Contents

45

Towards a Universal Data Type for Scientific Computing

Martin Berz

ABSTRACT Modern scientific computing uses an abundance of data types. Besides floating point numbers, we routinely use intervals, univariate Taylor series, Taylor series with interval coefficients, and more recently multivariate Taylor series. Newer are Taylor models, which allow verified calculations like intervals, but largely avoid many of their limitations, including the cancellation effect, dimensionality curse, and low-order scaling of resulting width to domain width. Another more recent structure is the Levi-Civita numbers, which allow viewing many aspects of scientific computation as an application of arithmetic and analysis with infinitely small numbers, and which are useful for a variety of purposes including the assessment of differentiability at branch points. We propose new methods based on partially ordered Levi-Civita algebras that allow for a unification of all these various approaches into one single data type.

45.1 Introduction

In this chapter we attempt to combine various data types used in scientific computation into a single one. Our primary interest lies in the ability to compute derivatives of as wide a class of functions representable on a computer as possible; in addition, we strive to make our arguments rigorous by employing interval techniques. To this end, we provide a generalization of the Levi-Civita numbers that on the one hand covers the multidimensional case, and on the other hand employs interval techniques to keep all arguments rigorous. The interval technique in particular will also allow answering a variety of questions concerned with proper exception handling merely within the one-dimensional case.

The Levi-Civita field $\mathcal{R}$ can be viewed as a set of functions from the rational numbers into the real numbers that are zero except on a set of rational numbers forming at most a strictly monotonically diverging sequence, as described in [4]. The structure forms an ordered field that is algebraically closed and admits infinitely small and large numbers [5]. In [4] and [19, 20] it is shown that for functions that can be represented as a computational graph consisting of elementary operations and the common intrinsics, derivatives can be obtained up to infinitely small error by evaluation of the difference quotient$(f(x+\Delta x)-f(x))/\Delta x$ for any infinitesimally small Δx; similar formulae hold for higher derivatives. Thus, the method retroactively justifies the thinking of Newton and Leibnitz, and allows treatment of infinitely small numbers even on the computer. Additionally, the algebraic closure implies that derivatives can also be calculated and proven to exist in many cases of branch points, and even under the presence of non-differentiable pieces as long as the overall graph represents a differentiable function [19].

The method of [4] is limited to functions of one variable. This allowed the treatment of partial derivatives by repeated evaluation, but no assessment of differentiability over entire multidimensional neighborhoods. Below the structures are extended to the canonical treatment of v independent differentials. One obtains functions from Q^v into R such that the sequence formed by adding the v rational numbers $q_1 + ... + q_n$ at nonzero entries forms a strictly diverging sequence. Most favorable properties are retained, in particular the structure still admits roots for most numbers, and similar to above it is now possible to obtain partial derivatives of any order as the respective difference quotients with respect to different infinitesimals. Again the methods also hold under branching and in the presence of certain non-differentiable pieces.

In a practical implementation requiring mathematical rigor, it is also necessary to account for possible floating point errors. This can be achieved by embedding the structures into functions from Q^v into the set of floating point intervals. Using this method, the presence/absence of critical points can be detected in a rigorous way if appropriate.

In practice, first and higher order partial derivatives are often used in the framework of sensitivity analysis. We note that it is possible to develop a further extension of the previous numbers that even support rigorous bounds of the errors made in a sensitivity representation over a pre-specified domain. This approach provides a generalization of the Taylor Model approach [14, 15] that now also rigorously accounts for non-differentiability. For reasons of space, we have to refer to a forthcoming paper for details.

Altogether, the newly proposed data type unites differentiation data types for any order and number of variables, interval methods, the sharper approach of Taylor models, and has the ability to assert differentiability even if not all pieces are differentiable.

45.2 The Levi-Civita Numbers $\mathcal{R}_v$

We begin our discussion with a group of definitions.

Definition 1 ***(Diagonal Finiteness)*** *We say a subset M of the rational v-tuples Q^v is diagonally finite if for any m, there are only finitely many elements $(q_1, ..., q_v) \in M$ that satisfy $q_1 + ... + q_v \leq m$.*

Diagonal finiteness is a generalization of the left-finiteness introduced for the Levi-Civita numbers [4]. In a similar way, we now introduce a set of numbers as follows:

Definition 2 ***(Extended Levi-Civita Numbers)*** *We define the extended Levi-Civita numbers $\mathcal{R}_v$ to be the set of functions x from Q^v into the real numbers R such that the points $\mathbf{q} = (q_1, ..., q_v)$ in the support of x, i.e. those points that satisfy $x[\mathbf{q}] \neq 0$, are diagonally finite.*

For two extended Levi-Civita Numbers x and y in $\mathcal{R}_v$, we define $(x+y)$ and $(x \cdot y)$ via

$$(x+y)[\mathbf{q}] = x[\mathbf{q}] + y[\mathbf{q}] \tag{45.1}$$

$$(x \cdot y)[\mathbf{q}] = \sum_{\mathbf{q}=\mathbf{r}+\mathbf{s}} x[\mathbf{r}] \cdot y[\mathbf{s}] \tag{45.2}$$

where for a given $\mathbf{q}$, the sum is carried over only those $\mathbf{r}$ and $\mathbf{s}$ that satisfy $\mathbf{q} = \mathbf{r} + \mathbf{s}$ and $x[\mathbf{r}] \neq 0$, $y[\mathbf{s}] \neq 0$, of which there are only finitely many.

Definition 3 *We define the interval Levi-Civita Numbers $\mathcal{R}_v^I$ to be the set of functions from Q^v into intervals with diagonally finite support. The arithmetic for the intervals (45.1) and (45.2) follows standard practice ([1, 11, 16, 17, 18]). The arithmetic for the $\mathbf{q}$ is performed as exact rational arithmetic with check for overflow beyond a maximum q_o and underflow beyond a minimum q_u.*

For a generalization of the Taylor model approach [13, 14, 15], one would map into a product space (I_1, I_2) of two intervals, the first of which agrees with the previous one, and the second of which would represent a bound of the remainder error for truncation at the respective order.

We note that implementation of the Levi-Civita numbers $\mathcal{R}_v$ is a straightforward generalization of high-order multivariable Taylor methods; to any given Levi-Civita number, one decides a depth l to which support points are kept. For the finitely many support points below l, the common denominator is determined, and the numerators are manipulated just as in high-order multivariate automatic differentiation [3].

Definition 4 ***(Degree, Finiteness, Core and Pyramidality)*** *To a number nonzero $x \in \mathcal{R}_v$, we define the degree*

$$\lambda(x) = \min\{q_1 + ... + q_v \mid x[q_1, ..., q_v] \neq 0\}, \tag{45.3}$$

i.e. the minimum of the support of x. We also define $\lambda(0) = +\infty$. If $\lambda(x) = 0$, > 0, or < 0, we say x is finite, infinitely small, or infinitely large, respectively. Any point $(q_1^c, ..., q_v^c)$ that satisfies $x[q_1^c, ..., q_v^c] \neq 0$ and $q_1 + ... + q_v = \lambda(x)$ is called a core of x. We say x is pyramidal if it has only one core.

The most important case, representing the case of conventional automatic differentiation, corresponds to finite numbers with support points $(q_1, ..., q_v)$ that satisfy $q_i \geq 0$. These numbers are pyramidal with core $(0, ..., 0)$.

Definition 5 ***(Positive Numbers, Ordering)*** *Let $x \in \mathcal{R}_v$ be pyramidal, and let $\mathbf{q}_c$ be the core of x. We say x is positive if x if $x[\mathbf{q}_c] > 0$, and negative if $x[\mathbf{q}_c] < 0$. We say $x > y$ if $(x - y)$ is pyramidal and positive.*

Apparently the ordering is only partial, as some numbers are neither positive nor negative. Altogether, we arrive at the following situation.

Theorem 1 ***(Structure of R_v)*** *The Levi-Civita Numbers $\mathcal{R}_v$ form an extension of the real numbers R by virtue of the embedding*

$$r \in R \to x \in \mathcal{R}_v \text{ with } x[\mathbf{0}] = r \text{ and } x[\mathbf{q}] = 0 \text{ for } \mathbf{q} \neq \mathbf{0} \tag{45.4}$$

which preserves arithmetic and order of R.

They form a partially ordered real algebra, and the order is compatible with the algebraic operations. The order is non-Archimedean; for example, the elements d_v given by

$$d_v[e_v] = 1, \text{ and } 0 \text{ else} \tag{45.5}$$

where $\mathbf{e}_v$ is the v-th Cartesian basis vector, are infinitely small, and their multiplicative inverses are infinitely large.

Pyramidal elements admit inverses and odd roots, and positive pyramidal elements admit even roots.

The proofs of these assertions follow rather parallel to the corresponding ones in the case of $\mathcal{R}$ presented in [4]. Specifically, let $(q_1^c, ..., q_v^c)$ be the (unique) core of the number x. We then write

$$x = x[q_1^c, ..., q_v^c] \cdot (1 + \tilde{x})$$

and observe that by the definition of pyramidality, $\lambda(\tilde{x}) > 0$. This allows for the treatment of $(1 + \tilde{x})$ through a (converging) power series for the root, in a similar way as in [4].

45.3 Intrinsic Functions for Finite and Infinite Arguments

In [4, 5, 20], the common intrinsic functions were extended to $\mathcal{R}$ based on a theory of power series on $\mathcal{R}$ and their power series representation. The concept of weak convergence on which the theory of power series on $\mathcal{R}$ is based can be extended directly to $\mathcal{R}_v$ in the same way as before based on the family of seminorms

$$||x||_r = \max_{q_1 + \ldots + q_v \leq r} (|x[q_1, ..., q_v]|).$$

In this way, one obtains that the conventional power series all converge in the same way as in $\mathcal{R}$, and thus all conventional intrinsic functions are available in the same fashion.

However, the canonical treatment of intrinsics via power series only applies to arguments within the finite radius of convergence, and thus for example does not cover infinitely large numbers, limiting their use for cases where also in all intermediate computations, the arguments stay finite, and precludes the study of cases like $f(x) = x^2 \cdot \log(x)$ $(x \neq 0)$, $f(0) = 0$ at the origin. Within the framework of $\mathcal{R}$ and $\mathcal{R}_v$, it is however also desirable to assign values to intrinsics for non-finite numbers.

We now extend the common intrinsic functions to R_v beyond the domains in which they can be represented by power series by representing their asymptotic behavior in terms of intervals, and as a consequence we will be able to study cases like $x^2 \log(x)$.

Definition 6 ***(Intrinsic Functions)***

$$\sin(x) = \begin{cases} \sum_{i=0}^{\infty} (-1)^{i+1} \frac{x^{2i+1}}{(2i+1)!} & \lambda(x) \geq 0 \\ [-1,1] & x \textit{ infinitely large} \end{cases}$$

$$\exp(x) = \begin{cases} \sum_{i=0}^{\infty} \frac{x^i}{i!} & \lambda(x) \geq 0 \\ [0,\infty] \cdot d_1^{q_o} \cdot \ldots \cdot d_v^{q_o} & x \textit{ positive and infinitely large} \\ [0,\infty] \cdot d_1^{q_u} \cdot \ldots \cdot d_v^{q_u} & x \textit{ negative and infinitely large} \end{cases}$$

$$\log(x) = \begin{cases} \log(\mathrm{Re}(x)) + \sum_{i=0}^{\infty} \frac{(x-\mathrm{Re}(x))^i}{i \cdot \mathrm{Re}(x)^i} & x \textit{ positive and } \lambda(x) = 0 \\ [-\infty, 0] & x \textit{ positive and infinitely small} \\ [0,\infty] & x \textit{ positive and infinitely large} \end{cases}$$

where q_o and q_u are the largest and smallest representable rational numbers introduced above, and $\mathrm{Re}(x)$ *denotes the real part* $x[0,...,0]$ *introduced above.*

Apparently these definitions have a slight similarity with assigning things such as "NAN" etc. for floating point exceptions, where here in addition it is possible to distinguish different speed of divergence by assigning "different kinds of infinity." The sine stays bounded asymptotically, the exponential diverges faster than any rational power, and the logarithm, while diverging, does so slower than any rational power. With these and similar definitions for the other intrinsics not explicitly listed, it is now possible to treat many different kinds of branch points and exceptions, as exemplified below.

We have similar theorems as in $\mathcal{R}$:

Theorem 2 ***(Continuity and Differentiability)***

The function f is infinitely often partially differentiable at $\mathbf{x}$ *in R^v, and is totally differentiable if it can be evaluated for some true interval inclusion that has* $\mathbf{x}$ *in its interior without tripping a code branch during its evaluation. In this case, all partial derivatives of f can be evaluated up to* *infinitely small error by the corresponding divided difference formulae with infinitely small differences.*

The function f is continuous in the direction of unit vector $\mathbf{e}_v$ if it can be evaluated for an interval that may have $\mathbf{x}$ on its boundary, as well as $\mathbf{x} \pm d_v$, and if the evaluation $f(\mathbf{x})$ and $f(\mathbf{x} \pm d_v)$ agree up to infinitely small error, even if $\mathbf{x}$ and $\mathbf{x} \pm d_v$ lie on different code branches.

The function f is partially differentiable in direction $\mathbf{e}_v$ if it can be evaluated for an interval that may have $\mathbf{x}$ on its boundary as well as $\mathbf{x} \pm d_v$ and if the difference quotients $(f(x+d_v) - f(x))/d_v$ and $(f(x) - f(x-d_v))/d_v$ agree up to infinitely small error, even if $\mathbf{x}$ and $\mathbf{x} \pm d_v$ lie on different code branches.

The function f is higher-order partially differentiable in direction $\mathbf{e}_v$ if it can be evaluated for an interval that may have $\mathbf{x}$ on its boundary as well as $\mathbf{x} + \sum c_j d_j$ for suitable coefficients c_j and if the corresponding higher-order difference quotients agree up to infinitely small error, even if $\mathbf{x}$ and the $\mathbf{x} + \sum c_j d_j$ lie on different code branches.

In this view, the treatment of branch conditions in IF structures etc. through intervals I are to be viewed in a set theoretical sense with a partial order $I_1 < I_2$ if $x_1 < x_2$ for all $x_1 \in I_1$ and $x_2 \in I_2$. This is similar to other interval-based treatment of branches [2, 10]. In the first case, if there is a true inclusion that is "safe" for evaluation, this already implies that all the arguments in the divided difference schemes can be evaluated. The other cases parallel the situation in $\mathcal{R}$ [4], except that now also partial derivatives are possible, and there is no reliance on the accuracy of floating point arithmetic.

45.4 Critical Points of Computer Functions and Their Distribution

It has been widely recognized [2, 8, 10] that the detection of critical points during code execution is a difficult task without relying on verified techniques such as interval methods or the interval-Levi Civita numbers introduced above. However, it is frequently at least assumed that any critical points that can occur in computer code to be processed by automatic differentiation tools through branching or inherent singularities of intrinsic functions are isolated and hence easily distinguishable. However, this intuitive notion is indeed rather misleading, as we shall illustrate with a few examples. In particular, it is possible to construct computer functions with critical points at nearly every computer number in a certain range. Consider for example the function f and its critical points

$$f(x) = 1/\sin(\exp(1/x)) \tag{45.6}$$

$$x_n^{crit} = 1/\ln(n\pi) \tag{45.7}$$

Apparently the critical points of f are spread all over the interval $[0, 1]$, and they get closer together as we approach 0. Table 45.1 shows the distribution of the critical points of f in $(0, 1)$. Furthermore, any point in the interval

TABLE 45.1. Distribution of the critical points of f in $(0, 1)$

Subinterval	Number of critical points of f
$\left(\frac{1}{16}, \frac{1}{8}\right]$	$2,827,587$
$\left(\frac{1}{32}, \frac{1}{16}\right]$	$25,134,688,039,947$
$\left(\frac{1}{64}, \frac{1}{32}\right]$	more than 1.9×10^{27}

$(0, 0.03]$ is less than 10^{-16} away from a critical point of f. In passing we note that the actual size of the interval is rather immaterial since through a simple appropriate re-scaling we can apparently obtain the same result in $(0, 1]$ rather than in $(0, 0.03]$. Between any two computer numbers in $(0, 0.03]$, there lies a critical point of f, and thus critical points cannot computationally be distinguished from non-critical points.

To illustrate this point, we investigate what happens if we evaluate f at the critical point $x = 1/\ln\left(10^{15}\pi\right) \in (0, 0.03]$. We first evaluate x numerically with 14, 16, 18, 20 digits precision and then we evaluate f with the same precision, using Maple. The results are given in Table 45.2 The results

TABLE 45.2. Critical points cannot computationally be distinguished from non-critical points

Digits	$x = 1/\ln\left(10^{15}\pi\right)$	$f(x)$
14	$2.802\,415\,189\,056\,6 \times 10^{-2}$	$-1.410\,081\,582\,760\,7$
16	$2.802\,415\,189\,056\,638 \times 10^{-2}$	$1.000\,812\,471\,855\,461$
18	$2.802\,415\,189\,056\,637\,88 \times 10^{-2}$	$-9.237\,866\,180\,822\,379\,43$
20	$2.802\,415\,189\,056\,637\,879\,1 \times 10^{-2}$	$-791.989\,471\,416\,785\,333\,8$

apparently do not reflect the fact that x is a critical point of f, but each individual evaluation looks rather inconspicuous. The situation is rather similar for other choices of critical points. With only minor additional effort, the situation can become even much more complicated; for example, consider the function $g(x)$ with critical points $x^c_{m,n}$ given by

$$g(x) = 1/\sin(f(x)) = 1/\sin\left(1/\sin(\exp(1/x))\right);$$

$$x^c_{m,n} = \frac{1}{\ln\left(n\pi + \arcsin\left(\frac{1}{m\pi}\right)\right)} \quad \text{where } m, n \text{ are nonzero integers.}$$

Thus, to every critical point of f in $(0, 1)$ there corresponds a whole family of infinitely many critical points of g. All these difficulties that are not transparent in a floating point environment are overcome more or less directly when working in any kind of rigorous interval environment. This is also the case for the interval Levi-Civita numbers R_v^I.

45.5 Examples

To conclude, we illustrate the behavior of the method for various pathological cases, some of which have been considered previously [2, 7, 8, 9, 10, 12].

Problem 1 *$f_1(x) = 1$ for $x^2 > 2$, and 0 otherwise, at $x = \sqrt{2}$.*

Depending on whether the floating point result of the calculation of $\sqrt{2}$ is above or below the true value of $\sqrt{2}$, using floating point methods, the value of f_1 would be chosen as 1 or 0. The derivative would be returned as 0. Evaluation with interval arithmetic would recognize the branch and conclude that it cannot decide differentiability. Depending on implementation of the IF structure and the possibility of launching branch threads, an implementation may even return the result $[0, 1]$. Evaluation in $\mathcal{R}_v^1$ would recognize the branch and conclude that it cannot decide differentiability.

Problem 2 *$f_2(x) = x \cdot \exp(1/x)$ for $x \neq 0$ and $f_2(0) = 0$, at $x = 0$*

Evaluation $\mathcal{R}_v^1$ for $f(0)$ and $f(0+d)$ yields $[0, 0] \cdot d^0$ and $d \cdot [0, \infty] \cdot d_1^{q_o} \cdot \ldots \cdot d_v^{q_o}$. Since $f(0)$ and $f(0 + d)$ differ by more than an infinitely small amount, we conclude that f_2 is discontinuous at the origin.

Problem 3 *$f_3(x) = x \cdot \sin(1/\sqrt{|x|})$ for $x \neq 0$ and $f_3(0) = 0$, at $x = 0$*

Since $f_3(0) = 0$, $f_3(-d) = -d \cdot [-1, 1]$, $f_3(d) = d \cdot [-1, 1]$ all differ only by infinitely small amounts, we conclude f_3 is continuous at 0. Since $(f_3\,(d) - f_3(0))/d = [-1, 1]$ is finite, we know f is not differentiable at 0.

Problem 4 *$f_4(x) = |x|^{3/2} \cdot \log(|x|)$ for $x \neq 0$ and $f_4(0) = 0$, at $x = 0$*

Since $f_4(0) = 0$, $f_4(-d) = d^{3/2} \cdot [-\infty, 0]$, $f_4(d) = d^{3/2} \cdot [-\infty, 0]$ all differ only by infinitely small amounts, we conclude f_4 is continuous at 0. Since $(f_4\,(d) - f_4(0))/d = (f_4\,(0) - f_4(-d))/d = d^{1/2} \cdot [-\infty, 0]$ are equal and thus differ by not more than infinitely small amounts, f_4 is differentiable at 0.

Problem 5 *$f_5(x, y) = 0$ if $x^2 = y$ and $(x, y) \neq (0, 0)$, $f_5(0, 0) = 2$, and $f_5(x, y) = (2x + 1)(|y| + 2)$ otherwise, at $x = y = 0$.*

Evaluating the function at any small interval containing the origin triggers a code branch, avoiding the affirmative answer that the function is totally differentiable at the origin. Moreover, if within the implementation of code transformation, all branches can be followed, it is concluded that the resulting interval has width of at least $[0, 2]$, and hence the function is concluded not to be totally differentiable. Testing partial differentiability in the x direction, we have $f_5(0, 0) = 2$, $f_5(d_x, 0) = 2 + 4d_x$, and $f_5(-d_x, 0) =$

$2-4d_x$. Hence $(f_5(d_x,0)-f_5(0,0))/d_x = (f_5(-d_x,0)-f_5(0,0))/(-d_x) = 4$, and we conclude that f is partially differentiable in the x direction with derivative 4. On the other hand, evaluating $f_5(0,0) = 2$, $f_5(0,d_y) = 2+d_y$, and $f_5(0,-d_y) = 2 + d_y$, we have $(f_5(0,d_y) - f_5(0,0))/d_y = 1$, while $(f_5(0,-d_y) - f_5(0,0))/(-d_y) = -1$, and so we conclude f_5 is not partially differentiable in the y direction.

Acknowledgments: For performing the calculations on the distribution of critical points of the example functions I would like to thank Khodr Shamseddine. For various fruitful conversations about critical points, I would like to thank George Corliss and Baker Kearfott.

[1] Götz Alefeld and Jürgen Herzberger. *Introduction to Interval Computations.* Academic Press, New York, 1983.

[2] T. Beck and Herbert Fischer. The IF-problem in automatic differentiation. *Journal of Comput. Appl. Math.*, 50:119–131, 1995.

[3] Martin Berz. Forward algorithms for high orders and many variables with application to beam physics. In Andreas Griewank and George F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 147–156. SIAM, Philadelphia, Penn., 1991.

[4] Martin Berz. Calculus and numerics on Levi-Civita fields. In Berz et al. [6], pages 19–35.

[5] Martin Berz. Analytical and computational methods for the Levi-Civita fields. In *Proc. Sixth International Conference on Nonarchimedean Analysis*, pages 21–34. Marcel Dekker, New York, 2000.

[6] Martin Berz, Christian Bischof, George Corliss, and Andreas Griewank, editors. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, Penn., 1996.

[7] Andreas Griewank. Automatic directional differentiation of nonsmooth composite functions. In *Recent Developments in Optimization, French-German Conference on Optimization*, Dijon, 1994.

[8] Andreas Griewank. Private communication, 2000.

[9] Eric Hassold and André Galligo. Automatic differentiation applied to nonsmooth convex optimization. In Berz et al. [6], pages 287–297.

[10] R. Baker Kearfott. Automatic differentiation of conditional branches in an operator overloading context. In Berz et al. [6], pages 75–81.

[11] Ulrich W. Kulisch and Willard L. Miranker. *Computer Arithmetic in Theory and Practice.* Academic Press, New York, 1981.

[12] Claude Lemaréchal and R. Mifflin. Nonsmooth optimization. In *Proceedings of a IIASA Workshop*, Oxford, 1977. Pergamon Press.

[13] Kyoko Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators.* PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998. Also MSUCL-1093 and `bt.nscl.msu.edu/papers-cgi/display.pl?name=makinophd`.

[14] Kyoko Makino and Martin Berz. Remainder differential algebras and their applications. In Berz et al. [6], pages 63–74.

[15] Kyoko Makino and Martin Berz. New applications of Taylor model methods. In George Corliss, Christèle Faure, Andreas Griewank, Laurent Hascoët, and Uwe Naumann, editors, *Automatic Differentiation: From Simulation to Optimization*, Computer and Information Science, chapter 43, pages 359–364. Springer, New York, 2001.

[16] Ramon E. Moore. *Interval Analysis.* Prentice Hall, Englewood Cliffs, N.J., 1966.

[17] Ramon E. Moore. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, Penn., 1979.

[18] Ramon E. Moore, editor. *Reliability in Computing: The Role of Interval Methods in Scientific Computing.* Academic Press, San Diego, 1988.

[19] Khodr Shamseddine and Martin Berz. Exception handling in derivative computation with nonarchimedean calculus. In Berz et al. [6], pages 37–51.

[20] Khodr Shamseddine and Martin Berz. Convergence on the Levi-Civita field and study of power series. In *Proc. Sixth International Conference on Nonarchimedean Analysis*, pages 283–299. Marcel Dekker, New York, 2000.

Index