THE DIFFERENTIAL ALGEBRAIC STRUCTURE OF THE LEVI-CIVITA FIELD AND APPLICATIONS

Khodr Shamseddine[1] and Martin Berz[2]
Department of Mathematics
Department of Physics and Astronomy and
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824, USA
[1] shamseddine@nscl.msu.edu
[2] berz@msu.edu

Abstract: It is shown that the non-Archimedean field introduced by Levi-Civita [7, 8] admits a derivation and hence is a differential algebraic field [1, 2, 4].

The differential algebraic structure of the Levi-Civita field is utilized for the decision of differentiability of functional dependencies on a computer, as well as the practical computation of derivatives. As such, it represents a new method for computational differentiation [5] that avoids the well-known accuracy problems of numerical differentiation tools. It also avoids the often rather stringent limitations of formula manipulators that restrict the complexity of the function that can be differentiated, and the orders to which differentiation can be performed. Examples for the use of the method for typical pathological problems are given.

AMS Subject Classification: 26E30, 12J25, 33F99.
Key Words: Non-Archimedean Analysis, Computational Differentiation, Levi-Civita Field, Computer Functions, Differential Algebra.

1. Introduction

The general question of efficient differentiation is at the core of many parts of the work on perturbation and aberration theories relevant in Physics and Engineering; for an overview, see for example [5]. In this case, derivatives of highly complicated functions have to be computed to high orders. However, even when the derivative of the function is known to exist at the given point, numerical methods fail to give an accurate value of the derivative; the error increases with the order, and for orders greater than three, the errors often become too large for the results to be practically useful. On the other hand, while formula manipulators like Mathematica are successful in finding low-order derivatives of simple functions, they fail for high-order derivatives of very complicated functions. Consider, for example, the function

$$g(x) = \frac{\sin\left(x^3 + 2x + 1\right) + \frac{3+\cos(\sin(\ln|1+x|))}{\exp\left(\tanh\left(\sinh\left(\cosh\left(\frac{\sin(\cos(\tan(\exp(x))))}{\cos(\sin(\exp(\tan(x+2))))}\right)\right)\right)\right)}}{2 + \sin\left(\sinh\left(\cos\left(\tan^{-1}\left(\ln\left(\exp(x) + x^2 + 3\right)\right)\right)\right)\right)}. \tag{1.1}$$

Using the differential algebraic (DA) methods discussed in the subsequent sections and implemented in COSY INFINITY [3, 6], we find $g^{(n)}(0)$ for $0 \le n \le 19$. These numbers are listed in table 1; we note

Order n	$g^{(n)}(0)$	CPU Time
0	1.004845319007115	1.820 msec
1	0.4601438089634254	2.070 msec
2	-5.266097568233224	3.180 msec
3	-52.82163351991485	4.830 msec
4	-108.4682847837855	7.700 msec
5	16451.44286410806	11.640 msec
6	541334.9970224757	18.050 msec
7	7948641.189364974	26.590 msec
8	-144969388.2104904	37.860 msec
9	-15395959663.01733	52.470 msec
10	-618406836695.3634	72.330 msec
11	-11790314615610.74	97.610 msec
12	403355397865406.1	128.760 msec
13	$0.5510652659782951 \times 10^{17}$	168.140 msec
14	$0.3272787402678642 \times 10^{19}$	217.510 msec
15	$0.1142716430145745 \times 10^{21}$	273.930 msec
16	$-0.6443788542310285 \times 10^{21}$	344.880 msec
17	$-0.5044562355111304 \times 10^{24}$	423.400 msec
18	$-0.5025105824599693 \times 10^{26}$	520.390 msec
19	$-0.3158910204361999 \times 10^{28}$	621.160 msec

TABLE 1. $g^{(n)}(0), 0 \le n \le 19$, computed with DA methods

that, for $0 \le n \le 19$, we list the CPU time needed to obtain all derivatives of g at 0 up to order n and not just $g^{(n)}(0)$. For comparison purposes, we give in table 2 the function value and the first six deriva-

Order n	$g^{(n)}(0)$	CPU Time
0	1.004845319007116	0.11 sec
1	0.4601438089634254	0.17 sec
2	−5.266097568233221	0.47 sec
3	−52.82163351991483	2.57 sec
4	−108.4682847837854	14.74 sec
5	16451.44286410805	77.50 sec
6	541334.9970224752	693.65 sec

TABLE 2. $g^{(n)}(0), 0 \leq n \leq 6$, computed with Mathematica

tives computed with Mathematica. Note that the respective values listed in tables 1 and 2 agree. However, Mathematica used much more CPU time to compute the first six derivatives, and it failed to find the seventh derivative as it ran out of memory. We also list in table 3 the

Order n	$g^{(n)}(0)$	Relative Error
1	0.4601437841866840	54×10^{-9}
2	−5.266346392944456	47×10^{-6}
3	−52.83767867680922	30×10^{-5}
4	−87.27214664649106	0.20
5	19478.29555909866	0.18
6	633008.9156614641	0.17
7	−12378052.73279768	2.6
8	−1282816703.632099	7.8
9	83617811421.48561	6.4
10	91619495958355.24	149

TABLE 3. $g^{(n)}(0), 1 \leq n \leq 10$, computed numerically

first ten derivatives of g at 0 computed numerically using the numerical differentiation formulas

$$g^{(n)}(0) = (\Delta x)^{-n} \left(\sum_{j=0}^{n} (-1)^{n-j} \begin{pmatrix} n \\ j \end{pmatrix} g\,(j\Delta x) \right), \quad \Delta x = 10^{-16/(n+1)},$$

for $1 \leq n \leq 10$, together with the corresponding relative errors obtained by comparing the numerical values with the respective exact values computed with DA.

On the other hand, formula manipulators fail to find the derivatives of certain functions at given points even though the functions are differentiable at the respective points. For example, the functions

$$g_1(x) = |x|^{5/2} \cdot g(x) \text{ and } g_2(x) = \begin{cases} \frac{1-\exp(-x^2)}{x} \cdot g(x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases},$$

where $g(x)$ is the function given in Equation (1.1), are both differentiable at 0; but the attempt to compute their derivatives using formula manipulators fails. This is not specific to g_1 and g_2, and is generally connected to the occurrence of non-differentiable parts that do not affect the differentiability of the end result, of which case g_1 is an example, as well as the occurrence of branch points in coding as in IF-ELSE structures, of which case g_2 is an example.

We show that the differential algebraic structure of the Levi-Civita field $\mathcal{R}$ [1, 2, 4, 12, 10] allows to study many problems connected to computational differentiation [2, 11]. Using the calculus on $\mathcal{R}$, we formulate a necessary and sufficient condition for the derivatives of a large class of functions representable on a computer to exist, and show how to find these derivatives whenever they exist.

2. The Differential Algebraic Structure of $\mathcal{R}$

In this section, we introduce an operator ∂ on $\mathcal{R}$ which will be useful for the concept of differentiation.

Definition 1. *Define* $\partial : \mathcal{R} \to \mathcal{R}$ *by* $(\partial x)[q] = (q+1)x[q+1]$.

Lemma 1. *∂ is a derivation on $\mathcal{R}$; that is*

$$\partial(x+y) = \partial x + \partial y \text{ and } \partial(x \cdot y) = (\partial x) \cdot y + x \cdot (\partial y) \text{ for all } x, y \in \mathcal{R}.$$

Thus, $(\mathcal{R}, +, \cdot, \partial)$ is a differential algebraic field. Furthermore, we have that $\lambda(\partial x) = \lambda(x) - 1$ if $\lambda(x) \neq 0, \infty$ and $\partial 0 = 0$. However, if $\lambda(x) = 0$, then $\lambda(\partial x)$ can be either greater than, equal to, or smaller than $\lambda(x)$.

Proof. Let $x, y \in \mathcal{R}$ and let $q \in \mathbb{Q}$ be given. Then

$$\begin{aligned} (\partial(x+y))[q] &= (q+1)(x+y)[q+1] \\ &= (q+1)x[q+1] + (q+1)y[q+1] \\ &= (\partial x)[q] + (\partial y)[q]. \end{aligned}$$

This is true for all $q \in \mathbb{Q}$; hence $\partial(x+y) = \partial x + \partial y$.

For all $q \in \mathbb{Q}$, we also have that

$$
\begin{aligned}
& (\partial(x\cdot y))\,[q] = (q+1)(x\cdot y)[q+1] \\
= \; & (q+1) \sum_{\substack{q_1+q_2=q+1 \\ q_1\in\operatorname{supp}(x),\, q_2\in\operatorname{supp}(y)}} x[q_1]y[q_2] \\
= \; & \sum_{\substack{q_1+q_2=q+1 \\ q_1\in\operatorname{supp}(x),\, q_2\in\operatorname{supp}(y)}} (q+1)x[q_1]y[q_2] \\
= \; & \sum_{\substack{q_1+q_2=q+1 \\ q_1\in\operatorname{supp}(x),\, q_2\in\operatorname{supp}(y)}} (q_1x[q_1]y[q_2] + x[q_1]q_2x[q_2]) \\
= \; & \sum_{\substack{s+t=q \\ s+1\in\operatorname{supp}(x),\, t\in\operatorname{supp}(y)}} (s+1)x[s+1]y[t] + \\
& \sum_{\substack{s+t=q \\ s\in\operatorname{supp}(x),\, t+1\in\operatorname{supp}(y)}} x[s](t+1)y[t+1] \\
= \; & \sum_{\substack{s+t=q \\ s\in\operatorname{supp}(\partial x),\, t\in\operatorname{supp}(y)}} (\partial x)[s]y[t] + \sum_{\substack{s+t=q \\ s\in\operatorname{supp}(x),\, t\in\operatorname{supp}(\partial y)}} x[s](\partial y)[t] \\
= \; & ((\partial x)\cdot y)\,[q] + (x\cdot(\partial y))\,[q] = ((\partial x)\cdot y + x\cdot(\partial y))\,[q].
\end{aligned}
$$

This is true for all $q \in \mathbb{Q}$; and hence $\partial(x\cdot y) = (\partial x)\cdot y + x\cdot(\partial y)$.

Now let $x \in \mathcal{R}$ be given such that $\lambda(x) \neq 0, \infty$. Then for all $q < \lambda(x)-1$, we have that $q+1 < \lambda(x)$; and hence $(\partial x)[q] = (q+1)x[q+1] = 0$. Hence $\lambda(\partial x) \geq \lambda(x) - 1$; but $(\partial x)[\lambda(x)-1] = \lambda(x)x[\lambda(x)] \neq 0$. Hence, $\lambda(\partial x) = \lambda(x) - 1$.

On the other hand, we have that $(\partial 0)[q] = (q+1)0[q+1] = 0$ for all $q \in \mathbb{Q}$. Thus, $\partial 0 = 0$; and hence $\lambda(\partial 0) = \lambda(0) = \infty$.

To prove the last statement, let $x_1 = 1$, $x_2 = 1+d$, and $x_3 = 1+d^{1/2}$; then $\lambda(x_j) = 0$ for $j = 1, 2, 3$. We have that

$$
\begin{aligned}
\partial x_1 &= 0, \text{ and hence } \lambda(\partial x_1) > \lambda(x_1); \\
\partial x_2 &= 1, \text{ and hence } \lambda(\partial x_2) = \lambda(x_2); \\
\partial x_3 &= \frac{1}{2}d^{-1/2}, \text{ and hence } \lambda(\partial x_3) < \lambda(x_3).
\end{aligned}
$$

□

3. Computer Environment Functions

At the machine level, a function $f : \mathbb{R} \to \mathbb{R}$ is characterized by what it does to the original set of memory locations. So f induces a function $\vec{F}(f) : \mathbb{R}^m \to \mathbb{R}^m$, where m is the number of memory locations affected in the process of computing f. We note here that, without compiler optimization, $\vec{F}(f)$ is unique up to flipping of the memory locations; on the other hand, with compiler optimization, $\vec{F}(f)$ is unique in the subspace describing the true variables. Moreover, at the machine level, any code constitutes solely of intrinsic functions, arithmetic operations and branches. In the following, we formally define the machine level representations of intrinsic functions, the Heaviside function, and the arithmetic operations.

Definition 2. *Let $\mathcal{I} = \{H, \sin, \cos, \tan, \exp, \ldots\}$ be the set consisting of the Heaviside function H and all the intrinsic functions on a computer, which for the sake of convenience are assumed to include the reciprocal function; and let $\mathcal{O} = \{+, \cdot\}$.*

Definition 3. *For $f \in \mathcal{I}$, define $\vec{F}_{i,k,f} : \mathbb{R}^m \to \mathbb{R}^m$ by*

$$\vec{F}_{i,k,f}(x_1, x_2, \ldots, x_m) = (x_1, \ldots, x_{k-1}, \underbrace{f(x_i)}_{k}, x_{k+1}, \ldots, x_m);$$

so the kth memory location is replaced by $f(x_i)$. Then $\vec{F}_{i,k,f}$ is the machine level representation of f. For $\otimes \in \mathcal{O}$, define $\vec{F}_{i,j,k,\otimes} : \mathbb{R}^m \to \mathbb{R}^m$ by

$$\vec{F}_{i,j,k,\otimes}(x_1, x_2, \ldots, x_m) = (x_1, \ldots, x_{k-1}, \underbrace{x_i \otimes x_j}_{k}, x_{k+1}, \ldots, x_m),$$

so the kth memory location is replaced by $x_i \otimes x_j$. Then $\vec{F}_{i,j,k,\otimes}$ is the machine level representation of $\otimes$. Finally, let

$$\mathcal{F} = \{\vec{F}_{i,k,f} : f \in \mathcal{I}\} \cup \{\vec{F}_{i,j,k,\otimes} : \otimes \in \mathcal{O}\}.$$

Definition 4. *A function $f : \mathbb{R} \to \mathbb{R}$ is called a computer function if and only if it can be obtained from intrinsic functions and the Heaviside function through a finite number of arithmetic operations and compositions. In this case, there are some $\vec{F}_1, \vec{F}_2, \ldots, \vec{F}_N \in \mathcal{F}$ such that $\vec{F}(f) = \vec{F}_N \circ \vec{F}_{N-1} \circ \cdots \circ \vec{F}_2 \circ \vec{F}_1$, and we call $\vec{F}(f) : \mathbb{R}^m \to \mathbb{R}^m$, already mentioned above, the machine level representation of f.*

Obviously, the so defined class of computer functions in a formal way describes all those functions that can be evaluated on a computer. Since we will be studying only computer functions, it will be useful to define the domain D_c of computer numbers as the subset of the real numbers representable on a computer.

We recall the following result [1, 2, 4] which allows us to extend all intrinsic functions given by power series to $\mathcal{R}$. Also, for a detailed study of power series with $\mathcal{R}$ coefficients, we refer the reader to [12, 10].

Theorem 1 (Power Series with Purely Real Coefficients)**.** *Let $\sum_{n=0}^{\infty} a_n X^n$ be a power series with real coefficients and with classical radius of convergence equal to η. Let $x \in \mathcal{R}$, and let $A_n(x) = \sum_{i=0}^{n} a_i x^i \in \mathcal{R}$. Then, for $|x| < \eta$ and $|x| \not\approx \eta$, the sequence $(A_n(x))$ converges absolutely weakly. We define the limit to be the continuation of the power series on $\mathcal{R}$.*

Remark 1. *The continuation $\bar{H}$ of the real Heaviside function H is defined for all $x \in \mathcal{R}$ by*

$$\bar{H}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} .$$

The functions $\sqrt[n]{x}$ and $1/x$ are continued to $\mathcal{R}$ via the existence of roots and multiplicative inverses on $\mathcal{R}$.

Definition 5. *Let $f \in \mathcal{I}$, let D be the domain of definition of f in $\mathbb{R}$, let $x_0 \in D$, and let $s \in \mathcal{R}$. Then we say that f is extendable to $x_0 + s$ if and only if $x_0 + s$ belongs to the domain of definition of $\bar{f}$, the continuation of f to $\mathcal{R}$, where $\bar{f}$ is given by Theorem 1 and Remark 1.*

Let $f_1, f_2 \in \mathcal{I}$ with domains of definition D_1 and D_2 in $\mathbb{R}$ respectively, let $x_0 \in D_1 \cap D_2$, let $s \in \mathcal{R}$, and let $\otimes \in \{+, \cdot\}$. Then we say that $f_2 \otimes f_1$ is extendable to $x_0 + s$ if and only if f_1 and f_2 are both extendable to $x_0 + s$.

Let $f_1, f_2 \in \mathcal{I}$ with domains of definition D_1 and D_2 in $\mathbb{R}$ respectively, let $x_0 \in D_1$ be such that $f_1(x_0) \in D_2$, and let $s \in \mathcal{R}$. Then we say that $f_2 \circ f_1$ is extendable to $x_0 + s$ if and only if f_1 is extendable to $x_0 + s$ and f_2 extendable to $f_1(x_0 + s)$.

Finally, let f be a real computer function, let D be the domain of definition of f in $\mathbb{R}$, let $x_0 \in D$, and let $s \in \mathcal{R}$; then f is obtained in finitely many steps from functions in $\mathcal{I}$ via compositions and arithmetic operations. We define extendability of f to $x_0 + s$ inductively.

We have the following result about the local form of computer functions, which will prove useful in studying the differentiability of computer functions.

Theorem 2. *Let f be a real computer function with domain of definition D, and let $x_0 \in D$ be such that f is extendable to $x_0 \pm d$. Then* *there exists $\sigma > 0$ in $\mathbb{R}$ such that, for $0 < x < \sigma$,*

$$f(x_0 \pm x) = A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x), \tag{3.1}$$

where $A_i^{\pm}(x), 0 \le i \le i^{\pm}$, is a power series in x with a radius of convergence no smaller than σ, $A_i^{\pm}(0) \neq 0$ for $i = 1, \ldots, i^{\pm}$, and the $q_i^{\pm}$'s are nonzero rational numbers that are not positive integers.

Remark 2. *Noninteger rational powers may appear in Equation (3.1) as a result of the root function.*

Proof. The statement of the theorem can easily be verified for each $f \in \mathcal{I}$.

Let f_1 and f_2 be two computer functions with domains of definition D_1 and D_2 in $\mathbb{R}$, respectively. Let $x_0 \in D_1 \cap D_2$, let f_1 and f_2 be both extendable to $x_0 \pm d$, and let f_1 and f_2 satisfy Equation (3.1) around x_0. For $\otimes \in \{+, \cdot\}$, let $F_{\otimes} = f_2 \otimes f_1$. Thus we have that $f_1(x_0 \pm x) = A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ for $x \in (0, \sigma_1)$, and $f_2(x_0 \pm x) = B_0^{\pm}(x) + \sum_{j=1}^{j^{\pm}} x^{t_j^{\pm}} B_j^{\pm}(x)$ for $x \in (0, \sigma_2)$, where σ_1 and σ_2 are both positive real numbers; $A_i^{\pm}(x), 0 \le i \le i^{\pm}$, and $B_j^{\pm}(x), 0 \le j \le j^{\pm}$, are power series in x with radii of convergence no smaller than $\sigma = \min\{\sigma_1, \sigma_2\}$; $A_i^{\pm}(0) \neq 0$ for $i \in \{1, \ldots, i^{\pm}\}$ and $B_j^{\pm}(0) \neq 0$ for $j \in \{1, \ldots, j^{\pm}\}$; and the $q_i^{\pm}$'s and the $t_j^{\pm}$'s are nonzero rational numbers that are not positive integers. As a reminder, we note that σ_1, σ_2, the $A_i^{\pm}$'s, the $B_j^{\pm}$'s, the $q_i^{\pm}$'s, and the $t_j^{\pm}$'s depend on x_0.

For $0 < x < \sigma$, we have that

$$\begin{aligned} F_{\otimes}(x_0 \pm x) &= f_2(x_0 \pm x) \otimes f_1(x_0 \pm x) \\ &= \left(\sum_{i=0}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x) \right) \otimes \left(\sum_{j=0}^{j^{\pm}} x^{t_j^{\pm}} B_j^{\pm}(x) \right), \end{aligned} \tag{3.2}$$

where $q_0^{\pm} = t_0^{\pm} = 0$. It is easy to check that, for $\otimes = +$ or $\otimes = \cdot$, the result in Equation (3.2) is an expression of the form of Equation (3.1).

Now let f_1 and f_2 be two computer functions with domains of definition D_1 and D_2 in $\mathbb{R}$, respectively. Let $x_0 \in D_1$, let f_1 be extendable to $x_0 \pm d$, let f_2 be extendable to $f_1(x_0 \pm d)$, and let f_1 and f_2 satisfy Equation (3.1) around x_0 and $f_1(x_0)$, respectively. Let $F_{\circ} = f_2 \circ f_1$.

Thus we have that

$$\begin{aligned} f_1(x_0 \pm x) &= A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x) \text{ for } x \in (0, \sigma_1), \\ f_2(f_1(x_0) \pm y) &= B_0^{\pm}(y) + \sum_{j=1}^{j^{\pm}} y^{t_j^{\pm}} B_j^{\pm}(y) \text{ for } y \in (0, \sigma_2), \end{aligned}$$

where σ_1 and σ_2 are positive real numbers; $A_i^{\pm}(x), 0 \leq i \leq i^{\pm}$ and $B_j^{\pm}(y), 0 \leq j \leq j^{\pm}$, are power series in x and y with radii of convergence no smaller than $\sigma = \min\{\sigma_1, \sigma_2\}$; $A_i^{\pm}(0) \neq 0$ for $i \in \{1, \ldots, i^{\pm}\}$ and $B_j^{\pm}(0) \neq 0$ for $j \in \{1, \ldots, j^{\pm}\}$; and the $q_i^{\pm}$'s and the $t_j^{\pm}$'s are nonzero rational numbers that are not positive integers. Without loss of generality, we may assume that at least one of the series $B_j^{\pm}(y)$ is infinite. It follows, since f_2 is extendable to $f_1(x_0 \pm d)$, that the $q_i^{\pm}$'s are all positive and that $A_0^{\pm}(0) = f_1(x_0)$. Let $A_{00}^{\pm}(x) = A_0^{\pm}(x) - A_0^{\pm}(0) = A_0^{\pm}(x) - f_1(x_0)$. Then $A_{00}^{\pm}(x)$ has no constant term, and we have, for $0 < x < \sigma_1$, that $f_1(x_0 \pm x) = f_1(x_0) + A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$. Since $A_{00}^{\pm}(x)$ has no constant term and the $q_i^{\pm}$'s are all positive, there exists $\sigma \in \mathbb{R}$, $0 < \sigma \leq \sigma_1$, such that $|A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)| < \sigma_2$ and $A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ has the same sign for all x satisfying $0 < x < \sigma$. To prove the last statement, note that since $g^{\pm}(x) = A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ is continuous at 0, there exists $\delta_1 \in \mathbb{R}$, $0 < \delta_1 \leq \sigma_1$, such that $0 < x < \delta_1 \Rightarrow |g^{\pm}(x) - g^{\pm}(0)| = |A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)| < \sigma_2$. Now let $\alpha^{\pm} x^{q^{\pm}}$ be the leading term of $g^{\pm}(x)$. Write $g^{\pm}(x) = \alpha^{\pm} x^{q^{\pm}} \left(1 + g_1^{\pm}(x)\right)$, where $g_1^{\pm}(x)$ is continuous at 0 and $g_1^{\pm}(0) = 0$. Hence there exists $\delta_2 \in \mathbb{R}$, $0 < \delta_2 \leq \sigma_1$, such that $0 < x < \delta_2 \Rightarrow |g_1^{\pm}(x)| < 1/2 \Rightarrow 1 + g_1^{\pm}(x) > 0 \Rightarrow g^{\pm}(x)$ has the same sign as $\alpha^{\pm}$. Let $\sigma = \min\{\delta_1, \delta_2\}$. Then $0 < \sigma \leq \sigma_1$, and $0 < x < \sigma \Rightarrow |A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)| < \sigma_2$ and $A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ has the same sign as $\alpha^{\pm}$. Thus, for $0 < x < \sigma$, we have that

$$\begin{aligned} &F_{\circ}(x_0 \pm x) = f_2\left(f_1(x_0 \pm x)\right) = f_2\left(f_1(x_0) + A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right) \\ &= E_0\left(A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right) + \\ &\sum_{j=1}^{J}\left\{\left|A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right|^{s_j} E_j\left(A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right)\right\}, \end{aligned}$$

where $E_j, 0 \le j \le J$, are power series; $E_j(0) \neq 0$ for $1 \le j \le J$; and the s_j's are nonzero rational numbers that are not positive integers.

Note that for $1 \le j \le J$,

$$\begin{aligned} \left| A_{00}^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x) \right|^{s_j} &= \left|\alpha^{\pm}\right|^{s_j} x^{s_j q^{\pm}} \left(1 + g_1^{\pm}(x)\right)^{s_j} \\ &= \left|\alpha^{\pm}\right|^{s_j} x^{s_j q^{\pm}} S_j(g_1^{\pm}(x)), \end{aligned}$$

where $g_1^{\pm}(x)$ is of the form of Equation (3.1), $g_1^{\pm}(0) = 0$, $|g_1^{\pm}(x)| < 1/2$, and $S_j(g_1^{\pm}(x)) = \left(1 + g_1^{\pm}(x)\right)^{s_j}$ is a power series in $g_1^{\pm}(x)$. Thus, it suffices to show that a power series of an expression of the form of Equation (3.1), in which the $q_i^{\pm}$'s are all positive and in which $A_0^{\pm}(0) = 0$, yields an expression of the same form.

So let $S(y) = \sum_{m=0}^{\infty} a_m y^m$ be a power series with positive radius of convergence η. Then, for x sufficiently small,

(3.3)

$$S\left(A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right) = \sum_{m=0}^{\infty} a_m \left(A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right)^m .$$

For each $i \in \{1, \ldots, i^{\pm}\}$, write $q_i^{\pm} = m_i^{\pm}/n_i^{\pm}$, where $m_i^{\pm}$ and $n_i^{\pm}$ are positive and relatively prime. Expanding the powers in Equation (3.3), the only exponents of x that may occur are of the form $k + s$, where k is a positive integer and

$$s \in T = \left\{ \frac{m_i^{\pm}}{n_i^{\pm}}, \ldots, (n_i^{\pm} - 1)\frac{m_i^{\pm}}{n_i^{\pm}} / i = 1, \ldots, i^{\pm} \right\},$$

a finite set. For each m let $S_m(x) = a_m \left(A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right)^m$. Then S_m is an infinite series

$$S_m(x) = \sum_{n=0}^{\infty} u_{mn}(x), \tag{3.4}$$

where $u_{mn}(x)$ is of the form $a_{mn} x^{k+s}$ with $a_{mn} \in \mathbb{R}$, k a positive integer, and $s \in T$. Let η_1 be the radius of convergence of $A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$, and let $0 < x < \eta_1/2$ be such that

$$\left| A_0^{\pm}(x) + \sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x) \right| < \frac{\eta}{2}.$$

Then for each m, the sum in Equation (3.4) converges absolutely; so we can rearrange the terms in S_m. Moreover, the double sum

$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} u_{mn}(x)$ converges; so we can interchange the order of the sums (see for example [9], pages 205-208) and we obtain that

$$S\left(A_0^{\pm}(x)+\sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right)=\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} u_{mn}(x)=\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} u_{mn}(x).$$

Thus rearranging and regrouping the terms in Equation (3.3), we obtain an expression of the form $C_0^{\pm}(x)+\sum_{p=1}^{p^{\pm}} x^{r_p^{\pm}} C_p^{\pm}(x)$, where $C_p^{\pm}(x), 0 \leq p \leq p^{\pm}$, are power series, $C_p^{\pm}(0) \neq 0$ for $1 \leq p \leq p^{\pm}$, $p^{\pm}$ is finite, and the $r_p^{\pm}$'s are nonzero rational numbers which are not positive integers. Hence $S\left(A_0^{\pm}(x)+\sum_{i=1}^{i^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)\right)$ is of the form of Equation (3.1). It follows that $F_{\circ}(x_0 \pm x)$ in Equation (3.3) is itself of the form of Equation (3.1).

Now let f be a real computer function with domain of definition D, and let $x_0 \in D$ be such that f is extendable to $x_0 \pm d$. Then f is obtained in finitely many steps from functions in $\mathcal{I}$ via compositions and arithmetic operations. Using induction, we obtain the result immediately from the above. □

Since the family of computer functions is closed under differentiation to any order n, Theorem 2 holds for derivatives of computer functions as well.

Definition 6. *(Continuation of Real Computer Functions) Let f be a real computer function with domain of definition D and let $x_0 \in D$ be such that f is extendable to $x_0 \pm d$. Then f is given around x_0 by a finite combination of roots and power series. Since roots and power series have already been extended to $\mathcal{R}$, f is extended to $\mathcal{R}$ around x_0 in a natural way similar to that of the extension of power series from $\mathbb{R}$ to $\mathbb{C}$. That is, if $f(x_0 \pm x)=A_0^{\pm}(x)+\sum_{i=1}^{i_k^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ for $0<x<\sigma$, then we have for the continued function $\bar{f}$ that $\bar{f}(x_0 \pm x)=A_0^{\pm}(x)+\sum_{i=1}^{i_k^{\pm}} x^{q_i^{\pm}} A_i^{\pm}(x)$ for all $x \in \mathcal{R}$ satisfying $0<x<\sigma$ and $x \not\approx \sigma$.*

Having built the necessary theoretical tools, we next try to use the results of this section to compute derivatives of real functions. In the rest of this paper we will use f instead of $\bar{f}$ to represent the continuation of a real computer function f.

4. Computation of Derivatives with Derivations

In this section, we develop a criterion that will allow us not only to check the continuity and the differentiability of a real computer function f at a point x_0, but also to obtain all existing derivatives of f at x_0.

Lemma 2. *Let f be a computer function. Then f is defined at $x_0 \in D_c$ if and only if $f(x_0)$ can be evaluated on a computer.*

This lemma of course hinges on a careful implementation of the intrinsic functions and operations, in particular in the sense that they should be executable for any floating point number in the domain of definition that produces a result within the range of allowed floating point numbers.

Lemma 3. *Let f be a computer function, let D be the domain of definition of f in $\mathbb{R}$, let $x_0 \in D \cap D_c$, and let $s \in \mathcal{R}$. Then f is extendable to $x_0 + s$ if and only if $f(x_0 + s)$ can be evaluated on the computer.*

Lemma 4. *Let f be a computer function, and let x_0 be such that f is defined at x_0 and extendable to $x_0 \pm d$. Then f is continuous at x_0 if and only if $f(x_0 - d) =_0 f(x_0) =_0 f(x_0 + d)$.*

Proof. Since f is a computer function, defined at x_0 and extendable to $x_0 \pm d$, we have that

$$f(x_0+x) = A_0(x) + \sum_{j=1}^{J_r} x^{q_j} A_j(x) \text{ and } f(x_0-x) = B_0(x) + \sum_{j=1}^{J_l} x^{t_j} B_j(x)$$

for $0 < x < \sigma$, where σ is a positive real number; where the A_j's and the B_j's are power series in x, where $A_j(0) \neq 0$ for $1 \leq j \leq J_r$ and $B_j(0) \neq 0$ for $1 \leq j \leq J_l$; and where the q_j's and the t_j's are nonzero rational numbers that are not positive integers. Let $A_0(x) = \sum_{i=0}^{\infty} \alpha_i x^i$ and $B_0(x) = \sum_{i=0}^{\infty} \beta_i x^i$. Then f is continuous at x_0 if and only if $q_j > 0$ for all $j \in \{1, \ldots, J_r\}$, $t_j > 0$ for all $j \in \{1, \ldots, J_l\}$, and $\alpha_0 = \beta_0 = f(x_0)$; that is, if and only if $f(x_0 + d) =_0 f(x_0) =_0 f(x_0 - d)$. □

Theorem 3. *Let f be a computer function that is continuous at x_0 and extendable to $x_0 \pm d$. Then f is m times differentiable at x_0 if and only if, for all $j \in \{1, \ldots, m\}$, $\partial^j (f(x_0 + d))$ and $(-1)^j \partial^j (f(x_0 - d))$ are both at most finite in absolute value and their real parts agree. Moreover, in this case*

$$\partial^j (f(x_0 + d)) =_0 f^{(j)}(x_0) =_0 (-1)^j \partial^j (f(x_0 - d)) \text{ for } 1 \leq j \leq m.$$

Proof. Since f is continuous at x_0, we have that

$$f(x_0 + x) = f(x_0) + \sum_{i=1}^{\infty} \alpha_i x^i + \sum_{j=1}^{J_r} x^{q_j} A_j(x)$$

$$f(x_0 - x) = f(x_0) + \sum_{i=1}^{\infty} \beta_i x^i + \sum_{j=1}^{J_l} x^{t_j} B_j(x) \tag{4.1}$$

for $0 < x < \sigma$, where σ is a positive real number, where the A_j's and the B_j's are power series in x that do not vanish at $x = 0$, and where the q_j's and the t_j's are noninteger positive rational numbers. Thus,

$$f(x_0 + d) = f(x_0) + \sum_{i=1}^{\infty} \alpha_i d^i + \sum_{j=1}^{J_r} d^{q_j} A_j(d)$$

$$f(x_0 - d) = f(x_0) + \sum_{i=1}^{\infty} \beta_i d^i + \sum_{j=1}^{J_l} d^{t_j} B_j(d)$$

Assume f is m times differentiable at x_0. Then $q_j > m$ for all $j \in \{1, \ldots, J_r\}$, $t_j > m$ for all $j \in \{1, \ldots, J_l\}$, and $\alpha_j = (-1)^j \beta_j = f^{(j)}(x_0)/j!$ for all $j \in \{1, \ldots, m\}$. Hence,

$$f(x_0 + d) =_m f(x_0) + \sum_{j=1}^{n} \frac{f^{(j)}(x_0)}{j!} d^j \text{ and}$$

$$f(x_0 - d) =_m f(x_0) + \sum_{j=1}^{n} (-1)^j \frac{f^{(j)}(x_0)}{j!} d^j,$$

from which we obtain that

$$\partial^j \left(f(x_0 + d)\right) =_0 f^{(j)}(x_0) =_0 (-1)^j \partial^j \left(f(x_0 - d)\right) \text{ for } 1 \le j \le m.$$

The converse is proved similarly. □

All the arithmetic operations and all the transcendental functions have been implemented in COSY INFINITY [3, 6]. This allows us to apply the theoretical results of Section 4 for the computation of derivatives of real functions.

5. Examples

As a first example, we consider a simple function and study its differentiability at 0. Let $f(x) = x\sqrt{|x|} + \exp(x)$. It is easy to see that f is differentiable at 0 with $f(0) = f'(0) = 1$ and that f is not twice differentiable at 0. We will show now how using the result of Theorem

3 will lead us to the same conclusion. First we note that f is defined at 0 and extendable to $\pm d$.

It is useful to look at what goes on inside the computer for this simple example. Altogether, we need six memory locations to store the variable, the intermediate values, and the function value. These six memory locations are

$$\begin{array}{lll} x, & S_1 = \text{abs}(x), & S_2 = \text{sqrt}(S_1), \\ S_3 = x * S_2, & S_4 = \exp(x), & a = S_3 + S_4. \end{array}$$

So we can look at $\vec{F}(f)$ as a function from $\mathbb{R}^6$ into $\mathbb{R}^6$. Let

$$\begin{cases} \vec{E} : \mathbb{R} \to \mathbb{R}^6; & \vec{E}(x) = (x, 0, 0, 0, 0, 0) \\ \vec{F} : \mathbb{R}^6 \to \mathbb{R}^6; & \vec{F}(x, p_2, p_3, p_4, p_5, p_6) = (x, S_1, S_2, S_3, S_4, a) \\ P : \mathbb{R}^6 \to \mathbb{R}; & P(x, S_1, S_2, S_3, S_4, a) = a \\ G : \mathbb{R} \to \mathbb{R}; & G(x) = P \circ \vec{F} \circ \vec{E}(x) \end{cases}.$$

Then $G(x) = a =_M f(x)$, where M is an upper bound of the support points that can be obtained on the computer.

If we input the value $x = -d$, then the six memory locations will be filled as follows:

$$\begin{array}{lll} x = -d, & S_1 = d, & S_2 = d^{1/2}, \\ S_3 = -d^{3/2}, & S_4 = \sum_{j=0}^{M} (-1)^j d^j / j!, & a = -d^{3/2} + \sum_{j=0}^{M} (-1)^j d^j / j!. \end{array}$$

So the output will be $G(-d) = 1 - d - d^{3/2} + d^2/2! + \sum_{j=3}^{M} (-1)^j d^j / j! =_M f(-d)$. If we input the value $x = 0$, the output is $G(0) = 1$. Since $f(0)$ is real and $f(0) =_M G(0)$, we infer that $f(0) = 1$. Similarly, we find that $G(d) = 1 + d + d^{3/2} + d^2/2! + \sum_{j=3}^{M} d^j / j! =_M f(d)$.

Note that $f(-d) =_0 1 = f(0) =_0 f(d)$; hence f is continuous at 0. Since $\partial\,(f(d)) =_0 1 =_0 -\partial\,(f(-d))$, we infer that f is differentiable at 0, with $f'(0) = 1$. However, $\partial^2\,(f(d)) \sim d^{-1/2}$, which implies that $|\partial^2\,(f(d))\,|$ is infinitely large. Hence f is not twice differentiable at 0.

Next, we consider the two functions already mentioned in the introduction, g_1 and g_2, which are clearly computer functions. Consider first the function $g_1(x)$. If we input the values $x = -d, 0, d$, we obtain the following output up to depth 3

$$\begin{array}{rcl} g_1(\pm d) & =_3 & 1.004845319007115 d^{5/2} \\ g_1(0) & = & 0. \end{array}$$

Since $g_1(-d) =_0 g_1(0) =_0 g_1(d)$, g_1 is continuous at 0. From $\partial\left(g_1(d)\right) \sim d^{3/2} \sim -\partial\left(g_1(-d)\right)$, we infer, applying Theorem 3, that g_1 is differentiable at 0, with $g_1'(0) = 0$. Similarly we show that g_1 is twice differentiable at 0 with $g_1^{(2)}(0) = 0$. On the other hand, $\partial^3\left(g_1(d)\right) \sim d^{-1/2}$, which entails that $|\partial^3\left(g_1(d)\right)|$ is infinitely large. Hence g_1 is not three times differentiable at 0.

By evaluating $g_2(-d)$ and $g_2(d)$ up to any fixed depth and applying Theorem 3, we obtain that g_2 is differentiable at 0 up to arbitrarily high orders. In table 4, we list only the function value and the first nineteen

Order n	$g_2^{(n)}(0)$	CPU Time
0	0.	3.400 msec
1	1.004845319007115	4.030 msec
2	0.9202876179268508	5.710 msec
3	-18.81282866172102	8.240 msec
4	-216.8082597872205	12.010 msec
5	-364.2615904917884	17.570 msec
6	101933.1724529188	25.150 msec
7	3798311.370563978	35.700 msec
8	60765353.84260825	49.790 msec
9	-1441371402.871872	67.210 msec
10	-156736847166.3961	89.840 msec
11	-6725706835826.155	118.950 msec
12	-131199307184575.8	154.530 msec
13	5770286440090848.	200.660 msec
14	$0.7837443136320079 \times 10^{18}$	256.460 msec
15	$0.4850429351252696 \times 10^{20}$	321.630 msec
16	$0.1734774579876559 \times 10^{22}$	400.140 msec
17	$-0.1757849296527536 \times 10^{23}$	478.940 msec
18	$-0.9350429649226352 \times 10^{25}$	582.150 msec
19	$-0.9521402181303937 \times 10^{27}$	702.390 msec

TABLE 4. $g_2^{(n)}(0), 0 \le n \le 19$, computed with DA methods on $\mathcal{R}$

derivatives of g_2 at 0, together with the CPU time needed to compute all derivatives up to the respective order. The numbers in table 4 were obtained using the implementation of $\mathcal{R}$ in COSY INFINITY [3, 6].

References

[1] M. Berz. Analysis on a nonarchimedean extension of the real numbers. Lecture Notes, 1992 and 1995 Mathematics Summer Graduate Schools of the German National Merit Foundation. MSUCL-933, Department of Physics, Michigan State University, 1994.

[2] M. Berz. Calculus and numerics on Levi-Civita fields. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications, and Tools*, pages 19–35, Philadelphia, 1996. SIAM.

[3] M. Berz. COSY INFINITY Version 8 reference manual. Technical Report MSUCL-1088, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1997. see also http://www.beamtheory.nscl.msu.edu/cosy.

[4] M. Berz. Nonarchimedean analysis and rigorous computation. *International Journal of Applied Mathematics*, 2:889–930, 2000.

[5] M. Berz, C. Bischof, A. Griewank, G. Corliss, and Eds. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, 1996.

[6] M. Berz, G. Hoffstätter, W. Wan, K. Shamseddine, and K. Makino. COSY INFINITY and its applications to nonlinear dynamics. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications, and Tools*, pages 363–365, Philadelphia, 1996. SIAM.

[7] Tullio Levi-Civita. Sugli infiniti ed infinitesimi attuali quali elementi analitici. *Atti Ist. Veneto di Sc., Lett. ed Art.*, 7a, 4:1765, 1892.

[8] Tullio Levi-Civita. Sui numeri transfiniti. *Rend. Acc. Lincei*, 5a, 7:91,113, 1898.

[9] William Fogg Osgood. *Functions of Real Variables.* G. E. Stechert & CO., New York, 1938.

[10] K. Shamseddine. *New Elements of Analysis on the Levi-Civita Field.* PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1999. also MSUCL-1147.

[11] K. Shamseddine and M. Berz. Exception handling in derivative computation with nonarchimedean calculus. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications, and Tools*, pages 37–51, Philadelphia, 1996. SIAM.

[12] K. Shamseddine and M. Berz. Power series on the Levi-Civita field. *International Journal of Applied Mathematics*, 2:931–952, 2000.