



TOM STREUBEL¹, CAREN TISCHENDORF, ANDREAS
GRIEWANK²

Piecewise Polynomial Taylor Expansions – The Generalization of Faà di Bruno’s Formula

¹  0000-0003-4917-7977

²  0000-0001-9839-1473

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Piecewise Polynomial Taylor Expansions – The Generalization of Faà di Bruno’s Formula

Tom Streubel and Caren Tischendorf and Andreas Griewank

Abstract We present an extension of Taylor’s theorem towards nonsmooth evaluation procedures incorporating absolute value operations. Evaluation procedures are computer programs of mathematical functions in closed form expression and allow a different treatment of smooth operations and calls to the absolute value function. The well known *classical* Theorem of Taylor defines polynomial approximation of sufficiently smooth functions and is widely used for the derivation and analysis of numerical integrators for systems of ordinary differential or differential algebraic equations, for the construction of solvers for the continuous nonlinear optimization of finite dimensional objective functions and for root solving of nonlinear systems of equations. The herein provided proof is constructive and allows efficiently designed algorithms for the execution and computation of generalized piecewise polynomial expansions. As a demonstration we will derive a k -step method on the basis of polynomial interpolation and the proposed generalized expansions.

Key words: generalized Taylor expansion, implicit generation of splines, nonsmooth integration of differential algebraic equations (DAE and ODE), multistep methods, generalized Hermite interpolation, algorithmic piecewise differentiation (AD and APD), evaluation procedures, treating absolute values (abs, max and min)

Tom Streubel
Zuse Institute Berlin and Humboldt University of Berlin, Germany, e-mail: streubel@zib.de

Caren Tischendorf
Humboldt University of Berlin, Germany

Andreas Griewank
School of Mathematical Sciences and Information Technology, Yachaytech, Ecuador

1 Introduction, Preliminaries and Notions

In [6] a piecewise linear generalization of first order Taylor expansions as alternative but also related to Bouligand-subdifferentials has been introduced and analyzed in detail. Furthermore several efficient approaches for numerical integration, root solving and optimization have been proposed. The concept originally evolved among many researchers from several countries with different backgrounds and from different areas within the field of applied mathematics to solve practical nonsmooth problems arising from industry, finance and economy. Many subsequent papers have been published, e.g. regarding numerical integration [10], [1], [22], root solving [11], [9], [21], [19], [18], [2] and optimization [8], [12], [5], [4], [14], [3]. All of them deal with the construction of new methods, their analysis and/or promising numerical experiments. This work derives and proves the existence of higher order piecewise polynomial expansions of certain non-smooth functions. In the spirit of [6] this works continuous and extends the original concept. The first order expansion in our sense coincides with the algorithmic piecewise linearization. In advance we will derive a class of higher order k -step methods based on the generalized Taylor expansion as a demonstration. Even though possible applications could be formulated and presented for all areas mentioned above we will focus on the simulation of nonsmooth differential algebraic equations. For the purpose of introducing new notions we will firstly formulate the *classical* Theorem of Taylor before its generalization.

Lemma 1 (Theorem of Taylor). *Let $f \in \mathcal{C}^{d,1}(\mathbb{R}^n, \mathbb{R})$ be a sufficiently smooth function, $\hat{x} \in \mathbb{R}^n$ a reference point and $\Delta x = x - \hat{x} \in \mathbb{R}^n$ an offset in the domain of f . Then the following recursive statement holds true:*

$$\forall 1 \leq m \leq d : f(\hat{x} + \Delta x) - f(\hat{x}) = \sum_{i=1}^m \Delta^{(i)} f(\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1}), \quad (1)$$

where $\Delta^{(i)} f(\hat{x}; \Delta x) = \sum_{|\delta|=i}^{\max} \left[\frac{\partial^\delta f(\hat{x})}{\partial x^\delta} \right] \cdot \Delta x^\delta$ is the sum of all mixed partial derivatives of order i in direction Δx .

The Theorem of Taylor defines polynomial approximations of sufficiently smooth functions. However many practical problems and most algorithms are not smooth everywhere. Instead evaluations of

$$\begin{aligned} \max(a, b) &= (a + b + \text{abs}(b - a))/2 \quad \text{and} \\ \min(a, b) &= (a + b - \text{abs}(b - a))/2 \end{aligned}$$

or of the absolute value function are necessary. With this in mind consider some piecewise smooth function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ in the sense of [20] that doesn't necessarily satisfy the prerequisites of theorem 1. We then call an ordered family of $d \geq 1$ Lipschitz-continuous functions $\mathbf{f} \equiv [f(\hat{x}), \Delta^{(1)} f(\hat{x}; \Delta), \dots, \Delta^{(d)} f(\hat{x}; \Delta)]$ a *general-*

ized Taylor expansion of order d of f on a nonempty open set $\mathcal{D} \subseteq \mathbb{R}^n$ if \mathbf{f} satisfies statement (1) for any $\hat{x} \in \mathcal{D}$ and any $\Delta x \in \mathcal{D} - \hat{x}$ anyway.

The i th element $\Delta^{(i)}f(\hat{x}; \Delta x)$ of \mathbf{f} is called *increment* of order i . Of course increments can't be defined via partial derivatives of f in general, since they might not exist. For the sake of convenience we define $\Delta^{(0)}f(\hat{x}; \Delta x) \equiv f(\hat{x})$ as point evaluation and we'll use the notational abbreviation $\Delta^{(i)}f \equiv \Delta^{(i)}f(\hat{x}; \Delta x)$ when the arguments are clear within the context. We will give algorithmic schemes for the evaluation of generalized Taylor expansions of piecewise smooth computer programs and prove the claimed approximation quality (1) for them. We will refer the computer programs as *composite piecewise smooth evaluation procedures* or short evaluation procedures. The next sections are organized as follows. In section 2 the concept of composite piecewise smooth evaluation procedures will be introduced and a propagation scheme for the algorithmic evaluation of their generalized Taylor expansions will be provided. In the subsequent section 3 the formula of Faà di Bruno will be proven in the context of this generalization. In section 4 a integrator for differential algebraic equations in semi explicit fashion will be derived on the basis of the proposed generalized Taylor expansions. The derived method is closely related to linear k -step methods such as Adams-Moulton (sometimes referred to as implicit Adams) or BDF methods (see e.g. [13]).

2 Propagation-Scheme of Expansions for Non-Smooth Evaluation Procedures

In this section an algorithm for the point evaluations of the generalized Taylor expansions will be provided. An evaluation procedure is a finite composition of so called unary $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ and binary $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ elementary operations, which are aggregated as a library $\Phi_{\text{abs}} = \Phi \cup \{\text{abs}\}$ in their symbolic form and thus make up the atomic constituents of complex and possibly vector-valued functions. Despite the absolute value function as the only exception any other elementary operation has to be at least d -times Lipschitz-continuously differentiable. This assumption is called elementary differentiability (ED). This means any evaluation procedure consisting solely of operations from Φ , which excludes the absolute value function, inherit their order d differentiability by chain rule. In our framework any unary operation complying to (ED) can be added to Φ by the user. But for the time being we want to restrict the selection of binary operations to $\{+, -, \cdot, /\}$, this is sum, difference, product and division. The dependencies among operations within some evaluation procedure define a partial ordering which is called data dependence relation. This relation corresponds to a directed acyclic graph or evaluation graph of the procedure.

Example 1 (evaluation procedure, elementary code instruction, evaluation graph).

Consider the following example evaluation procedure:

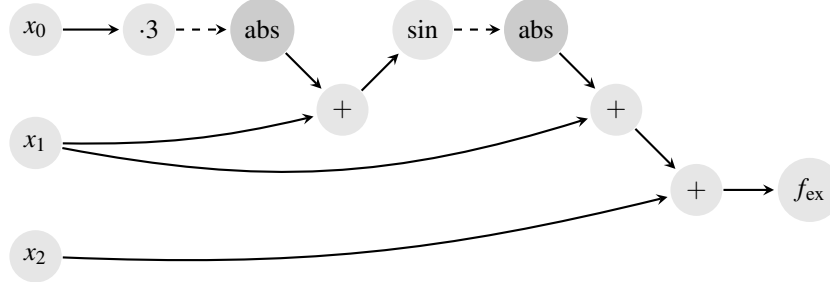
$$f_{\text{ex}}(x_0, x_1, x_2) = x_2 + x_1 + |\sin(x_1 + |3 \cdot x_0|)|$$

Ex 1.a – closed form expression

which maps \mathbb{R}^3 to \mathbb{R} . The acyclic directed evaluation graph is depicted below and is not an expression tree due to the 2 arcs leaving node x_1 . Each node defines a subgraph by the closure of its dependencies and thus can be interpreted as partial evaluation function f_{ex} up to this node within the full graph.

$v_0 \equiv x_0$
$z_0 = v_1 \equiv 3 \cdot v_0$
$v_2 = \text{abs}(z_0)$
$v_3 \equiv x_1$
$v_4 \equiv v_3 + v_2$
$z_1 = v_5 \equiv \sin(v_4)$
$v_6 \equiv \text{abs}(z_1)$
$v_7 \equiv v_3 + v_6$
$v_8 \equiv x_2$
$v_9 \equiv v_8 + v_7$
$f_{\text{ex}}(x_0, x_1, x_2) \equiv v_9$

Ex 1.b – elementary instructions



Ex 1.c – evaluation graph

Furthermore the closed form expression corresponds uniquely to an ordered list of elementary instructions. The total order match one possible extension of the partial dependency order. The left-hand-sides (LHS) within the elementary code instructions are intermediate variables of f_{ex} . Those of them which are arguments of absolute value operations deserve special attention because their signs might be flipped during an evaluation for example. They will be referred as *switching variables* throughout the text.

So any evaluation procedure within our scope can be expressed as ordered lists of elementary instructions and a data dependence relation denoted by \prec and thus any intermediate variable v_i for some interger $i \in \mathbb{N}$ matches either one of the cases described by table 1. Variable initialisations $v_i \equiv x_j$ can be interpreted as identity operation $v_i = \text{id}(x_j)$ applied to some scalar-component x_j of the input variable vector to fit into the scheme of table 1. We will use the notion $f \in \text{span}(\Phi_{\text{abs}})$ whenever a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ has at least one representation as evaluation procedure of elementary operations from Φ_{abs} .

Many properties and its computation of evaluation procedures can be carried out by induction through the elementary instructions w.r.t. its dependency ordering. This

$v_i \equiv \varphi(u_i),$	$\varphi \in \Phi_1$	where $u_i = v_j$ and $j \prec i$
$v_i \equiv \psi(u_i, w_i),$	$\psi \in \{+, -, \cdot, /\}$	where $u_i = v_j, w_i = v_k$ and $(j, k) \prec i$
$v_i \equiv \text{abs}(z_i)$		where $z_i = v_j$ and $j \prec i$

Table 1 generic table for elementary instructions and $\Phi_1 \subseteq \Phi$ being the sub-library of all unary operations $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ that comply to assumption (ED)

process is called *propagation*. For instance the Lipschitz-continuity of all elementary operations and repetitive application of the chain-rule implies the Lipschitz-continuity of all evaluation procedures $f \in \text{span}(\Phi_{\text{abs}})$ by propagation.

With the concept of propagation at hand we can implicitly define more complex processes and algorithms by imposing propagation rules to all occurring cases of table 1. For the application of operator overloading we will extend the interpretation of intermediate variables. From now on they consist of a scalar intermediate value $\hat{v}_i = v_i(\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{n-1}) \in \mathbb{R}$ and an ordered list $[\Delta^{(0)}v_i, \Delta^{(1)}v_i, \dots, \Delta^{(d)}v_i] \in \mathbb{R}^{d+1}$ of increments representing the point evaluation of the generalized Taylor expansion of the of the partial evaluation function up to v_i within its evaluation procedure. The propagation rules for the primal evaluation and for the expansions are defined as follows:

variable and constant initialization

Let $\gamma, \hat{x}, \Delta x \in \mathbb{R}$ and consider $v = x$ as well as $\tilde{v} = \gamma$, then the propagation rules are defined as follows:

$$\begin{aligned} v(\hat{x}) &= \hat{x}, & \Delta^{(1)}v &= \Delta x & \text{and} & \quad \forall 2 \leq m \leq d : \Delta^{(m)}v = 0, \\ \tilde{v}(\hat{x}) &= \gamma, & \forall 1 \leq m \leq d : \Delta^{(m)}\tilde{v} &= 0. \end{aligned}$$

sum, difference and linearity

Let $\alpha, \beta \in \mathbb{R}$, $\hat{x}, \Delta x \in \mathbb{R}^n$ and u, w two other intermediate variables. Consider $v = \alpha \cdot u \pm \beta \cdot w$, then $v(\hat{x}) = \alpha \cdot \hat{u} \pm \beta \cdot \hat{w}$ and the increment propagation is:

$$\forall 1 \leq m \leq d : \Delta^{(m)}v = \alpha \cdot \Delta^{(m)}u \pm \beta \cdot \Delta^{(m)}w.$$

product

Let u, w be some intermediate variables, $\hat{x}, \Delta x \in \mathbb{R}^n$ and consider $v = u \cdot w$, then the primal evaluation is $v(\hat{x}) = \hat{u} \cdot \hat{w}$ and the increment operation is given by:

$$\forall 1 \leq m \leq d : \Delta^{(m)}v = \sum_{i=0}^m \Delta^{(i)}u \cdot \Delta^{(m-i)}w$$

division

Let u, w be some intermediate variables, $\hat{x}, \Delta x \in \mathbb{R}^n$ and consider $v = \frac{u}{w}$, then the

primal evaluation is $v(\hat{x}) = \frac{\hat{u}}{\hat{w}}$ and the increment operation reads as follows:

$$\forall 1 \leq m \leq d : \Delta^{(m)}v = \frac{1}{\hat{w}} \left[\Delta^{(m)}u - \sum_{i=0}^{m-1} \Delta^{(i)}v \cdot \Delta^{(m-i)}w \right]$$

smooth unary operation

Let u be some intermediate variable, $\hat{x}, \Delta x \in \mathbb{R}^n$ and consider $v = \varphi(u)$ for some elementary operation $\varphi \in \Phi_1$ complying with assumption (ED). The primal evaluation is carried by the execution of the symbolic instruction $v(\hat{x}) = \varphi(\hat{u})$.

The formula of *Faà die Bruno* can be utilized for the formal definition of the propagation step:

$$\forall 1 \leq m \leq d : \Delta^{(m)}v = \sum_{(k_1, \dots, k_m) \in T_m} \varphi^{(k_1 + \dots + k_m)}(\hat{u}) \cdot \prod_{i=1}^m \frac{1}{k_i!} \cdot \Delta^{(i)}u^{k_i}, \quad (2)$$

where $(k_1, \dots, k_m) \in T_m \iff o(k_1, \dots, k_m) \equiv \sum_{i=1}^m i \cdot k_i = m$ and $\varphi^{(k_1 + \dots + k_m)}$ the $(k_1 + \dots + k_m)$ th derivative of φ .

But note that for the propagation of frequently used functions (such as sin, cos, exp, log, ... and other smooth operations from the standard `cmath`-library of `c++`) specialized formulas have been developed for the smooth algorithmic propagation of Taylor polynomial expansions. These formulas can be found e.g. in [7] or in [17] and are more efficient w.r.t. run-time and memory. Furthermore they are equivalent to the application of (2), but still need to be slightly adjusted in accordance to the incremental notion we use here.

So far the propagation rules fully comply with standard Taylor arithmetics. Thus the generalized Taylor expansion becomes a polynomial Taylor expansion in the sense of definition 1 for evaluation procedures $f \in \text{span}(\Phi) = \text{span}(\Phi_{\text{abs}} \setminus \{\text{abs}\})$ that fully comply with assumption (ED) in that all elementary instructions of f do. But now the absolute value operation is finally the last addition to our listing and the seed for nonsmoothness in evaluation procedures.

absolute value

Let u be some intermediate variable and consider $v = \text{abs}(u)$, then $v(\hat{x}) = |\hat{u}|$ and the propagation rule is defined recursively:

$$\forall 1 \leq m \leq d : \sum_{i=0}^m \Delta^{(i)}v = \left| \sum_{i=0}^m \Delta^{(i)}u \right| \implies \Delta^{(m)}v \equiv \left| \sum_{i=0}^m \Delta^{(i)}u \right| - \sum_{i=0}^{m-1} \Delta^{(i)}v.$$

We want to conclude this section with an example expansion of some nonsmooth evaluation procedure.

Example 2. Consider a nonlinear, nonsmooth evaluation procedure $\mathcal{F} \in \text{span}(\Phi_{\text{abs}})$, with $\mathcal{F}(x, y) = |\exp(x) - |y||$. There are 3 sets of non-differentiabilities:

$M_0 = \{(x, \exp(x)) \mid x \in \mathbb{R}\}$, $M_1 = \{(x, -\exp(x)) \mid x \in \mathbb{R}\}$ and $M_2 = \{(x, 0) \mid x \in \mathbb{R}\}$.

Plots of the function \mathcal{F} and of its generalized expansions $\mathcal{T}_{\mathcal{F}}^{(i)}$ ranging from order 1 (or piecewise linearization) up to order 5 can be found in figure 1.

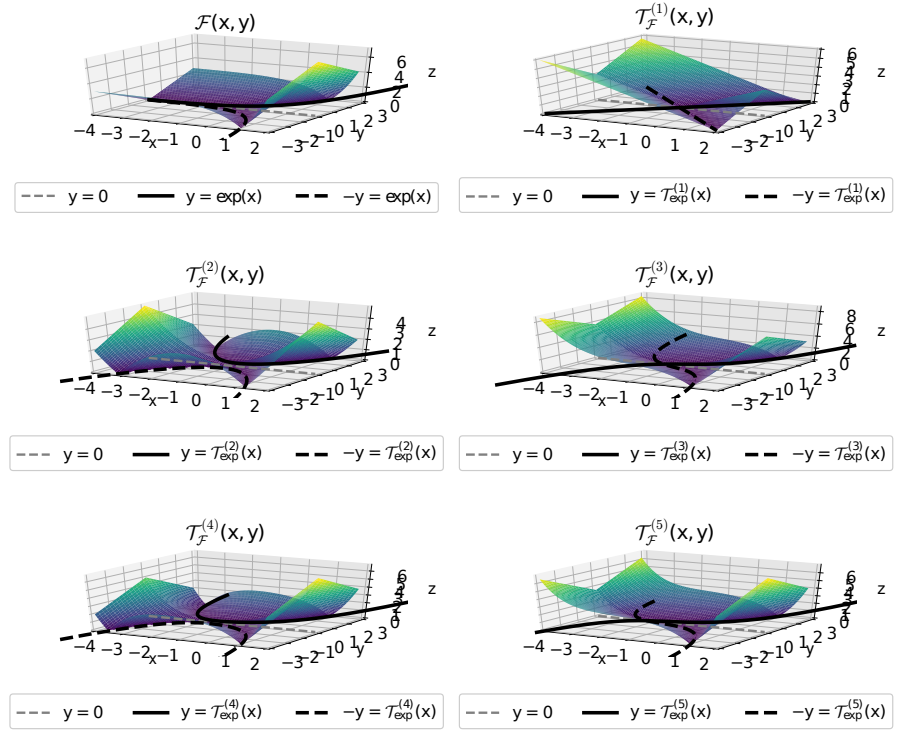


Fig. 1 plots of \mathcal{F} and its expansions $\mathcal{T}_{\mathcal{F}}^{(i)}$ ranging from order 1 to order 5

The solid and dashed lines in figure 1 in the x - y -planes are Taylor polynomial expansions of $y = \pm \exp(x)$ and $y = 0$. The manifolds of non-differentiabilities of the generalized expansions coincide with those of the polynomial expansions.

3 The Generalization of Taylor's Theorem

In this section we will prove that the propagation rules given in section 2 generate an approximating expansion in the sense of equation (1). Due to the concept of propagation introduced in the same section we only have to give a proof for any elementary instruction of table 1.

However most of the binary operations can be represented in terms of unary operations. The propagation by linear combinations $v = \alpha \cdot u \pm \beta \cdot w$, for u, w intermediate variables with generalized Taylor expansions (induction hypothesis) and $\alpha, \beta \in \mathbb{R}$ combines the sum $v = u + w$, the difference $v = u - w$ and unary scalar multiplication $v = \alpha \cdot u$ into one rule. The scalar multiplication alike the variable and constant initialization can be proven straight forward. The difference can be expressed in terms of a sum followed by a scalar multiplication by -1 as $v = u + w = u + ((-1) \cdot w)$.

For the sum $v = u + w$ we can deduce from the generalized expansions of u and w :

$$\begin{aligned} \forall 1 \leq m \leq d : v(\hat{x} + \Delta x) - \hat{v} &= u(\hat{x} + \Delta x) - \hat{u} + w(\hat{x} + \Delta x) - \hat{w} \\ &= \left[\sum_{i=1}^m \Delta^{(i)} u + \mathcal{O}(\|\Delta x\|^{m+1}) \right] + \left[\sum_{i=1}^m \Delta^{(i)} w + \mathcal{O}(\|\Delta x\|^{m+1}) \right] \\ &= \sum_{i=1}^m \Delta^{(i)} v + 2\mathcal{O}(\|\Delta x\|^{m+1}), \text{ where } \Delta^{(i)} v = \Delta^{(i)} u + \Delta^{(i)} w \end{aligned}$$

The division can be expressed in terms of a multiplication and an univariate inversion $\text{inv}(w) = w^{-1}$ as $v = \frac{u}{w} = u \cdot \text{inv}(w)$.

The multiplication can be represented by the so called *Apollonius* identity:

$$v = u \cdot w = \frac{1}{4} [(u + w)^2 - (u - w)^2]$$

an by that we've discussed any binary operation already. For the absolute value we can also immediately deduce:

$$\begin{aligned} \forall 1 \leq m \leq d : v(\hat{x} + \Delta x) &= |u(\hat{x} + \Delta x)| \\ &= \left| u(\hat{x}) + \sum_{i=1}^m \Delta^{(i)} u + \mathcal{O}(\|\Delta x\|^{m+1}) \right| = \left| u(\hat{x}) + \sum_{i=1}^m \Delta^{(i)} u \right| + \mathcal{O}(\|\Delta x\|^{m+1}) \\ &= v(\hat{x}) + \sum_{i=1}^m \Delta^{(i)} v + \mathcal{O}(\|\Delta x\|^{m+1}), \text{ where } \Delta^{(i)} v = \left| \sum_{j=0}^i \Delta^{(j)} u \right| - \sum_{j=0}^{i-1} \Delta^{(j)} v \end{aligned}$$

Finally unary operations are left. Corresponding to the propagation rules defined in section 2 it is sufficient to prove that Faà di Bruno's formula still applies in the context of piecewise polynomial expansions. To that end the following identity will be useful and is carried out by applying the *multinomial expansion* twice. Suppose values $a_1, a_2, \dots, a_i, b \in \mathbb{R}$ and let $A \equiv \sum_{j=1}^i a_j$, then

$$\begin{aligned}
\left(\sum_{j=1}^i a_j + b\right)^i &= (A + b)^i = \sum_{l=0}^i \binom{i}{l} A^l b^{i-l} = \sum_{l=0}^i \binom{i}{l} \left[\sum_{j=1}^i a_j\right]^l b^{i-l} \\
&= \sum_{l=0}^i \frac{i!}{l!} \left[\sum_{k_1+\dots+k_l=l} \binom{l}{(k_1, \dots, k_l)} \prod_{j=1}^l a_j^{k_j} \right] \frac{b^{i-l}}{(i-l)!} \\
&= \sum_{l=0}^i i! \left(\sum_{k_1+\dots+k_l=l} \prod_{j=1}^l \frac{a_j^{k_j}}{k_j!} \right) \frac{b^{i-l}}{(i-l)!}. \tag{3}
\end{aligned}$$

Theorem 1 (formula of Faà di Bruno).

Let u be some intermediate variable with a generalized Taylor expansion at $\hat{x} \in \mathbb{R}^n$, i.e. there is and an ordered list $(\Delta^{(i)}u(\hat{x}; \Delta x))_{i=0}^d$ of Lipschitz-continuous functions:

$$u : \{\hat{u} \equiv \Delta^{(0)}u, [\Delta^{(1)}u, \Delta^{(2)}u, \dots, \Delta^{(d)}u]\}$$

that satisfies the following statement (which is identical to equation (1)):

$$\forall 1 \leq m \leq d : u(\hat{x} + \Delta x) - \hat{u} = \sum_{i=1}^m \Delta^{(i)}u(\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1}). \tag{4}$$

Furthermore let $\varphi \in \Phi_1$ be some unary function $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, which complies to assumption (ED), i.e. it is d -times Lipschitz-continuously differentiable. Note that φ has a polynomial Taylor expansion:

$$\varphi(\mu) = \varphi(\hat{\mu}) + \sum_{i=1}^m \frac{\varphi^{(i)}(\hat{\mu})}{i!} (\mu - \hat{\mu})^i + \mathcal{O}(\|\mu - \hat{\mu}\|^{m+1}).$$

Then the composition $v = \varphi(u)$ has a generalized Taylor expansion, i.e. an ordered list of Lipschitz-continuous functions generated by the formula of Faà di Bruno:

$$\forall 1 \leq m \leq d : \Delta^{(m)}v(\hat{x}; \Delta x) = \sum_{(k_1, \dots, k_m) \in T_m} \varphi^{(k_1+\dots+k_m)}(\hat{u}) \cdot \prod_{i=1}^m \frac{1}{k_i!} \cdot \Delta^{(i)}u^{k_i},$$

such that the same statement holds true for v :

$$\forall 1 \leq m \leq d : v(\hat{x} + \Delta x) - v(\hat{x}) = \sum_{i=1}^m \Delta^{(i)}v(\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1}). \tag{5}$$

Proof. This proof is similar to a proof of the polynomial of Faà di Bruno's formula and can be found as Theorem 2.3 in [16].

Let $1 \leq m \leq d$, due to the sufficient differentiability of $\varphi \in \Phi_1$ a Taylor polynomial expansion of φ exists. Also u is assumed to have a generalized Taylor expansion, i.e.:

$$\varphi(\mu) = \varphi(\hat{\mu}) + \sum_{i=1}^m \frac{\varphi^{(i)}(\hat{\mu})}{i!} (\mu - \hat{\mu})^i + \mathcal{O}(\|\mu - \hat{\mu}\|^{m+1}), \quad (6)$$

$$\text{as well as: } u(\hat{x} + \Delta x) = u(\hat{x}) + \sum_{i=1}^m \Delta^{(i)} u(\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1}). \quad (7)$$

By choosing $\mu \equiv u(\hat{x} + \Delta x)$ and $\hat{\mu} = \hat{u}$ equation (7) can be substituted into (6):

$$\begin{aligned} & \varphi(u(\hat{x} + \Delta x)) - \varphi(\hat{u}) \\ &= \sum_{i=1}^m \frac{\varphi^{(i)}(\hat{u})}{i!} \left[\sum_{k=1}^m \Delta^{(k)} u + \mathcal{O}(\|\Delta x\|^{m+1}) \right]^i + \mathcal{O}(\|u(\hat{x} + \Delta x) - \hat{u}\|^{m+1}) \end{aligned} \quad (8)$$

and applying identity (3) to equation (8) the latter extends to:

$$= \sum_{i=1}^m \varphi^{(i)}(\hat{u}) \left[\sum_{l=0}^i \left(\sum_{k_1 + \dots + k_l = l} \prod_{j=1}^l \frac{\Delta^{(j)} u^{k_j}}{k_j!} \right) \frac{\mathcal{O}(\|\Delta x\|^{m+1})^{i-l}}{(i-l)!} \right] + \mathcal{O}(\|\Delta x\|^{m+1}). \quad (9)$$

Consider the term within outer brackets in line (9) closer for which we can deduce:

$$\forall l < i: \quad \left(\sum_{k_1 + \dots + k_l = l} \prod_{j=1}^l \frac{\Delta^{(j)} u^{k_j}}{k_j!} \right) \frac{\mathcal{O}(\|\Delta x\|^{m+1})^{i-l}}{(i-l)!} = \mathcal{O}(\|\Delta x\|^{m+1}) \quad (10)$$

since for the summands within the round brackets in line (10) holds true:

$$o(k_1, \dots, k_i) \equiv \sum_{j=1}^i j \cdot k_j \geq m+1 \quad \implies \quad \prod_{j=1}^i \frac{\Delta^{(j)} u^{k_j}}{k_j!} = \mathcal{O}(\|\Delta x\|^{m+1}).$$

The last deduction is a consequence of equation (4). and so equation (9) becomes:

$$\varphi(u(\hat{x} + \Delta x)) - \varphi(\hat{u}) = \sum_{i=1}^m \sum_{\substack{o(k_1, \dots, k_i) \leq m \\ k_1 + \dots + k_i = i}} \varphi^{(i)}(\hat{u}) \prod_{j=1}^i \frac{\Delta^{(j)} u^{k_j}}{k_j!} + \mathcal{O}(\|\Delta x\|^{m+1}) \quad (11)$$

$$= \sum_{i=1}^m \sum_{(k_1, \dots, k_i) \in T_i} \varphi^{(k_1 + \dots + k_i)}(\hat{u}) \prod_{j=1}^i \frac{\Delta^{(j)} u^{k_j}}{k_j!} + \mathcal{O}(\|\Delta x\|^{m+1}) \quad (12)$$

$$= \sum_{i=1}^m \Delta^{(i)} [\varphi \circ u](\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1}) \quad (13)$$

For the transformations from line (11) to (12) so following identities have been used:

$$\begin{aligned} k_1 + \dots + k_i = i & \implies o(k_1, \dots, k_i) \geq i \\ o(k_1, \dots, k_i) = i+1 & \implies (k_1, \dots, k_i, 0) \in T_{i+1}. \end{aligned}$$

All together the recursive property (5) holds true for the composition $v = \varphi(u)$

$$\forall m \leq n : \varphi(u(\hat{x} + \Delta x)) - \varphi(\hat{u}) = \sum_{i=1}^m \Delta^{(i)} [\varphi \circ u](\hat{x}; \Delta x) + \mathcal{O}(\|\Delta x\|^{m+1})$$

and this completes the proof. \square

4 A Generalized Integrator for Semi Explicit DAEs

In this section an implicit k -step integration method for differential equations will be derived, using a combined approach of polynomial interpolation similar to Adams-Multon methods and generalized Taylor expansions. First of all a definition of the problem is required and thus we define semi explicit systems of differential algebraic equations similar as in [15].

Definition 2 (partially nonsmooth semi explicit DAE). *A semi explicit system of differential algebraic equations with potentially nonsmooth differential equations (DAE) is a system of equations of the form:*

$$\dot{x}_1(t) \equiv \frac{d}{dt}x_1(t) = f_1(x_1(t), x_2(t), t), \quad (14a)$$

$$0 = f_2(x_1(t), x_2(t), t), \quad (14b)$$

where $x_1 : [0, T] \rightarrow \mathbb{R}^n$, $x_2 : [0, T] \rightarrow \mathbb{R}^m$, $f_1 : \mathbb{R}^n \times \mathbb{R}^m \times [0, T] \rightarrow \mathbb{R}^n$ the system function of differential equations (14a) and $f_2 : \mathbb{R}^n \times \mathbb{R}^m \times [0, T] \rightarrow \mathbb{R}^m$ the system function of algebraic equations (14b), for some time horizon $T > 0$. In advance let $f_1 \in \text{span}(\Phi_{\text{abs}})$ be a nonsmooth evaluation procedure and $f_2 \in \mathcal{C}^{1,1}$ at least once Lipschitz-continuously differentiable.

Systems such as (14) are mostly not symbolically solvable or the symbolic solution formulas are numerically unstable or the algebraic transformations are very expensive to carry out computationally. E.g. one approach is the initial transformation of a DAE into a system of ordinary differentiable equations. This can be achieved e.g. by a sequential series of differentiations. However symbolic differentiations usually extends the number of symbolic operations of the original algebraic representation exponentially in the number of differentiations necessary (for further details see [7]). Algorithmic differentiation derives functions exactly within machine precision of a computer system and the costs of calculating a directional derivative is bounded linearly by the costs of one single primal function evaluation (for further details see also [7]).

Instead a numeric approximation approach is an alternative. The goal is to generate a table of data points

$t_0 = 0$	$t_1 = h$...	$t_i = i \cdot h$...	$t_{\vartheta} = \vartheta \cdot h$
(x_1^0, x_2^0)	(x_1^1, x_2^1)	...	(x_1^i, x_2^i)	...	$(x_1^{\vartheta}, x_2^{\vartheta})$

where $0 < h \ll T$ and $\vartheta \in \mathbb{N}$ such that $(\vartheta - 1) \cdot h < T \leq \vartheta \cdot h$. The first pair is assumed to be a consistent initial value in that it satisfies $x_1^0 = x_1(0)$ and $x_2^0 = x_2(0)$ for some analytical solution $x_1 : [0, T] \rightarrow \mathbb{R}^n$, $x_2 : [0, T] \rightarrow \mathbb{R}^m$ of system (14). Any other data point $1 \leq i \leq m$ is an approximation $x_1^i \approx x_1(t_i)$ and $x_2^i \approx x_2(t_i)$ of the same solution. However interpolation polynomials of order $k \in \mathbb{N}$ in Newton base representation can be used to interpolate the data on each subinterval $[i \cdot h, (i+1) \cdot h]$ for $k \leq i < \vartheta$:

$$p_l(t_i; t) \equiv \sum_{j=0}^k \frac{\nabla_k^{(j)} x_l}{j!} \prod_{m=0}^j (t - t_{i-m}),$$

where $l \in \{1, 2\}$ and $\nabla^{(j)}$ the backward finite difference operator defined as:

$$\nabla_i^{(0)} x_l \equiv x_l^i, \quad \nabla_i^{(j)} x_l \equiv \frac{\nabla_i^{(j-1)} x_l - \nabla_{i-1}^{(j-1)} x_l}{h}.$$

Using the constant time-discretization scheme and polynomial interpolation consider the integrated differential equations of system (14) on the same subinterval:

$$\begin{aligned} \int_0^h \dot{x}_1(t_i + t) dt &= h \int_0^1 f_1(x_1(t_i + h\tau), x_2(t_i + h\tau), t_i + h\tau) d\tau \\ &\approx h \int_0^1 f_1(p_1(t_{i+1}; t_i + h\tau), p_2(t_{i+1}; t_i + h\tau), t_i + h\tau) d\tau. \end{aligned}$$

Henceforth we use an alias $y(\tau) \equiv f_1(p_1(t_{i+1}; t_i + h\tau), p_2(t_{i+1}; t_i + h\tau), t_i + h\tau)$. Clearly $y_i \in \text{span}(\Phi_{\text{abs}})$ since f_1 , p_1 and p_2 do so as well and thus we can apply a generalized Taylor expansion:

$$\int_0^h \dot{x}_1(t_i + t) dt = x_1(t_i + h) - x_1(t_i) \approx h \int_0^1 y(1) + \sum_{m=1}^k \Delta^{(m)} y(h; \tau) d\tau.$$

The generalized Taylor expansion defines piecewise polynomial approximations of f_1 , which can be integrated exactly within machine precision. This motivates the pseudo-algorithm 1 for a single numerical integration step. Let $\check{x}_1 \equiv x_1^i$ and $\check{x}_2 \equiv x_2^i$ the pseudo-algorithm 1 calculates the first iterates $\hat{x}_1^{(j)}$ and $\hat{x}_2^{(j)}$ of a converging sequence towards x_1^{i+1} and x_2^{i+1} .

Pseudo-Algorithm 1 (generalized order $k+1$ combined Newton-Taylor method).

- guess a starting value $(\hat{x}_1^{(0)}, \hat{x}_2^{(0)}) \in \mathbb{R}^{n+m}$, e.g. with some other predictor method
- iterate over $j = 0, 1, 2, \dots$ and terminate at $J \in \mathbb{N}$ when $\|\hat{x}_l^{(j)} - \hat{x}_l^{(j-1)}\| < \text{tol}_l$, for both $l \in \{1, 2\}$, is met:
 - generate interpolation polynomials $l \in \{1, 2\}$: $p_l^{(j)}$ of $\hat{x}_l^{(j)}, \check{x}_l, x_l^{i-1}, \dots, x_l^{i-k}$
 - define $y_j(\tau) \equiv f_1(p_1^{(j)}(t_{i+1}; t_i + h\tau), p_2^{(j)}(t_{i+1}; t_i + h\tau), t_i + h\tau)$

- calculate generalized quadrature¹: $b_j \equiv h \int_0^1 \sum_{m=1}^k \Delta^{(m)} y_j(h; \tau) d\tau$
- calculate next iterate $(\hat{x}_1^{(j+1)}, \hat{x}_2^{(j+1)}) \in \mathbb{R}^{n+m}$ as result of:

$$\underset{\substack{\hat{x}_1^{(j+1)} \in \mathbb{R}^n \\ \hat{x}_2^{(j+1)} \in \mathbb{R}^m}}{\text{solve}} \begin{bmatrix} \hat{x}_1^{(j+1)} - \check{x}_1 - h \cdot f_1(\hat{x}_1^{(j+1)}, \hat{x}_2^{(j+1)}, t_{i+1}) \\ f_2(\hat{x}_1^{(j+1)}, \hat{x}_2^{(j+1)}, t_{i+1}) \end{bmatrix} = \begin{bmatrix} b_j \\ 0 \end{bmatrix}$$

- set $x_1^{i+1} \equiv \hat{x}_1^{(j)}$, $x_2^{i+1} \equiv \hat{x}_2^{(j)}$ and the numerical integration step is finished.

If the generalized expansion will be replaced by a Taylor polynomial expansion in the sense of definition 1, pseudo-algorithm 1 turns into a linear k -step method in the classical sense as defined e.g. in [13]. The same happens whenever $f_1 \in \text{span}(\Phi) = \text{span}(\Phi_{\text{abs}} \setminus \{\text{abs}\})$ is satisfying assumption (ED), since the generalized and the Taylor polynomial expansion coincide for such functions. On the other hand when the generalized quadrature rule will be substituted by the simplified rule $b_j \equiv 0$ in every step, pseudo-algorithm 1 transforms into the well-known implicit Euler method which is of order 1.

The following conjecture guesses the consistency error for the differential variables x_1 based on observations from different numerical experiments and thus remains unproven for the time being.

Conjecture 1 (error of the generalized order $k+1$ combined Newton-Taylor method). Consider some DAE in the sense of definition 2. Suppose there is an analytical solution of that system in terms of functions $x_1 : [0, T] \rightarrow \mathbb{R}^n$ and $x_2 : [0, T] \rightarrow \mathbb{R}^m$ on the time interval $[0, T]$. Then the consistency error of pseudo-algorithm 1 is:

$$x_1(t_{i+1}) - x_1(t_i) - h \cdot f_1^{(i+1)} = h \left[\int_0^1 \sum_{j=1}^k \Delta^{(j)} \tilde{y}_i(h; \tau) d\tau \right] + \mathcal{O}(|h|^\gamma),$$

where $\tilde{y}_i(\tau) \equiv f_1(\tilde{p}_1(t_{i+1}; t_i + h\tau), \tilde{p}_2(t_{i+1}; t_i + h\tau), t_i + h\tau),$
 $f_1^{(i+1)} \equiv f_1(x_1(t_{i+1}), x_2(t_{i+1}), t_{i+1})$

and \tilde{p}_l are polynomial interpolations of exact data $x_l(t_{i+1}), x_l(t_i), \dots, x_l(t_{i-k})$, for both components $l \in \{1, 2\}$.

The order of the consistency error is $\gamma = \min(v+2, k+2)$, where $0 \leq v \leq \infty$ is the largest integer or infinity such that $y_i \in \mathcal{C}^{v,1}([0, 1], \mathbb{R}^n)$ and $f_2 \in \mathcal{C}^{v,1}(\mathbb{R}^n \times \mathbb{R}^m \times [0, T], \mathbb{R}^m)$. Thus v is the degree of smoothness of $f_1 \in \text{span}(\phi_{\text{abs}})$ inherited by $y_i \in \text{span}(\phi_{\text{abs}})$ and of f_2 . Note k is still the order of interpolation by Newton polynomials and the generalized Taylor expansion.

¹ where $b_j = h \int_0^1 y_j(1) + \sum_{m=1}^k \Delta^{(m)} y_j(h; \tau) d\tau - h f(\hat{x}_1^{(j)}, \hat{x}_2^{(j)}, t_{i+1})$ holds true

5 Conclusion & Outlook

We've derived and presented a Taylor expansion in terms of piecewise polynomials for piecewise smooth functions incorporating absolute value operations in their evaluations graph. In advance we've proven the residuum to be of $\mathcal{O}(\|x - \hat{x}\|^{d+1})$, where x is the point of evaluation, \hat{x} the reference point of expansion and d a user definable order, assuming certain parts of the underlying evaluations procedure to be sufficiently smooth. Applications in optimization, root solving and numerical integration are possible and their derivation and analysis seem to be logical next steps. The presented expansion concept allows the generalization of already existing methods in such a way that the original functionality remains unchanged for elementary differentiable objective functions or systems. A generalized integration method for semi explicit DAEs based on Newton interpolations generalized Taylor expansions was derived for the purpose of demonstration. One concrete next step is the extension of the presented method for semi explicit DAEs with non-smooth algebraic equations $f_2 \in \text{span}(\Phi_{\text{abs}})$ which was excluded from our considerations so far. Furthermore the Newton polynomial interpolation can and should be replaced by some generalized Hermite interpolation scheme. This would also relax the expected order of the consistency error from conjecture 1. Of course the conjecture itself need to be proven as well.

As mentioned in the last paragraph the underlying concept can be adapted for the propagation of Hermite interpolations through an evaluation graph of some non-smooth function f . By doing so we can calculate higher order approximations in one or several reference points $\hat{x}_0, \hat{x}_1, \dots \in \mathbb{R}^n$ satisfying $f(x) - \mathcal{H}_f[\hat{x}_0, \dots](x) = \mathcal{O}(\|x - \hat{x}_0\|^{\theta_0} \cdot \|x - \hat{x}_1\|^{\theta_1} \dots)$, where the exponents $\theta_0, \theta_1, \dots$ are natural numbers and \mathcal{H} denotes such a generalized Hermite interpolation of f .

Another possible starting point is the generalization of the multivariate formula of Faà di Bruno. The restriction onto mostly unary elementary operations would no longer be necessary and allows the generalized Taylor expansion of multivariate functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ in a so called Abs-Normal Form (ANF):

$$\begin{bmatrix} z \\ f(x) \end{bmatrix} = \begin{bmatrix} G(x, |z|) \\ F(x, |z|) \end{bmatrix},$$

where the functions $F : \mathbb{R}^{n+s} \rightarrow \mathbb{R}^n$ and $G : \mathbb{R}^{n+s} \rightarrow \mathbb{R}^s$ are assumed to be sufficiently smooth and the partial derivative matrix $\frac{\partial}{\partial |z|} G(x, |z|)$ is of strictly lower triangular form, everywhere. The latter condition allows the explicit computation of all the switching variables $z = (z_i)_{i=0}^{s-1}$ one by one. More precisely F and G may be black-box functions and not necessarily evaluations procedures in that context. I.e. providing arbitrarily designed computation routines for the point evaluation of F , G and their Taylor polynomial expansions would be sufficient, because the propagation process can be applied on a higher abstraction level by taking components of them as new elementary operations.

Acknowledgements The work for the article has been conducted within the Research Campus MODAL funded by the German Federal Ministry of Education and Research (BMBF) (fund number 05M14ZAM).

References

1. P. Boeck, B. Gompil, A. Griewank, R. Hasenfelder, and N. Strogies. Experiments with generalized midpoint and trapezoidal rules on two nonsmooth ode's. *mongolian mathematical Journal*, 17:39–49, 2013. URL: <http://iom.num.edu.mn/journal/2013/3.pdf>.
2. T. Bosse, S. H. K. Narayanan, and Laurent Hascoet. Piecewise linear ad via source transformation. 2016.
3. Tanja Clees, Igor Nikitin, and Lialia Nikitina. Making network solvers globally convergent. In Mohammad S. Obaidat, Tuncer Ören, and Yuri Merkuryev, editors, *Simulation and Modeling Methodologies, Technologies and Applications*, pages 140–153, Cham, 2018. Springer International Publishing.
4. Sabrina Fiege, Andrea Walther, and Andreas Griewank. An algorithm for nonsmooth optimization by successive piecewise linearization. *Mathematical Programming*, Apr 2018. URL: <https://doi.org/10.1007/s10107-018-1273-5>, doi:10.1007/s10107-018-1273-5.
5. Sabrina Fiege, Andrea Walther, Kshitij Kulshreshtha, and Andreas Griewank. Algorithmic differentiation for piecewise smooth functions: a case study for robust optimization. *Optimization Methods and Software*, 0(0):1–16, 2017. URL: <https://doi.org/10.1080/10556788.2017.1333613>, arXiv:<https://doi.org/10.1080/10556788.2017.1333613>, doi:10.1080/10556788.2017.1333613.
6. A. Griewank. On stable piecewise linearization and generalized algorithmic differentiation. *Optimization Methods and Software*, 28(6):1139–1178, 2013. URL: <http://dx.doi.org/10.1080/10556788.2013.796683>, doi:10.1080/10556788.2013.796683.
7. A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), 2008. URL: <http://epubs.siam.org/doi/book/10.1137/1.9780898717761>.
8. A. Griewank and A. Walther. First- and second-order optimality conditions for piecewise smooth objective functions. *Optimization Methods and Software*, 31(5):904–930, 2016. URL: <https://doi.org/10.1080/10556788.2016.1189549>, arXiv:<https://doi.org/10.1080/10556788.2016.1189549>, doi:10.1080/10556788.2016.1189549.
9. Andreas Griewank, Jens-Uwe Bernt, Manuel Radons, and Tom Streubel. Solving piecewise linear systems in abs-normal form. *Linear Algebra and its Applications*, 471:500 – 530, 2015. URL: <http://www.sciencedirect.com/science/article/pii/S0024379514008209>, doi:<https://doi.org/10.1016/j.laa.2014.12.017>.
10. Andreas Griewank, Richard Hasenfelder, Manuel Radons, Lutz Lehmann, and Tom Streubel. Integrating lipschitzian dynamical systems using piecewise algorithmic differentiation. *Optimization Methods and Software*, 0(0):1–19, 2017. URL: <https://doi.org/10.1080/10556788.2017.1378653>, arXiv:<https://doi.org/10.1080/10556788.2017.1378653>, doi:10.1080/10556788.2017.1378653.

11. Andreas Griewank, Tom Streubel, Lutz Lehmann, Manuel Radons, and Richard Hasenfelder. Piecewise linear secant approximation via algorithmic piecewise differentiation. *Optimization Methods and Software*, 0(0):1–19, 2017. URL: <https://doi.org/10.1080/10556788.2017.1387256>, arXiv:<https://doi.org/10.1080/10556788.2017.1387256>, doi:10.1080/10556788.2017.1387256.
12. Andreas Griewank, Andrea Walther, Sabrina Fiege, and Torsten Bosse. On lipschitz optimization based on gray-box piecewise linearization. *Mathematical Programming*, 158(1):383–415, Jul 2016. URL: <https://doi.org/10.1007/s10107-015-0934-x>, doi:10.1007/s10107-015-0934-x.
13. E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2008. URL: <https://books.google.de/books?id=F93u7VcSRyYC>.
14. K. Kubota. Enumeration of subdifferentials of piecewise linear functions with abs-normal form. *Optimization Methods and Software*, 0(0):1–17, 2018. URL: <https://doi.org/10.1080/10556788.2018.1458848>, arXiv:<https://doi.org/10.1080/10556788.2018.1458848>, doi:10.1080/10556788.2018.1458848.
15. R. Lamour, R. März, and C. Tischendorf. *Differential-Algebraic Equations: A Projector Based Analysis*. Differential-Algebraic Equations Forum. Springer Berlin Heidelberg, 2013. URL: <https://books.google.de/books?id=Ca1AAAAAQBAJ>.
16. Roy B. Leipnik and Charles E. M. Pearce. The multivariate fa di bruno formula and multivariate taylor expansions with explicit integral remainder term. *The ANZIAM Journal*, 48(3):327341, 2007. doi:10.1017/S1446181100003527.
17. Uwe Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. SIAM, 2012.
18. Manuel Radons. Direct solution of piecewise linear systems. *Theoretical Computer Science*, 626:97 – 109, 2016. URL: <http://www.sciencedirect.com/science/article/pii/S0304397516001304>, doi:<https://doi.org/10.1016/j.tcs.2016.02.009>.
19. Manuel Radons. A note on surjectivity of piecewise affine mappings. *Optimization Letters*, May 2018. URL: <https://doi.org/10.1007/s11590-018-1271-9>, doi:10.1007/s11590-018-1271-9.
20. S. Scholtes. *Introduction to Piecewise Differentiable Equations*. SpringerBriefs in optimization. Springer New York, 2012. URL: <http://link.springer.com/book/10.1007/978-1-4614-4340-7>.
21. Tom Streubel, Andreas Griewank, Manuel Radons, and Jens-Uwe Bernt. Representation and analysis of piecewise linear functions in abs-normal form. In Christian Pötzsche, Clemens Heuberger, Barbara Kaltenbacher, and Franz Rendl, editors, *System Modeling and Optimization*, pages 327–336, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
22. Tom Streubel, Christian Strohm, Philipp Trunschke, and Caren Tischendorf. Generic construction and efficient evaluation of network daes and their derivatives in the context of gas networks. *Operations Research Proceedings*, 2017. accepted for publication and to appear. doi:10.1007/978-3-319-89920-6.