 0, 1, ..., n+1 \quad , x_0 \in [x], \tag{13}$$

it is guaranteed that the optimal inclusion function has sharper or equal bounds compared to the bounds of VTS of any degree, which has been derived using the same set of information, for the entire domain $x \in [x]$.

Proof. Since the VTS of any given degree is a guaranteed inclusion function of function f:

$$f\left(x\right) \subseteq VTS^{d}_{f,x_0}\left(x\right), \forall d \leq n, \forall x \in [x], \tag{14}$$

one can select the sharpest bounds at each $x \in [x]$ provided by one of the VTS. The resulting bound is guaranteed to correctly bound the function f from above and below:

$$\max_d \left\{ \inf \left[VTS^{d}_{f,x_0}\left(x\right) \right] \right\} \geq f\left(x\right) \leq \min_d \left\{ \sup \left[VTS^{d}_{f,x_0}\left(x\right) \right] \right\} \quad \forall d \leq n, \ \forall x \in [x]. \tag{15}$$

Since all VTSs are derived using the given derivative information, the optimal inclusion function based on the same information must yield sharper or equal bounds. □

Proposition 3.1 can be used to derive a PIF. To derive the piecewise polynomial P^d_f, one must determine the intersection points between each pair of VTSs to determine the lowest valued VTS upper bound for each location in $[x]$. Finding the intersection points can be formulated as a root finding problem: $h(x) = \sup VTS^z(x) - \sup VTS^y(x) = 0$, for which many methods are available. For polynomials up to order 4 a closed form solution exists [1] (such as the quadratic formula for order 2). In the work of Mekwi [14] the methods of Bairstow, Bernoulli, Graeffe, Müller, Newton-Raphson, Jenkins-Traub and Laguerre are explained. Most available methods provide numeric approximations for the roots, and others use interval analysis to rigorously bound all roots. Irrespectively of the applied method for finding the roots, the PIF will remain a guaranteed inclusion of the function if the domain switch point x^* is chosen such that $\inf VTS^z_{f,x_0}(x^*) \geq \sup VTS^y_{f,x_0}(x^*)$ when transforming from degree z to y. For the given example of $f = \cos(x\pi)$ the resulting PIF is represented in Figure 3. Note that the sharpest bound is **not** necessarily formed by the highest order VTS as can be seen from the PIF^4_{f,x_0}:

$$P^4_f = \begin{cases} \sup VTS^{-1}_{f,x_0}\left(x\right), & x \in [0.000, 0.595] \\ \sup VTS^{2}_{f,x_0}\left(x\right), & x \in [0.595, 1.503] \\ \sup VTS^{4}_{f,x_0}\left(x\right), & x \in [1.503, 1.978] \\ \sup VTS^{-1}_{f,x_0}\left(x\right), & x \in [1.978, 2.044] \\ \sup VTS^{4}_{f,x_0}\left(x\right), & x \in [2.044, 2.675] \\ \sup VTS^{-1}_{f,x_0}\left(x\right), & x \in [2.675, 3.000] \end{cases} \tag{16}$$

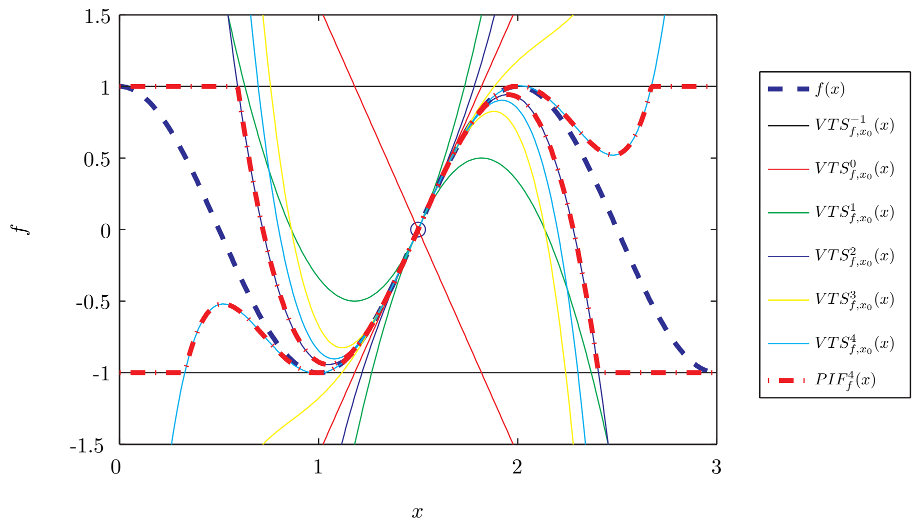

Figure 3: Verified Taylor Series (VTSs) and Polynomial Inclusion Function (PIF) for function $f = cos(x\pi)$ for domain $X = [0, 3]$ and expansion point $x_0 = 1.5$. The PIF is formed by applying Proposition 3.1.

$$P^4_{\underline{f}} = \begin{cases} \inf VTS^{-1}_{f,x_0}(x), & x \in [0.000, 0.331] \\ \inf VTS^{4}_{f,x_0}(x), & x \in [0.331, 0.962] \\ \inf VTS^{-1}_{f,x_0}(x), & x \in [0.962, 1.028] \\ \inf VTS^{4}_{f,x_0}(x), & x \in [1.028, 1.503] \\ \inf VTS^{2}_{f,x_0}(x), & x \in [1.503, 2.411] \\ \inf VTS^{-1}_{f,x_0}(x), & x \in [2.411, 3.000] \end{cases} . \tag{17}$$

3.2 Lowering VTS Order

The PIF formed using Proposition 3.1 always provides equal or sharper bounds than any VTS. However, the information regarding the guaranteed derivatives bounds is not fully used, i.e., the PIF may still violate these bounds. Proposition 3.2 can be used to form a PIF that, per domain, will consist of a polynomial that does not violate any derivative bounds (up to the order of the polynomial).

Proposition 3.2. *Consider a polynomial $P^d_{x_0}(x)$:*

$$P^d_{x_0}(x) = \sum_{j=0}^{d+1} \frac{1}{j!} a_i \left(x - x_0\right)\right)^j , \tag{18}$$

defined on domain $[x]$ for which

$$\frac{d^i f(x)}{dx^i} \leq \frac{d^i P^d_{x_0}(x)}{dx^i}, \quad \forall i \in [0, d+1], \forall x \in [x]. \tag{19}$$

Then the polynomial

$$P^q_{x^*}(x) = \sum_{j=0}^{q} \left\{ \frac{1}{j!} \left. \frac{d^i P^d_{x_0}(x)}{dx^i} \right|_{x^*} (x - x*)^j \right\} + \frac{1}{(q+1)!} b \left(x - x*\right)^{q+1} , \tag{20}$$

where $b \geq d^{q+1}f/dx^{q+1}, \forall x \in [x]$, *will also bound the function* f *and its derivatives up to order* $(q+1)$ *from above on domain* $[x^*, \sup[x]]$.

Proof. Since the value b is always equal or larger than the $(q+1)^{th}$ order derivative of f in domain $[x]$ the following holds

$$\left.\frac{d^q P^q_{x^*}(x)}{dx^q}\right|_x - \left.\frac{d^q P^q_{x^*}(x)}{dx^q}\right|_{x^*} \geq \left.\frac{d^q f(x)}{dx^q}\right|_x - \left.\frac{d^q f(x)}{dx^q}\right|_{x^*} \forall x \in [x^*, \sup[x]]. \tag{21}$$

Since it is also guaranteed that

$$\left.\frac{d^i P^q_{x^*}(x)}{dx^i}\right|_{x^*} \geq \left.\frac{d^i f(x)}{dx^i}\right|_{x^*}, \forall i \in [0, q], \tag{22}$$

(directly derived from the properties of $P^d_{x_0}(x)$), the following must hold:

$$\left.\frac{d^i P^q_{x^*}(x)}{dx^i}\right|_x \geq \left.\frac{d^i f(x)}{dx^i}\right|_x, \forall i \in [0, q+1], \forall x \in [x^*, \sup[x]]. \tag{23}$$

This completes the proof. □

The bounding of function f from above is guaranteed for any $x \in [x]$ when a VTS is used. By definition, the bounds obtained for the i^{th} order derivative of f via the VTS are guaranteed:

$$\left.\frac{d^i f}{dx^i}\right|_x \subseteq \left[\left.\frac{d^i VTS^d_f}{dx^i}\right|_x\right] \; \forall x \in [x], i \leq d+1. \tag{24}$$

This means that when one uses the upper bound of a VTS of order d to define the polynomial $P^d_{x_0}(x)$ and using $b = \sup[f^{(q+1)}([x])]$, all conditions of Proposition 3.2 are satisfied. Proposition 3.2 can be used to define a Taylor polynomial of order $q \leq d$ which will always bound the function f from above (and its derivatives up to order $q+1$). The new Taylor polynomial again satisfies the conditions of the proposition. This means that one can keep on switching to lower order Taylor polynomials when desired without sacrificing the guarantee of bounding function f from above.

Proposition 3.2 can be used to define a PIF which has Taylor polynomials for each domain in x which do not violate any of the derivative bounds (up to the order of the active polynomial (q)). The idea is that the order of the Taylor polynomial in a given sub-domain x (e.g., one part of the PIF formed by Proposition 3.1) is reduced to q if the $(q+1)^{th}$ order derivative of the polynomial crossed the $\sup[f^{(q+1)}([x])]$ bound. By altering the polynomial to a lower order, the inclusion function is guaranteed not to violate that derivative bound. As an example consider the function $f = \cos(x\pi)$ again for VTS up to order 4. When Proposition 3.1 and Proposition 3.2 are applied a PIF can be derived which has guaranteed equal or sharper bounds than any VTS. The result is represented in Figure 4, and the PIF description is given below (indication of the polynomial order used per sub-domain of x):

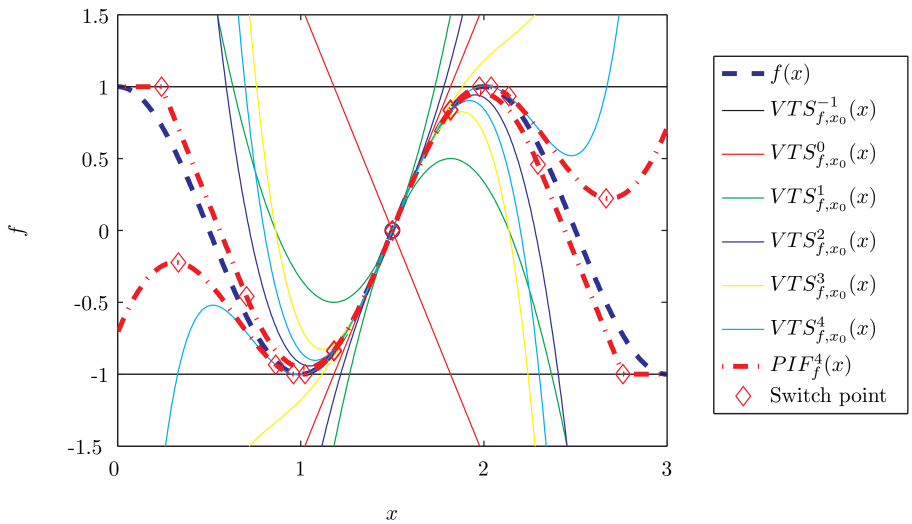

Figure 4: Verified Taylor Series(VTSs) and Polynomial Inclusion Function (PIF) for function $f = cos(x\pi)$ for domain $X = [0, 3]$ and expansion point $x_0 = 1.5$. The PIF is formed by applying proposition 3.1 and 3.2. At a switch point the polynomial of the PIF changes order.

$P^4_{\bar{f}}$

$x \in$	degree of polynomial
$[0.000, 0.240]$	0
$[0.240, 0.704]$	1
$[0.704, 1.182]$	2
$[1.182, 1.500]$	3
$[1.500, 1.818]$	5
$[1.818, 1.976]$	4
$[1.976, 2.039]$	0
$[2.039, 2.137]$	5
$[2.137, 2.667]$	3
$[2.667, 3.000]$	2

$P^4_{\underline{f}}$

$x \in$	degree of polynomial
$[0.000, 0.333]$	2
$[0.333, 0.863]$	3
$[0.863, 0.961]$	5
$[0.961, 1.024]$	0
$[1.024, 1.182]$	4
$[1.182, 1.500]$	5
$[1.500, 1.818]$	3
$[1.818, 2.296]$	2
$[2.296, 2.760]$	1
$[2.760, 3.000]$	0

Comparing to the result given in Figure 3, one can see that the effect of Proposition 3.2 can be severe. The obtained inclusion is much sharper than the original VTSs based on the same set of information.

3.3 Transitions Between Taylor Polynomials

Although the derived PIF is already an improvement, there still remains one issue: the transition between Taylor polynomials in the PIF. As an example, consider the two zoom plots of Figure 4 shown in Figure 5. Clearly the transition in both plots is discontinuous for derivative order higher than 1. A jump in the n^{th} order derivative means that the $(n+1)^{th}$ order derivative does not exist. The latter is not possible for the underlying function f since it is guaranteed that the derivatives remain within the

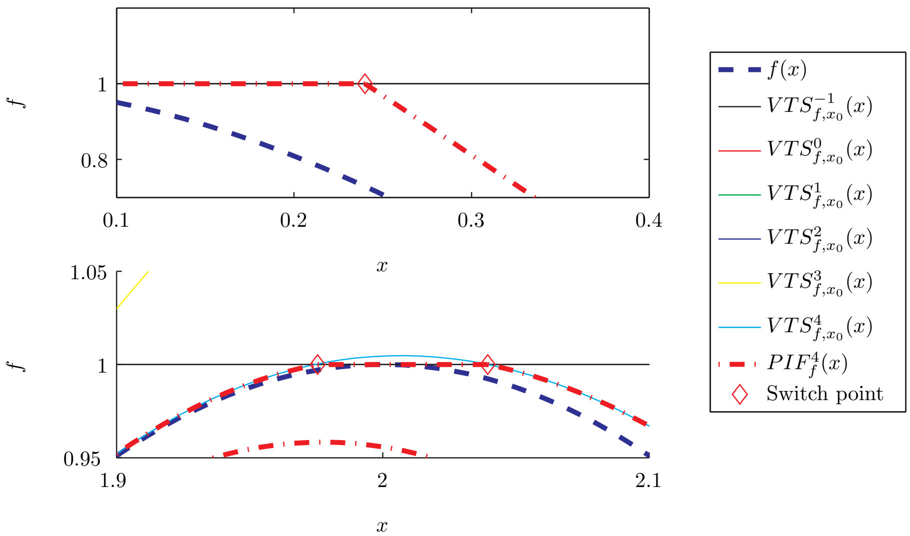

Figure 5: Zoom plots of the result given in figure 4. The figure demonstrates the discontinuous behavior of the PIF in the first order derivative between polynomials.

bounds specified by the natural inclusion function $[f^{(n)}([x])]$. This information can be used to sharpen the bounds on f as is demonstrated next.

Consider a location x^* for which the function f is at its maximum. Per definition the first order derivative of f must be zero, and the second order derivative ≤ 0. Nothing can be said of the values of the higher order derivatives since it depends on the value of the second order derivative (e.g., if it is zero then the 3^{rd} order derivative must be ≤ 0 and so on). The bounds on f given by $[f([x])]$ are guaranteed thus for any point on this bound the previous conditions of the first and second order derivative must hold. Now consider the PIF represented in Figures 4 and 5 at location $x = 0.240$ where the transition is made from a zero order Taylor polynomial to a first order Taylor polynomial. Suppose that a point $x_1 \in [0.000, 0.240]$ exists where f is equal to the active Taylor polynomial $P^{-1}_{\bar{f}}$, and a point $x_2 \in [0.240, 0.704]$ exists where f is equal to the active Taylor polynomial $P^{0}_{\bar{f}}$ on that domain (see Figure 6). If a second order polynomial is used to define the transition between the two points then the following must hold:

$$\left.\begin{array}{ll} p(x_1) & = P^{-1}_{\bar{f}}\,(x_1) \\ \frac{\partial p}{\partial x}\Big|_{x_1} & = 0 \\ p(x_2) & = P^{0}_{\bar{f}}\,(x_2) \\ \frac{\partial p}{\partial x}\Big|_{x_2} & = \left.\frac{\partial P^{0}_{\bar{f}}}{\partial x}\right|_{x=x_2} \\ \left.\frac{\partial^2 p}{\partial x^2}\right|_{x\in X} & = a \end{array}\right\} \begin{array}{ll} p\,(x) & = P^{-1}_{\bar{f}}\,(x_1) + \frac{1}{2}a\,(x - x_1)^2 \\ x_2 & = x_1 + \frac{\frac{\partial P^{0}_{\bar{f}}}{\partial x}}{a} \\ \Delta x & = \frac{\frac{\partial P^{0}_{\bar{f}}}{\partial x}}{a} \end{array}, \tag{25}$$

where a is the constant second order derivative value to be determined. To form a guaranteed inclusion of the function f the value of Δx must be as small as possible

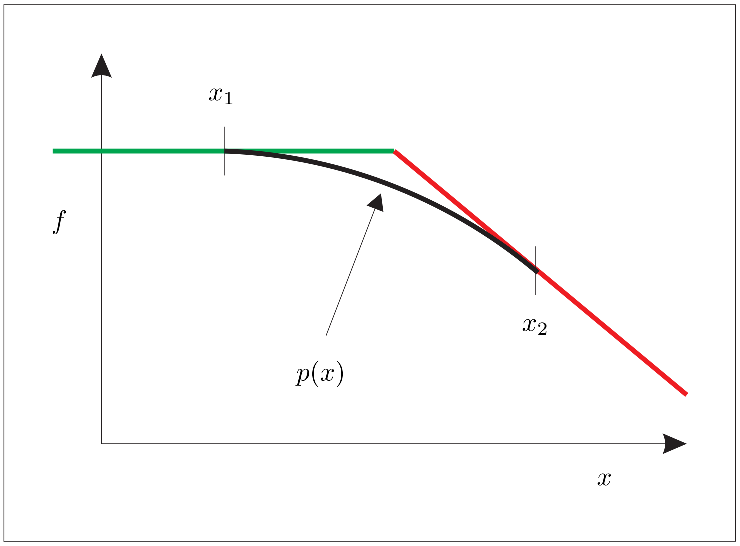

Figure 6: Transition polynomial between Taylor polynomials based on minimal second order derivative information.

which is obtained as follows:

$$\begin{array}{ll} \frac{\partial P_{\hat{f}}^0}{\partial x} > 0 & \rightarrow a = \sup\left[\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=[x]} \right] \\ \frac{\partial P_{\hat{f}}^0}{\partial x} < 0 & \rightarrow a = \inf\left[\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=[x]} \right] \end{array} . \tag{26}$$

By using the bounds obtained by the natural inclusion function the transition is guaranteed to be a valid upper bound on the function f. The demonstrated principle of transition between Taylor polynomial holds for any transition between Taylor polynomial. However, the procedure for defining the transition polynomial becomes less trivial for higher order transitions (i.e., p has a higher degree). As an example, the previously determined second order polynomial still violates the third order derivative bound when the transition is made between p and the Taylor polynomials at x_1 and x_2 (jump in second order derivative) which is not valid. Including higher order derivative information will complicate the derivation, but sharper inclusion functions can be formed.

The research on the smooth transitions between polynomials in the PIF is subject of ongoing research. This includes the analysis of speed versus tightness trade-offs. The main aspects have been highlighted here for completeness but will not be included in the next section dealing with performance evaluation of the PIF.

4 Comparing Methods

To determine the possible improvement of the bounds formed by the PIF compared to the standard Verified Taylor Series (VTS), several test cases are used:

- $f_1(x) = \cos(2x\pi)$
- $f_2(x) = \exp(-2\pi x^2)$
- $f_3(x) = \sum_{i=0}^{n} \sin(x\pi \frac{i!}{2^i} + \frac{\pi}{i!}), n = 5$

for $x \in [0, 1]$ (see Figure 7). For all test cases the volume V between the upper and lower bounds right of the expansion point is computed for the

- VTS up to order d,
- PIF formed using only Proposition 3.1 (PIF_1),
- PIF formed by using Proposition 3.1 and 3.2 (PIF_2).

To see the effect of the width of the domain (VTS accuracy increases with a decrease in domain width $[x]$) all test cases are evaluated for several domains with varying diameters right of the expansion point $x_0 = 0$, i.e., $[x] = x_0 + w([x]) \cdot [0, 1]$ (where $w([x])$ denotes the width of the domain $[x]$). The first test function is a simple function to demonstrate the basic properties of the VTS and the PIF. The other two functions are selected to demonstrate the performance in case of 'non-vanishing' dependency effects, i.e., functions leading to remainder blowup in case of Taylor Models. The results for test function f_1, f_2, and f_3 are given in Figures 8, 9, and 10, respectively.

The first conclusion is that the VTSs are indeed effective for 'small' domains $[x]$ ('small' is problem dependent). The effect of remainder blowup is clearly visible in all figures. One can see that for a given $w([x])$ the inclusion might become worse with increasing VTS order before decreasing again. This behavior is clearly not present for both PIF_1 and PIF_2. For the PIF the accuracy always increases with increase order, i.e., increasing available information. Moreover, it is clearly visible that the PIF always outperforms the VTS, in particular for 'larger' domains. Looking at the figures where the performance of PIF_2 is compared to that of PIF_1 one can clearly see that a huge improvement can be realized ($> 50\%$, expressed as a percentage of the $V(PIF_1)$) especially in the 'middle' values of $w([x])$. For larger values of $w([x])$ the effect in % drops since the volume $V(PIF_1)$ also becomes larger but the in terms of magnitude the gain is increasing. From all figures it becomes clear that the three methods: VTS, PIF_1, and PIF_2, perform almost equal for small $w([x])$ and large d. This is due to the fact that the PIF consist of a single polynomial (the optimal VTS) for the largest part of the domain. The reason that the number of switch point, i.e., number of polynomials, in the PIF_2 is not 1 is that at the end of the domain the switch to lower order polynomials is made. Since this only happens at the very end of the domain, the increase in efficiency is minimal.

The main conclusion is that the performance of the PIF_2 compared to the VTS and PIF_1 increases with increasing function complexity. Due to dependency effects the bounds on the derivatives are widened leading to non-tight bounds which affect the performance of the VTSs considerably. To attain the same accuracy as for simpler functions the model order must be increased considerably especially for larger domains. The latter is far less severe for the proposed PIF. Since higher order methods are most useful for complex functions (for less complex functions standard linear methods would suffice) the introduction of the PIF is a valuable addition to the field of inclusion functions.

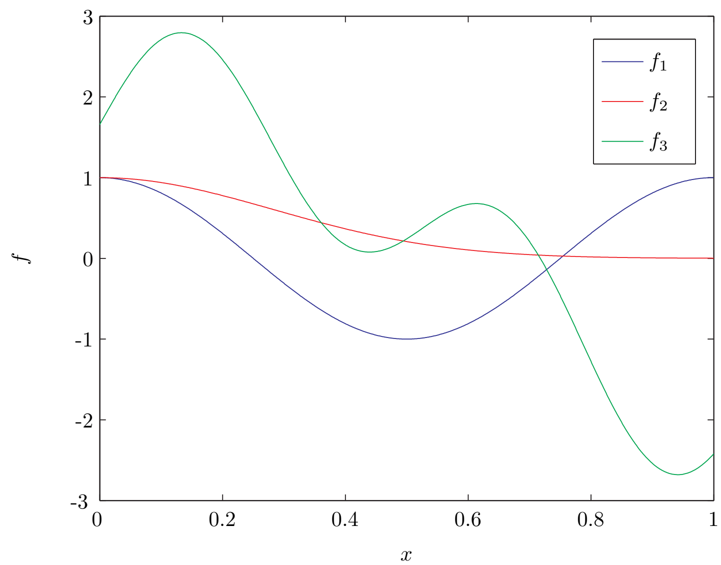

Figure 7: Test function used for the comparison of Taylor Models and Polynomial Inclusion Function performance.

5 Computational Cost Analysis

The main message in this paper is that more efficient handling of available information is possible, i.e., a tighter inclusion function can be obtained without additional function evaluations. Although this is the central topic, the overall computational load required to make the PIF should also be considered. One can argue that the additional computational load, required to derive the PIF, could also have been used to determine a second VTS on a sub-domain thereby improving the sharpness of the inclusion. In this section a cost analysis is made to identify the cases where it is more beneficial to use the PIF than the VTS and vice versa. The performance of both methods is expressed in terms of obtained inclusion function sharpness within the same amount of computation time.

The method of deriving the PIF requires the use of a root finding algorithm. The computational load for solving a one-dimensional root finding problem is relatively low but one can argue that the accumulated computational load can be high. To make a fair comparison between methods, both the PIF and the VTS implementation must be optimized. For the VTS only two function calls must be made, one at the expansion point and one for the entire domain. All the required information is deduced using automatic differentiation. The more complex the function, the longer it takes to derive the DAG required for deriving the derivative information. The computational load of the PIF consists of that of the VTS and additional load to apply the propositions given in this paper.

To eliminate the need for a numeric root finding algorithm (for higher order polynomials), a 'light' version of the PIF construction algorithm is made: Algorithm 1 given in Table 1. This algorithm, which is based on Proposition 3.2 only, was found to yield similar results as the one using both propositions (compare Figure 10 with Figure 11). Note that the guarantee that a tighter enclosure is found compared to the VTS is lost when only using Proposition 3.2[5]. The computational load required for executing Algorithm 1 is very low since only first order polynomial root finding must be performed. Algorithm 1 is be compared to the VTS for the function $f_3(x)$.

The computational load of the derivation of the VTS (t_{VTS}) and the computational load of the required additional computations to derive the PIF ($t_{\Delta PIF}$) are shown in Figure 11 ($t_{PIF} = t_{VTS} + t_{\Delta PIF}$). As one can see, the added computational load to derive the PIF is always lower than that required for deriving a VTS ($\pm$ half for this example). This means that at most two VTSs can be determined in the time it takes to derive one PIF. In Figures 12 and 13 the comparison is made between the inclusion performance in case of using two VTSs (derived for domain $[\inf[x], \mathrm{mid}[\mathrm{x}]]$ and $[\mathrm{mid}[\mathrm{x}], \sup[\mathrm{x}]]$ respectively - expansion point at the beginning of each domain) and the performance of the PIF derived using Algorithm 1 for the entire domain. Test function f_3 is used with once $n = 5$ and once $n = 6$. The results demonstrate that indeed applying two VTSs instead of one yields better results. This is clearly demonstrated when only the zero order derivative bounds are used, i.e., $d = -1$. The performance of the VTS is always better than that of the PIF. From Figures 12 and 13 one can conclude that the performance of the PIF with respect to the VTSs is problem dependent. The more complex the function, the better the performance of the PIF in terms of $V(VTS)$. The same holds for the width of the domain which is investigated. As also the results in the previous section show, it is problem dependent which approach yields the most optimal overall performance in terms of inclusion bounds for a given amount of computation time. If the domain of interest is small, or the function is smooth, there is no difference between $V(VTS)$ and $V(PIF)$ since both will consists of exactly the same polynomial. Therefore the VTS performs better in terms of the computational load. For larger domains and more non-linear functions the PIF quickly becomes more suitable to use, both in terms of sharpness of the bounds and in terms of computational load.

6 n-Dimensional Case

The discussion of the PIF has been restricted to one-dimensional functions. To be easily applicable to higher dimensional functions, it is proposed to apply the procedure given in this paper to all dimensions independently.
A higher dimensional Taylor series is given by:

$$T^{(n)}_{f,\mathbf{x}_0}(\mathbf{x}) = \begin{array}{l} \displaystyle\sum_{0 \le i_1 + \ldots + i_d \le n} \left\{ \frac{\partial^{i_1+\cdots+i_d} f(\mathbf{x})}{\partial x_1^{i_1} \ldots \partial x_d^{i_d}} \cdot \frac{(x_1 - x_{0,1})^{i_1} \cdot \ldots \cdot (x_d - x_{0,d})^{i_d}}{i_1! \cdot \ldots \cdot i_d!} \right\} \\ + L^n_{f,\mathbf{x}_0}(\xi, \mathbf{x}) \end{array} \quad (27)$$

[5]In the performed research the enclosure was always found to be sharper than that of the VTS (see also Figure 11).

Table 1: Algorithm 1: Determination of PIF_f^d.

0.	Determine DAG for expansion point x_0 and domain $[x]$ yielding $[f^d(x)]\forall d$ for x_0 and $[x]$.
1.	Determine upper bound $Pr_{\bar{f}}^d$ for domain $[x_0, \sup[x]]$ using algorithm 2
2a.	Apply transformation on y-axis: $[f^d(x)] = -[f^d(x)], \forall d$ for both x_0 and $[x]$
2b.	Determine upper bound $P_{\bar{f}}^d$ for domain $[x_0, \sup[x]]$ using algorithm 2
2c.	Apply transformation on y-axis on P: $P_{\bar{f}}^d \rightarrow Pr_{\underline{f}}^d$
3a.	Apply transformation on x-axis: $[f^d(x)] = -[f^d(x)], \forall uneven d$ for both x_0 and $[x]$
3b.	Determine upper bound $P_{\bar{f}}^d$ for domain $[\inf[x], x_0]$ using algorithm 2
3c.	Apply transformation on y-axis and x-axis on P: $P_{\bar{f}}^d \rightarrow Pl_{\underline{f}}^d$
4a.	Apply transformation on y-axis: $[f^d(x)] = -[f^d(x)], \forall d$ for both x_0 and $[x]$
4b.	Determine upper bound $P_{\bar{f}}^d$ for domain $[\inf[x], x_0]$ using algorithm 2
4c.	Apply transformation on x-axis on P: $P_{\bar{f}}^d \rightarrow Pl_{\bar{f}}^d$
5	Create full PIF: $PIF_f^d(x) = [P_{\underline{f}}^d, P_{\bar{f}}^d]\forall x \in [x]$ $P_{\underline{f}}^d = [Pl_{\underline{f}}^d, Pr_{\underline{f}}^d], P_{\bar{f}}^d] = [Pl_{\bar{f}}^d, Pr_{\bar{f}}^d]$

Table 2: Algorithm 2: Determination of the upper bound piecewise Taylor polynomial $P_{\bar{f}}^d$.

0.	Initialize $P_{\bar{f},x_0^*}^d = \sup[VTS_f^d]$ with $x_0^* = x_0$, set $q = d$, exit = false.
1.	Evaluate $\sup[VTS_f^d(\sup[x])]\forall d$ and set $\hat{d}$ such that: $\sup[VTS_f^{\hat{d}}(\sup[x])] \leq \sup[VTS_f^d(\sup[x])]\forall d$
WHILE !exit AND $q > \hat{d}$	
2.	Determine location x^* where $P_{\bar{f},x_0^*}^q = \sup[f^q]$
3.	IF $x^* <= \sup[x]$ THEN compute $P_{\bar{f},x^*}^{q-1}$ using Proposition 3.2 ELSE exit = true.
4.	Check derivative values: IF $\frac{\partial^i P_{\bar{f}}^{q-1}}{\partial x^i} > \sup[f^i([x])]$ for any $i \in [1, q-\hat{d}]$ THEN execute algorithm 3 to determine i^* and Δx^* .
5.	Set $x^* = x^* + \Delta x^*$ and compute $P_{\bar{f},x^*}^{i^*}$ using Proposition 3.2.
6.	Add $P_{\bar{f},x_0^*}^q$ as part of the piece-wise polynomial and set $q = i^*, x_0^* = x^*$
END	

Table 3: Algorithm 3: Check derivative bounds.

0.	Set $\Delta_i = 0,\ \forall i \in [1, q-\hat{d}]$.
FOR $i \in [1, q-\hat{d}]$:	
1.	Set $\epsilon = -\infty$
	WHILE $\epsilon < -0.01(\sup[x] - x_0)$
2.	$\epsilon = \sup[f^i([x])] - \partial^i P_{\bar{f}}^{q-1}/\partial x^i\|_{x=x^*+\Delta_i}$
3.	IF $\sup[f^{i+1}([x])] \leq 0$ THEN set $\epsilon = 0, \Delta x_i = x_0 - x^*$
4.	ELSE Compute $\Delta x_i = \epsilon / \sup[f^{i+1}([x])]$
	END
5.	Define i^* such that $\Delta x_{i^*} < \Delta x_i, \forall\ i \in [1, q-\hat{d}]$
	Define $\Delta x^* = \Delta x_{i^*}$
END	

The terms corresponding solely to one dimension can be taken out:

$$
\begin{aligned}
T^n_{f,\mathbf{x}_0}(\mathbf{x}) &= \sum_d \left\{ T^{(n)}_{f,x_{0,d}}(x_d) + L^n_{f,x_{0,d}}(\xi_d, x_d) \right\} + R \\
T^n_{f,x_{0,d}}(x_d) &= \sum_{0 \leq i_d \leq n} \left\{ \frac{\partial^{i_d} f(x_d)}{\partial x_d^{i_d}} \cdot \frac{(x_d - x_{0,d})^{i_d}}{i_d!} \right\} \\
L^n_{f,x_{0,d}}(\xi_d, x_d) &= \frac{1}{(n+1)!} \left. \frac{\partial^{n+1} f}{\partial x_d^{n+1}} \right|_{\xi_d} (x_d - x_{0,d})^{n+1}
\end{aligned}
\quad , \tag{28}
$$

where R contain all the cross terms. By applying the PIF method to all $(T^{(n)}_{f,x_{0,d}}(x_d), L^n_{f,x_{0,d}}(\xi_d, x_d))$ independently, the method given in this paper can be applied to higher dimensions.

7 Conclusions

A novel inclusion function has been introduced called the Polynomial Inclusion Function (PIF). It has been proven and shown through several examples that the PIF is guaranteed to provide equal or sharper bounds that any (combination of) Taylor Models and Verified Taylor Series without the need for additional information, i.e. function evaluations. The assumption hereby is that the required derivative information is derived using automatic differentiation.

Irrespectively of the assumption made, the accuracy of the PIF always improves with increasing order, a trait that Taylor Models or Verified Taylor Series do not possess. This means that the PIF does not suffer from the inclusion function blowup effect encountered with Taylor Models (remainder blowup) for highly non-linear functions (non-vanishing derivatives) and/or wider domains. The PIF has been deduced for the one-dimensional case and can be easily extended to n-dimensional functions.

In this paper the PIF has been compared to the VTS inclusion function. In future research the PIF will be compared to TMs which have been constructed using TM arithmetic. It has been shown that given a TM and a VTS, there exist a domain $[x_0 - \delta_1, x_0 + \delta_2], \delta_1 \geq 0, \delta_2 \geq 0$ for which the VTS (and thus also the PIF) is guaranteed to have tighter enclosure of f than the TM, irrespectively of the method used for deriving the remainder. The fact that the inclusion formed by a PIF start with nearly zero width is an advantage over TM which have a constant width enclosure ($=w(I)$) over the entire domain. Future research will indicate what the value of $\delta_{1,2}$ will be for different types of function and different domains. Moreover, a thorough investigation regarding accuracy versus generation cost must be made to see if the new PIF can be used more efficiently than TM. The TM of Makino and Berz compromise accuracy over speed. If in the time a PIF is derived, multiple TM can be derived, then the overall accuracy of combined TMs might be higher.

Although the PIF provides significantly improved bounds compared to VTS, there still remains room for improvement. Future research will focus on including even more (available) information in the construction of the PIF to obtain even sharper bounds. The results presented in this paper prove that the PIF are a worthy addition to the field of inclusion functions.

Acknowledgements

This work is performed as part of the MicroNED MISAT cluster, project 1-D-1.

References

[1] R G Ayoub. Paolo Ruffini's contributions to the quintic. *Archive for History of Exact Science*, 23:253 – 277, 1980.

[2] M. Berz and G. Hoffstatter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4:83–97, 1998.

[3] M. Berz and K. Makino. COSY INFINITY version 8. *Nuclear Instruments and Methods in Physics Research A*, 427:338 – 343, 1999.

[4] J.T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21:193 – 207, 1998.

[5] E. de Weerdt, Q.P. Chu, and J.A. Mulder. Neural network output optimization using interval analysis. *IEEE Transactions on Neural Networks*, 20(4):638–653, 2009.

[6] Y. Fang. Optimal bicentered form. *Reliable Computing*, 9:291–302, 2004.

[7] Eldon R. Hansen. On solving systems of equations using interval arithmetic. *Mathematics of Computation*, 22(102):374–384, 1968.

[8] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis.* Springer-Verlag London Berlin Heidelberg, 2001. ISBN 1-85233-219-0.

[9] Y. Lin and M.A. Stadtherr. Validated solution of ODEs with parametric uncertainties. *Appl. Numer. Math.*, 58:1145–1162, 2007.

[10] K. Makino and M. Berz. New applications of Taylor model methods. *Automatic Differentiation of Algorithms: From Simulation To Optimization*, pages 359 – 364, 2000.

[11] K. Makino and M. Berz. Higher order verified inclusions of multidimensional systems by Taylor Models. *Nonlinear Analysis*, 47:3503–3514, 2001.

[12] K. Makino and M. Berz. Taylor models and other validated functional inclusion functions. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003.

[13] K. Makino and M. Berz. Higher order multivariate automatic differentiation and validated computation of remainder bounds. *Transactions on Computers*, 4:1611–1618, November 2005.

[14] W.R. Mekwi. *Iterative Methods for Roots of Polynomials.* PhD thesis, Exeter College, University of Oxford, 2001.

[15] N.S. Nedialkov. Interval Tools for ODEs and DAEs. In *12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics,*. IEEE Conference Publications, 2006.

[16] M. Neher, K.R. Jackson, and N.S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45:236–262, 2007.

[17] A. Neumaier. Taylor forms - use and limits. *Reliable Computing*, 9:43–79, 2002.

[18] Hermann Schichl and Arnold Neumaier. Interval analysis on directed acyclic graphs for global optimization. *Journal of Global Optimization*, 33(4):541–562, 2005.

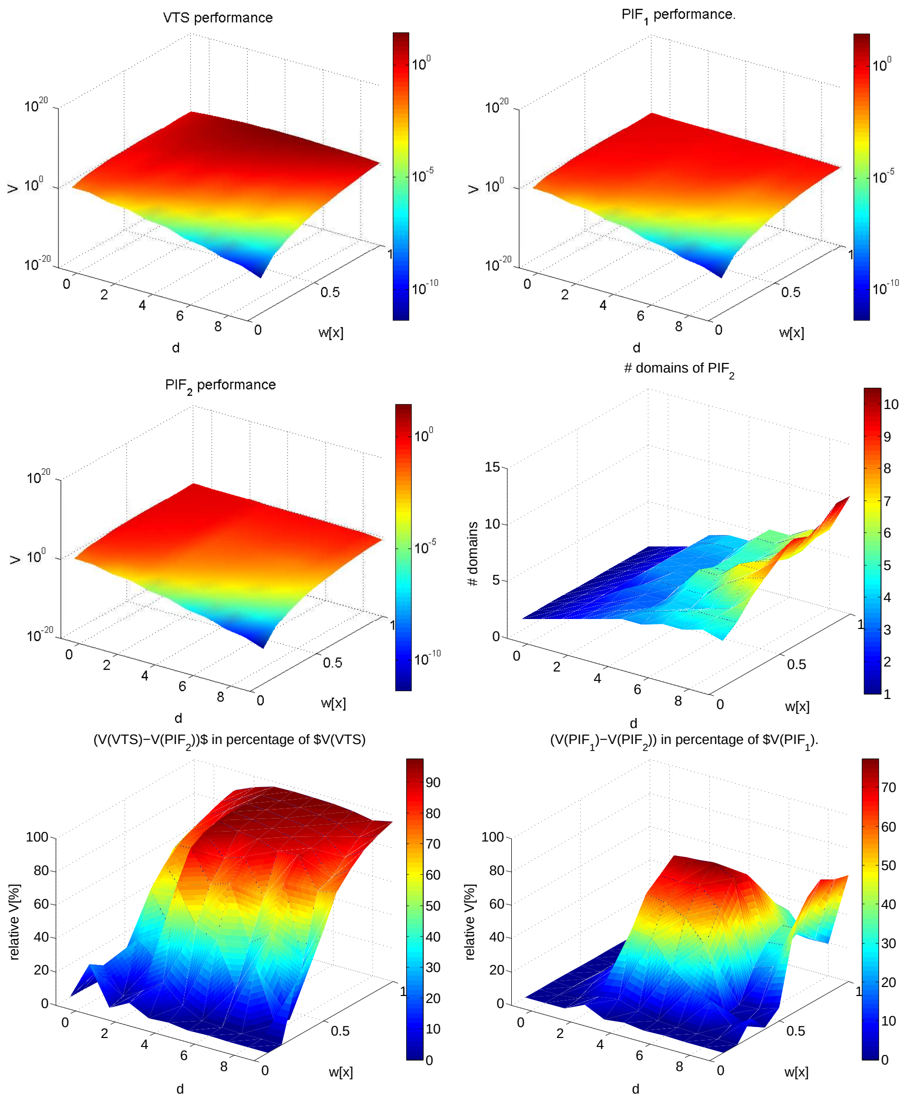

Figure 8: Test case 1 ($f_1(x) = \cos(2x\pi)$) - Performance evaluation of the PIF compared to VTS. PIF_1 is created using Proposition 3.1 only, while PIF_2 is created using Proposition 3.2 too. V denotes the area between the bounds provide by the inclusion function.

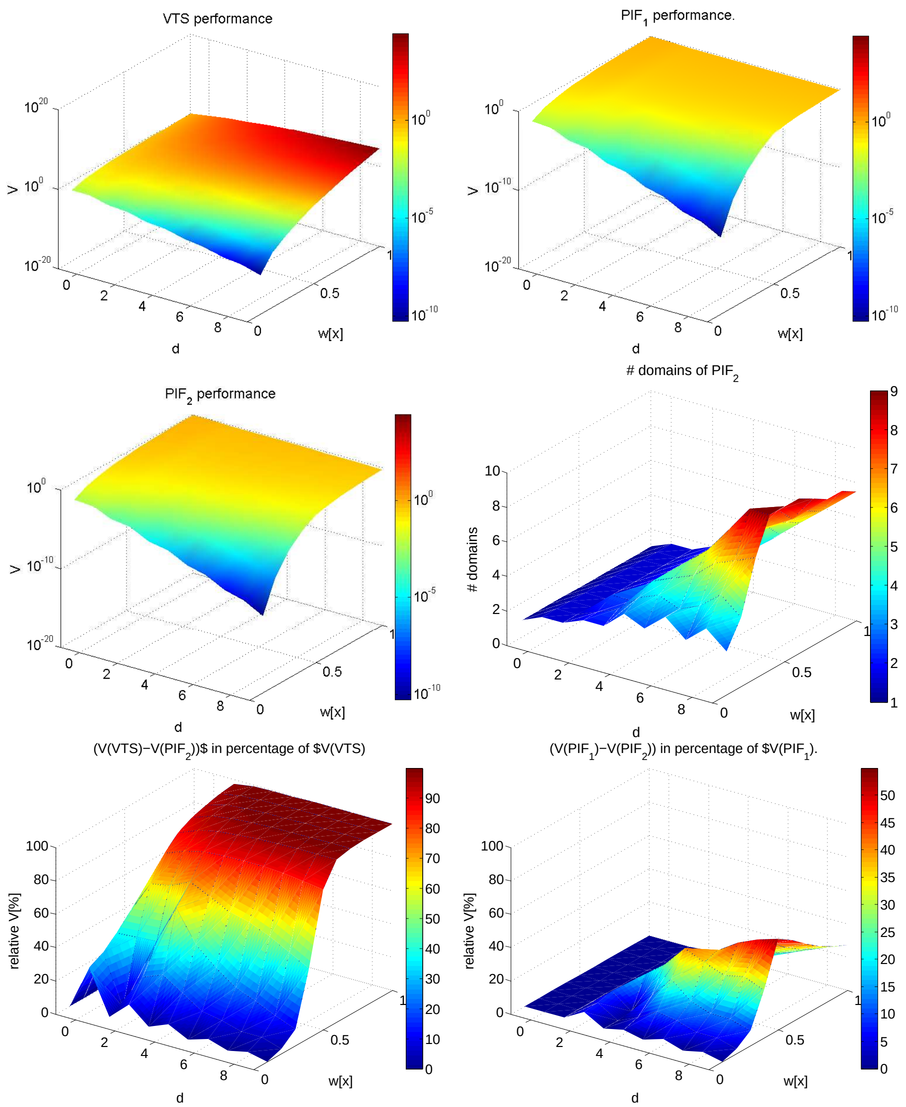

Figure 9: Test case 2 ($f_2(x) = \exp(-2\pi x^2)$) - Performance evaluation of the PIF compared to VTS. PIF_1 is created using Proposition 3.1 only, while PIF_2 is created using Proposition 3.2 too. V denotes the area between the bounds provide by the inclusion function.

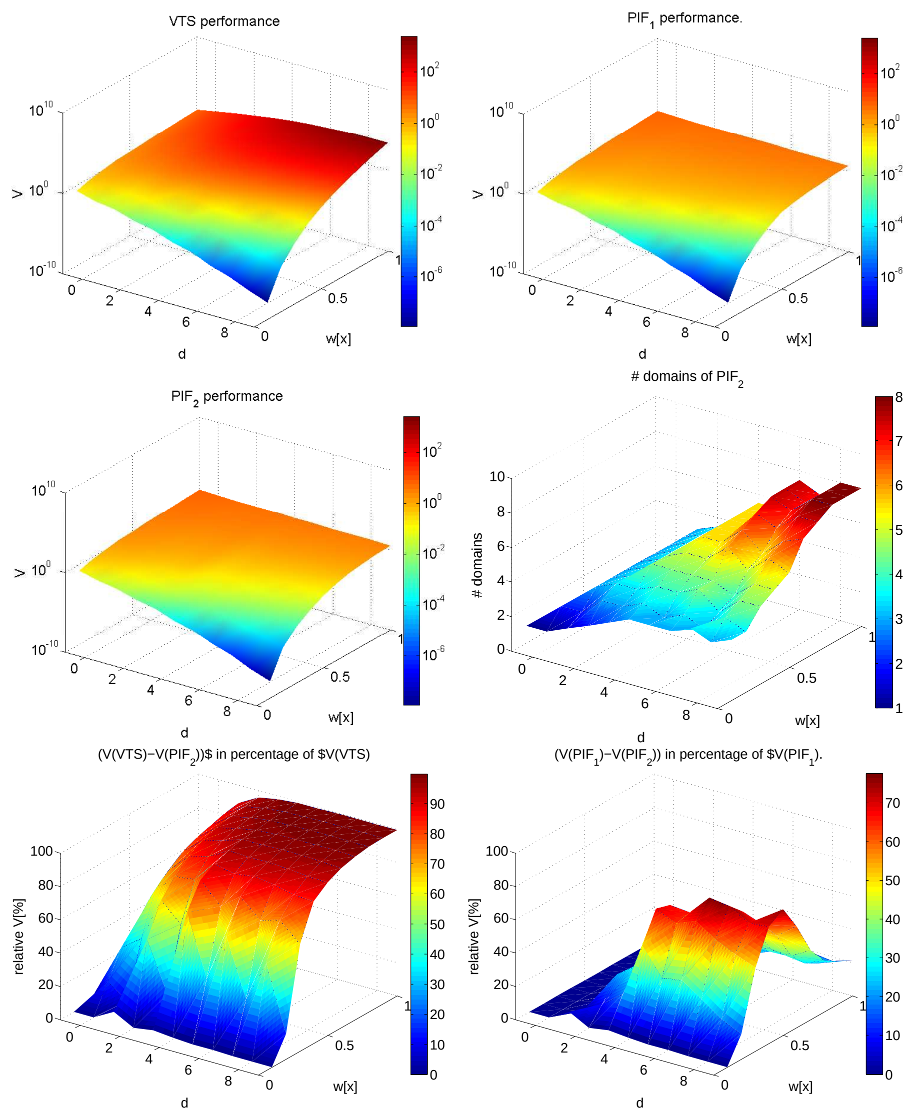

Figure 10: Test case 3 ($f_3(x) = \sum_{i=0}^{n} \sin(x\pi \frac{i!}{2^i} + \frac{\pi}{i!}), n = 5$) - Performance evaluation of the PIF compared to VTS. PIF_1 is created using Proposition 3.1 only, while PIF_2 is created using Proposition 3.2 too. V denotes the area between the bounds provide by the inclusion function.

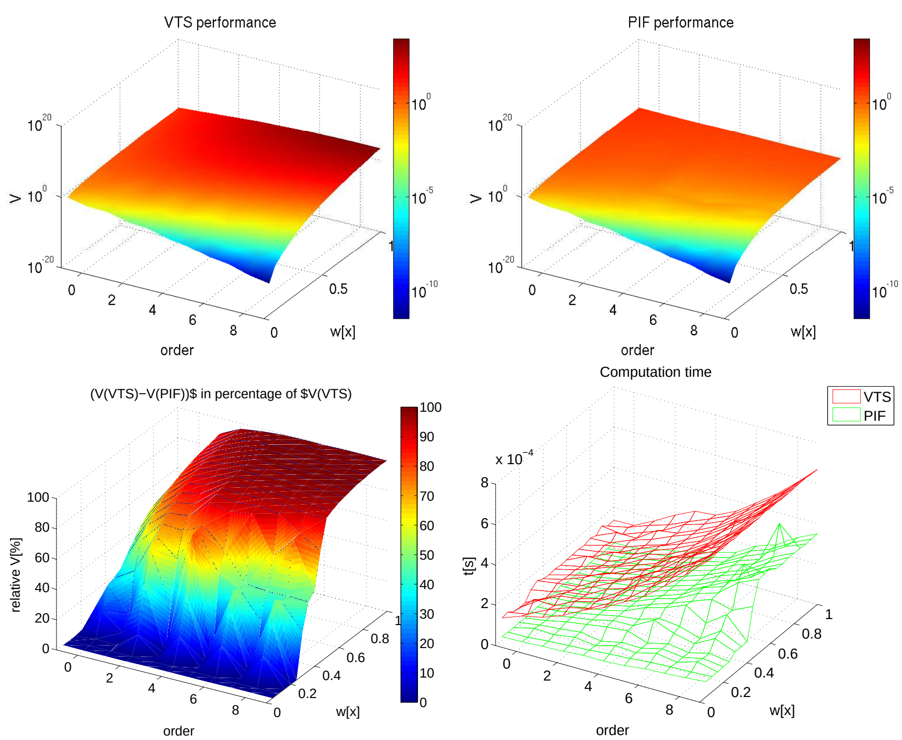

Figure 11: Test case 3 ($f_3(x) = \sum_{i=0}^{n} \sin(x\pi \frac{i!}{2^i} + \frac{\pi}{i!}), n = 5$) - Performance evaluation of the PIF (Algorithm 1) compared to VTS. V denotes the area between the bounds provide by the inclusion function.

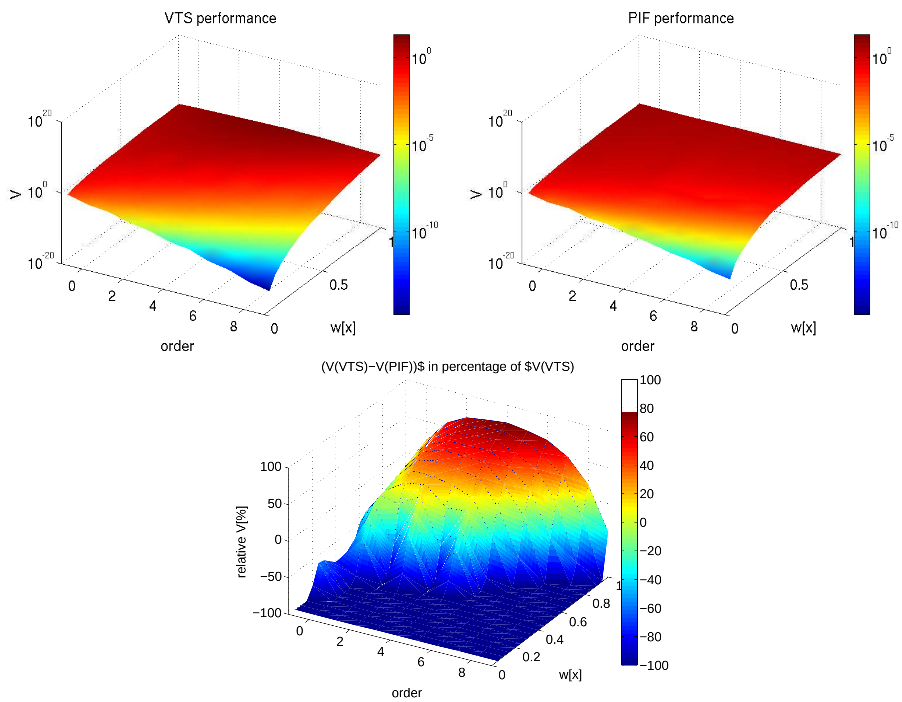

Figure 12: Test case 3 ($f_3(x) = \sum_{i=0}^{n} \sin(x\pi\frac{i!}{2^i} + \frac{\pi}{i!}), n = 5$) - Performance evaluation of the PIF (Algorithm 1) compared to two VTSs. One VTS is used to form an inclusion of domain $[\inf[x], \mathrm{mid[x]}]$, while the other is used for domain $[\mathrm{mid[x]}, \mathrm{sup[x]}]$. V denotes the area between the bounds provide by the inclusion function. For the bottom plot, values lower than -100% have been set to -100%.

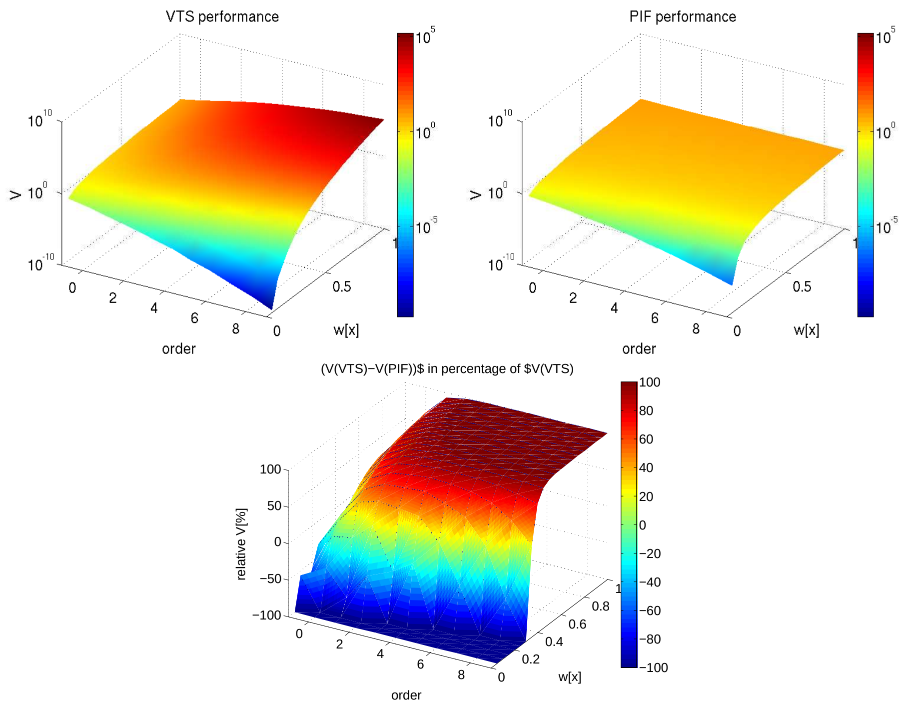

Figure 13: Test case 3 ($f_3(x) = \sum_{i=0}^{n} \sin(x\pi\frac{i!}{2^i} + \frac{\pi}{i!}), n = 6$) - Performance evaluation of the PIF (Algorithm 1) compared to two VTSs. One VTS is used to form an inclusion of domain $[\inf[x], \mathrm{mid}[\mathrm{x}]]$, while the other is used for domain $[\mathrm{mid}[\mathrm{x}], \sup[\mathrm{x}]]$. V denotes the area between the bounds provide by the inclusion function. For the bottom plot, values lower than -100% have been set to -100%.