Institut für Angewandte Mathematik und Statistik

Automatic Differentiation: A Structure-Exploiting Forward Mode with Almost Optimal Complexity for Kantorovič Trees

Michael Ulbrich and Stefan Ulbrich

Technical Report Nr. IAMS1996.1TUM

January 1996

Technische Universität München

Automatic Differentiation:
A Structure-Exploiting Forward Mode with
Almost Optimal Complexity for Kantorovič Trees

MICHAEL ULBRICH AND STEFAN ULBRICH

Institut für Angewandte Mathematik und Statistik
Technische Universität München

Dedicated to Klaus Ritter on his 60th birthday.

Abstract. A structure-exploiting forward mode is discussed that achieves almost optimal complexity for functions given by Kantorovič trees. It is based on approriate representations of the gradient and the Hessian. After a brief exposition of the forward and reverse mode of automatic differentiation for derivatives up to second order and compact proofs of their complexities, the new forward mode is presented and analyzed. It is shown that in the case of functions $f : \mathbb{R}^n \to \mathbb{R}$ with a tree as Kantorovič graph the algorithm is only $O(\ln(n))$ times as expensive as the reverse mode. Except for the fact that the new method is a very efficient implementation of the forward mode, it can be used to significantly reduce the length of characterizing sequences before applying the memory expensive reverse mode. For the Hessian all discussed algorithms are shown to be efficiently parallelizable. Some numerical examples confirm the advantages of the new forward mode.

Keywords. automatic differentiation, characterizing sequence, code list, forward mode, reverse mode, Kantorovič graph, Kantorovič tree, time complexity, parallelization.

1. Introduction

The paper is organized as follows. In Section 2 we introduce the basic definition of a characterizing sequence (or code list), which describes how a function $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ can be evaluated at the point x by applying atomic unary and binary operations. The characterizing sequence can be converted into a corresponding Kantorovič graph. In this paper, characterizing sequences with a tree as Kantorovič graph are of special interest. In Section 3 we establish notations of computational complexity and collect some results concerning the complexity of partial derivatives. Section 4 introduces the basic idea of automatic differentiation of characterizing sequences and describes the forward mode for first and second order derivatives. The complexity of forward differentiation is treated in Section 5. Sections 6 and 7 deal with the reverse mode and include a new compact proof of its complexity. In Section 8 we describe the reverse mode for the Hessian and proof a complexity result. Section 9 contains the main part of this paper. Here we introduce a new structure-exploiting forward mode and show that it achieves almost optimal complexity for characterizing sequences with a tree as Kantorovič graph. The complexity considerations base on the observation that for Kantorovič trees, there exists a crucial relation between the total number of atomic operations necessary to compute $f(x)$, $\nabla f(x)$ (and $\nabla^2 f(x)$) and the total number of 'non-trivially generated' nonzero components in temporary gradients, which is shown to grow with order $O(\ln(n)\#(f,S))$. Here $\#(f,S)$ denotes the computational costs to calculate $f(x)$ by means of the characterizing sequence S. In Section 10 some applications of the new method are proposed. A data structure for vectors and matrices tailored to the new algorithm is presented in Section 11. Section 12 briefly discusses how the methods of automatic differentiation can be efficiently parallelized. Some numerical results that show the advantages of the new forward mode can be found in Section 13.

2. Characterizing Sequences and Kantorovič Graphs

Let $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, $n \in \mathbb{N}$, be a twice differentiable function on an open set D. For $x \in D$, we denote by $\nabla f(x) \in \mathbb{R}^n$ the gradient (a column vector) and by $\nabla^2 f(x) \in \mathbb{R}^{n\times n}$ the Hessian of f. In the following we will deal with functions f, which can be expressed as composition of elementary unary operations $\varphi \in \mathcal{U}$, binary operations $\beta \in \mathcal{B}$, the canonical projections $\pi_i : x \longmapsto x_i$ and constant functions $x \longmapsto c \in \mathcal{C} = \mathbb{R}$. Functions of the above type can be evaluated in an algorithmic manner by characterizing sequences.

Definition 2.1 (characterizing sequence, code list)

Set $I = \{1,\ldots,m\}$, $J = \{1,\ldots,n+m-1\}$. The vector

$$S = ((\omega_i, k_i, l_i))_{i\in I} \in \Big((\mathcal{C}\times\{0\}^2) \cup (\mathcal{U}\times J\times\{0\}) \cup (\mathcal{B}\times J^2)\Big)^m ,$$

$m \in \mathbb{N}$, is called a (connected) *charcterizing sequence* (code list) for f, iff

a) $k_i < i+n$, $l_i < i+n$, $\displaystyle\bigcup_{i=1}^{m}\{k_i,l_i\} \supset \{n+1,\ldots,n+m-1\}$

b) $f(x) = f_{m+n}$ for $x \in D$ and

$f_j = x_j, \quad j = 1,\ldots,n$

$$f_{i+n} = \begin{cases} \omega_i & , \ \omega_i \in \mathcal{C} \\ \omega_i(f_{k_i}) & , \ \omega_i \in \mathcal{U} \\ \omega_i(f_{k_i}, f_{l_i}) & , \ \omega_i \in \mathcal{B} \end{cases} \quad , \quad i = 1,\ldots,m$$

□

Each characterizing sequence describes a specific way to evaluate the corresponding *Kantorovič graph.* This directed labeled graph $G = (N,E)$ can be obtained as follows:

The set of nodes is given by

$$N = \bigcup_{i=1}^{m}\{k_i,l_i\} \cup \{n+m\} \setminus \{0\} .$$

Each node $j \in N$ is labeled by a labeling function λ:

$$\lambda(j) = \begin{cases} x_j & , \ 1 \le j \le n \\ \omega_{j-n} & , \ n < j \le n+m \end{cases}$$

The set $E \subset N^2 \times \{1,2\}$ of directed labeled edges is given by

$$E = \{(k_i, i+n, 1);\ k_i > 0,\ 1 \le i \le m\} \cup \{(l_i, i+n, 2);\ l_i > 0,\ 1 \le i \le m\}\,.$$

The edge $e = (j_1, j_2, \mu) \in E$ is directed from node j_1 to j_2. The label μ distinguishes between first and second operand of a binary operation. e is called *in-edge* of node j_2 and *out-edge* of j_1. A node without in-edge is called *leaf.*

Remark 2.2

- Leaves are always labeled by x_i or $c \in \mathcal{C}$.
- G is connected.
- G is cycle-free, because each edge $(j_1, j_2, \mu) \in E$ satisfies $j_1 < j_2$ and thus the nodes along a directed path are strictly increasing.
- In general, different characteristic sequences may lead to equivalent Kantorovič graphs.

□

Due to Definition 2.1, $r = n + m$ is the only node without out-edge. If all nodes with more than one out-edge are leaves, then G is called *Kantorovič tree* with *root* r. In fact, a tree in the usual sense can be obtained by splitting a leaf with k out-edges into k different leaves with the same label and one out-edge. In this case, each temporary result f_i, $i > n$, is used for the calculation of exactly one f_j, $j > i$.

3. Complexity of Partial Derivatives

In the following, the binary operations are given by

$$\mathcal{B} = \left\{\beta : D_\beta \subset \mathrm{I\!R}^2 \longrightarrow \mathrm{I\!R}\,;\ \beta(a,b) = a \otimes b,\ \otimes \in \{+, -, \cdot, /\}\right\} \tag{1}$$

For the analysis of time complexity we assume that all unary operations $\omega \in \mathcal{U}$, their derivatives ω', ω'' and the binary operations are *atomic* and can be executed in one time unit. Evaluations of constant functions and their vanishing derivatives of arbitrary order are free. Further, we don't count sign changes. Hence, the time complexity for computing $f(x)$ is proportional to the number of atomic operations.

Definition 3.1 Let S be a characterizing sequence for f of length m, then

$$\#(f, S) := |\{i\,;\ 1 \le i \le m,\ \omega_i \notin \mathcal{C}\}|$$

denotes the number of atomic operations to evaluate S for a given x. The minimal operation count is denoted by $\#(f)$:

$$\#(f) := \min\left\{\#(f, S)\,;\ S \text{ is a characterizing sequence for } f\right\}\,.$$

If A is an algorithm to compute $f(x)$ and $\nabla^k f(x)$, $k = 1, \ldots, s$, by means of S, $\#(f, \nabla f, \ldots, \nabla^s f, S, A)$ denotes the number of necessary atomic operations.

Finally, we set

$$\#(f, \nabla f, \ldots, \nabla^s f, S) := \min\{\#(f, \nabla f, \ldots, \nabla^s f, S, A)\,;$$
$$A \text{ computes } f(x), \ldots, \nabla^s f(x) \text{ for given } x, S\}\,,$$

$$\#(f, \nabla f, \ldots, \nabla^s f) := \min\{\#(f, \nabla f, \ldots, \nabla^s f, S)\,;\ S \text{ char. sequence for } f\}\,.$$

□

The following basic result gives a bound for $\#(f, \nabla f)$ in terms of $\#(f)$.

Theorem 3.2 (W. Baur, V. Strassen, 1983)
Let $f : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ be a rational function. Then the following inequality holds:

$$\#(f, \nabla f) \;\leq\; 4\,\#(f)$$

Proof: See [1].

A similar result can be found in [3]: H. Fischer and H. Warsitz propose an algorithm $RM.G$ with the following property:

Theorem 3.3 (H. Fischer, H. Warsitz, 1995)
Let $f : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ be a rational function and S an arbitrary characterizing sequence for f. Then:

$$\#(f, \nabla f, S, RM.G) \leq 4\,\#(f, S)$$

Remark 3.4 In Section 6 we will introduce a simple alternative to algorithm $RM.G$, which also works for $\mathcal{U} \neq \emptyset$, and give a proof of the above theorem.

□

4. Automatic Differentiation

For a representative overview of the subject see [2], [5], [7], and the references therein. The main idea consists in differentiating the characterizing sequence. Let $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ be a twice differentiable function given by S.

Characterizing sequence for the function f:

$j = 1, \ldots, n$: $\quad f_j = x_j$

$i = 1, \ldots, m$:

$$f_{i+n} = \begin{cases} \omega_i & ,\ \omega_i \in \mathcal{C} \\ \omega_i(f_{k_i}) & ,\ \omega_i \in \mathcal{U} \\ \omega_i(f_{k_i}, f_{l_i}) & ,\ \omega_i \in \mathcal{B} \end{cases}$$

Sequence for the gradient ∇f:

$j = 1, \ldots, n$: $\quad g_j = e_j = (\delta_{jk})_{1 \le k \le n}$

$i = 1, \ldots, m$:

$$g_{i+n} = \begin{cases} 0 & , \quad \omega_i \in \mathcal{C} \\ \omega_i'(f_{k_i})\, g_{k_i} & , \quad \omega_i \in \mathcal{U} \\ \partial_a \omega_i(f_{k_i}, f_{l_i})\, g_{k_i} + \partial_b \omega_i(f_{k_i}, f_{l_i})\, g_{l_i} & , \quad \omega_i \in \mathcal{B} \end{cases}$$

Sequence for the Hessian $\nabla^2 f$:

$j = 1, \ldots, n$: $\quad H_j = 0$

$i = 1, \ldots, m$:

$$H_{i+n} = \begin{cases} 0 & , \quad \omega_i \in \mathcal{C} \\ \omega_i'(f_{k_i})\, H_{k_i} + \omega_i''(f_{k_i})\, g_{k_i} g_{k_i}^T & , \quad \omega_i \in \mathcal{U} \\ \partial_a \omega_i(f_{k_i}, f_{l_i})\, H_{k_i} + \partial_b \omega_i(f_{k_i}, f_{l_i})\, H_{l_i} & \\ \quad + (g_{k_i}, g_{l_i})\, \nabla^2 \omega_i(f_{k_i}, f_{l_i})\, (g_{k_i}, g_{l_i})^T & , \quad \omega_i \in \mathcal{B} \end{cases}$$

Now the values $f(x) = f_{n+m}$, $\nabla f(x) = g_{n+m}$ and $\nabla^2 f(x) = H_{n+m}$ can be obtained in a straightforward way by evaluating the above sequences in the order

$$f_j, g_j, H_j, \quad j = 1, \ldots, n+m\,.$$

This method is called *forward mode*.

5. Complexity of the Forward Mode

The following tables show the number of *additional* atomic operations for the calculation of g_{i+n} and H_{i+n}, respectively. For the Hessian, we use the symmetry and results already computed for the gradient. The resulting algorithm for computing $f(x)$, $\nabla f(x)$ (and $\nabla^2 f(x)$) will be called F_g (F_H). We set $j = i + n$.

ω_i	Function	Gradient	Ops. for g_j
$\varphi(\cdot)$	$f_j = \varphi(f_{k_i})$	$g_j = \varphi'(f_{k_i})\, g_{k_i}$	$n+1$
$\pm$	$f_j = f_{k_i} \pm f_{l_i}$	$g_j = g_{k_i} \pm g_{l_i}$	n
$\cdot$	$f_j = f_{k_i} \cdot f_{l_i}$	$g_j = f_{l_i}\, g_{k_i} + f_{k_i}\, g_{l_i}$	$3n$
$/$	$f_j = f_{k_i} / f_{l_i}$	$g_j = (g_{k_i} - f_j\, g_{l_i}) / f_{l_i}$	$3n$

Table 1: Number of atomic operations for the calculation of g_{i+n}

ω_i	Hessian	Ops. for H_j
$\varphi(\cdot)$	$H_j = \varphi'(f_{k_i})\, H_{k_i} + \varphi''(f_{k_i})\, g_{k_i}\, g_{k_i}^T$	$(3\,n+2)\,(n+1)/2$
$\pm$	$H_j = H_{k_i} \pm H_{l_i}$	$n\,(n+1)/2$
$\cdot$	$H_j = f_{l_i}\, H_{k_i} + f_{k_i}\, H_{l_i} + g_{k_i}\, g_{l_i}^T + g_{l_i}\, g_{k_i}^T$	$n\,(7n+5)/2$
$/$	$H_j = (H_{k_i} - f_j\, H_{l_i} - g_j\, g_{l_i}^T - g_{l_i}\, g_j^T)/f_{l_i}$	$n\,(7n+5)/2$

Table 2: Number of atomic operations for the calculation of H_{i+n}

Theorem 5.1 *Let S be a characterizing sequence for the twice differentiable function $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$. Then the following inequalities hold:*

$$(n+1)\,\#(f,S) \leq \quad \#(f,\nabla f,S,F_g) \quad \leq (3\,n+1)\,\#(f,S)$$

$$\frac{n^2+3n+2}{2}\,\#(f,S) \leq \#(f,\nabla f,\nabla^2 f,S,F_H) \leq \frac{7n^2+11n+2}{2}\,\#(f,S)$$

Proof: According to Table 1, the additional computation of ∇f_{i+n} in step i, $i = 1,\ldots,m$, requires at least n and at most $3n$ operations. To calculate also the Hessian H_{i+n}, $n(n+1)/2$ to $n(7n+5)/2$ additional operations are necessary (see Table 2).

□

6. The Reverse Mode

Let S be a characterizing sequence for f of length m. Based on S, we define the sequence of functions $F_1,\ldots,F_m$ as follows:

$i = 1,\ldots,m-1$:

$$F_i : D_i \subset \mathbb{R}^{n+i-1} \longrightarrow \mathbb{R}^{n+i}, \quad F_i(y_1,\ldots,y_{n+i-1}) = \begin{pmatrix} y_1 \\ \vdots \\ y_{n+i-1} \\ \gamma_i \end{pmatrix}$$

$$F_m : D_m \subset \mathbb{R}^{n+m-1} \longrightarrow \mathbb{R}, \quad F_m(y_1,\ldots,y_{n+m-1}) = \gamma_m$$

with $D_i = \left\{ y \in \mathbb{R}^{n+i-1}\,;\ y_{k_i} \in D_{\omega_i} \text{ if } \omega_i \in \mathcal{U}\,,\ (y_{k_i}, y_{l_i}) \in D_{\omega_i} \text{ if } \omega_i \in \mathcal{B} \right\}$,

$$\gamma_i = \begin{cases} \omega_i & ,\ \omega_i \in \mathcal{C} \\ \omega_i(y_{k_i}) & ,\ \omega_i \in \mathcal{U} \\ \omega_i(y_{k_i}, y_{l_i}) & ,\ \omega_i \in \mathcal{B} \end{cases} \quad , \quad i = 1,\ldots,m\,.$$

We set

$$G_0 := x\ , \quad G_i^T := (f_1,\ldots,f_{n+i})^T = F_i \circ F_{i-1} \circ \cdots \circ F_1(x)\ ,\ i = 1,\ldots,m-1.$$

Differentiating the identity

$$f(x) = f_{n+m} = F_m \circ F_{m-1} \circ \cdots \circ F_1(x)$$

leads to

$$\nabla f(x)^T = DF_m(G_{m-1})\, DF_{m-1}(G_{m-2}) \cdots DF_1(x)\,, \tag{2}$$

where DF denotes the Jacobian of F. The forward mode corresponds to the evaluation of (2) *from the right to the left.* On the other hand, if $f_1, \ldots, f_{n+m}$ are already computed, (2) may also be evaluated *from the left to the right.* This method is called *reverse mode.*

7. Complexity of the Reverse Mode

In detail, (2) has the form

$$\nabla f(x)^T = \begin{pmatrix} \underset{\underset{k_m}{\uparrow}}{\kappa_m} & \underset{\underset{l_m}{\uparrow}}{\lambda_m} \end{pmatrix} \left(\begin{array}{cc} \multicolumn{2}{c}{I_{n+m-2}} \\ \hline \underset{\underset{k_{m-1}}{\uparrow}}{\kappa_{m-1}} & \underset{\underset{l_{m-1}}{\uparrow}}{\lambda_{m-1}} \end{array}\right) \cdots \left(\begin{array}{cc} \multicolumn{2}{c}{I_{n}} \\ \hline \underset{\underset{k_{1}}{\uparrow}}{\kappa_{1}} & \underset{\underset{l_{1}}{\uparrow}}{\lambda_{1}} \end{array}\right)$$

If $k_i = 0$ or $l_i = 0$, ignore the corresponding entry in the last row. The reverse mode is carried out in m steps:

$$v^m := DF_m(G_{m-1})\,, \quad v^i := v^{i+1} DF_i(G_{i-1})\,, \quad i = m-1, \ldots, 1$$

Denoting by v_j^i the j^{th} component of v^i, v^i originates from v^{i+1} by adding $\kappa_i v_{i+n}^{i+1}$ (if $k_i \neq 0$) to the k_i^{th}, $\lambda_i v_{i+n}^{i+1}$ (if $l_i \neq 0$) to the l_i^{th} and deleting the last component. To obtain minimal complexity, we omit the additions, if $v_{k_i}^{i+1} = 0$ or $v_{l_i}^{i+1} = 0$, respectively. Table 3 collects the number of atomic operations to compute v^i from v^{i+1}; we set $j = i + n$ and define $\sigma : \mathbb{R} \longrightarrow \{0, 1\}$ by $\sigma(0) = 0$, $\sigma(\xi) = 1$ for $\xi \neq 0$.

ω_i	Function	κ_i	λ_i	Ops. κ_i, λ_i	Ops. $v^{i+1} \to v^i$
c	$f_j = c$	–	–	0	0
$\varphi(\cdot)$	$f_j = \varphi(f_{k_i})$	$\varphi'(f_{k_i})$	–	1	$2 + \sigma(v_{k_i}^{i+1})$
$\pm$	$f_j = f_{k_i} \pm f_{l_i}$	1	± 1	0	$\sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})$
$\cdot$	$f_j = f_{k_i} \cdot f_{l_i}$	f_{l_i}	f_{k_i}	0	$2 + \sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})$
$/$	$f_j = f_{k_i} / f_{l_i}$	$1/f_{l_i}$	$-f_j\, \kappa_i$	2	$2 + \sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})$

Table 3: Operation counts for a step of the reverse mode

The resulting algorithm is called R_g.

Theorem 7.1 *Let $f : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ be a differentiable non-constant function and S an arbitrary characterizing sequence for f of length m. Then:*

$$\#(f, \nabla f, S, R_g) \leq 5\#(f,S) - \left| \bigcup_{i=1}^{m} \{k_i, l_i\} \setminus \{0\} \right| \leq 5\#(f,S) - m \leq 4\#(f,S)$$

Proof: From $f \neq$ const. and Definition 2.1 a) follows $m > 1$, $\omega_m \notin \mathcal{C}$ and (note the dependence on at least one x_j)

$$N := \left| \bigcup_{i=1}^{m} \{k_i, l_i\} \setminus \{0\} \right| \geq m \,. \tag{3}$$

For simplicity, we set $v_0^i = 0$, $i = 1, \ldots, m$. According to Table 3, each of the $\#(f,S) - 1$ steps $v^{i+1} \rightarrow v^i$ with $\omega_i \notin \mathcal{C}$ costs $\leq 2 + \sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})$ operations, those with $\omega_i \in \mathcal{C}$ are free. The computation of v^m requires ≤ 2 operations. Thus, we have

$$\begin{aligned} \#(f, \nabla f, S, R_g) &\leq \#(f,S) + 2 + \sum_{\substack{1 \leq i < m \\ \omega_i \notin \mathcal{C}}} (2 + \sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})) \\ &= 3\#(f,S) + \sum_{\substack{1 \leq i < m \\ \omega_i \notin \mathcal{C}}} (\sigma(v_{k_i}^{i+1}) + \sigma(v_{l_i}^{i+1})) \end{aligned} \tag{4}$$

Since in step $v^{i+1} \rightarrow v^i$ additional non-zero elements can only arise at position k_i and l_i (if > 0), we have

$$\left\{ j\,;\ 1 \leq j \leq n+i-1,\ v_j^i \neq 0 \right\} \subset \left(\bigcup_{j=i}^{m} \{k_j, l_j\} \right) \cap \{1, \ldots, n+i-1\} \,.$$

With (3) and (4) follows the assertion:

$$\begin{aligned} \#(f, \nabla f, S, R_g) &\leq 3\#(f,S) + \Big(2(\#(f,S) - 1) - (N-2) \Big) = 5\#(f,S) - N \\ &\leq 5\#(f,S) - m \leq 4\#(f,S) \end{aligned}$$

□

Remark 7.2 If f is constant, then the theorem remains true for all sequences S which contain a k_i or l_i in $\{1, \ldots, n\}$.

□

8. A Reverse Mode of Optimal Order for the Hessian

The following example demonstrates that there exist functions f with

$$\#(f, \nabla f, \nabla^2 f) \geq cn\#(f) \,, \quad c \geq \frac{1}{2} \text{ independent of } n.$$

Example 8.1 Let $f(x) = \left(\prod_{i=1}^{n} x_i\right)^{-1}$. Then

$$\frac{\partial f(x)}{\partial x_i} = -\frac{f(x)}{x_i} \;, \quad \frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \begin{cases} \dfrac{f(x)}{x_i x_j} & , \; i \neq j \\ \dfrac{2f(x)}{x_i^2} & , \; i = j \end{cases} \quad , \; i,j = 1,\ldots,n$$

and

$$\#(f) = n \;, \quad \#(f, \nabla f, \nabla^2 f) = 3n + \frac{n(n+1)}{2} > \frac{n+1}{2} \, \#(f) \, .$$

$\square$

If S is a characterizing sequence of an arbitrary twice differentiable function f, then $f(x) = f_{n+m}$ and the partial derivative $\partial_{x_k} f(x) = h(x, e_k) = h_{n+m}$ can be computed simultanously using the forward mode:

$j = 1, \ldots, n$: $\quad f_j = x_j, \; h_j = y_j$

$i = 1, \ldots, m$:

$$f_{i+n} = \begin{cases} \omega_i & , \; \omega_i \in \mathcal{C} \\ \omega_i(f_{k_i}) & , \; \omega_i \in \mathcal{U} \\ \omega_i(f_{k_i}, f_{l_i}) & , \; \omega_i \in \mathcal{B} \end{cases}$$

$$h_{i+n} = \begin{cases} 0 & , \; \omega_i \in \mathcal{C} \\ \omega_i'(f_{k_i}) \, h_{k_i} & , \; \omega_i \in \mathcal{U} \\ \partial_a \omega_i(f_{k_i}, f_{l_i}) \, h_{k_i} + \partial_b \omega_i(f_{k_i}, f_{l_i}) \, h_{l_i} & , \; \omega_i \in \mathcal{B} \end{cases}$$

Obviously, the expressions for h_j consist of at most three atomic operations (see Table 1) if we replace $\mathcal{U}$ by

$$\mathcal{U} \cup \{\varphi' \, ; \; \varphi \in \mathcal{U}\} \; .$$

Thus, after deleting unused temporary results f_j (all h_j are needed), we obtain a characterizing sequence S_h of length $m_h \leq 4m$ for $h(x, y)$ and application of the reverse mode to S_h at $(x, y) = (x, e_k)$ yields the k^{th} row of the Hessian. We call the resulting algorithm R_H and get the following result:

Theorem 8.2 *Let $f : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ be a twice differentiable non-constant function and S an arbitrary characterizing sequence for f. Then:*

$$\#(f, \nabla f, \nabla^2 f, S, R_H) \leq 16 \, n \, \#(f, S)$$

Proof: If the computation of f_j requires l operations, then we need $\leq 3l$ operations for h_j. Thus, the computation of $f(x)$ and $\partial_{x_k} f(x) = h(x, e_k)$ by the above forward sequence costs $\leq 4\,\#(f,S)$ operations and consequently

$$\#(h, S_h) \leq 4\,\#(f,S)\,.$$

Since S contains at least one x_j-dependence, S_h contains a y_j-dependence and application of Theorem 7.1 in connection with Remark 7.2 leads to

$$\#(f, \nabla f, \nabla^2 f, S, R_H) \leq 16\,n\,\#(f,S)$$

□

Remark 8.3 Since the f_j are independent of y, they can be computed only once instead of n times. This improved algorithm $\tilde{R}_H$ satisfies

$$\#(f, \nabla f, \nabla^2 f, S, \tilde{R}_H) \leq (15n+1)\,\#(f,S)\,.$$

□

Remark 8.4 The result of Remark 8.3 can also be found in [2]. In [6] M. Iri mentions an algorithm A_H with

$$\#(f, \nabla f, \nabla^2 f, S, A_H) \leq (10n+4)\,\#(f,S)$$

□

9. An Efficient New Forward Mode

The applicability of the reverse mode is severely restricted by the fact that the required memory grows linearly with $\#(f,S)$. A. Griewank overcomes this difficulty by formulating a recursive reverse mode (see [4]) that achieves time-complexity $O\left((c_1 + c_2 \log_2(\#(f,S)))\,\#(f,S)\right)$ and reduces the space-complexity to $O\left(c_3 \log_2(\#(f,S))\right)$. In the following, we will propose new (to the knowledge of the authors) structure-exploiting forward algorithms SF_g and SF_H with complexities

$$\#(f, \nabla f, S, SF_g) \leq 7\#(f,S) - b + (\#(f,S)+1)\ln(\max\{b,1\})$$

and

$$\#(f, \nabla f, \nabla^2 f, S, SF_H) \leq$$
$$\leq 5(b+1)\left(7\#(f,S) - \frac{3}{2}b + (\#(f,S)+1)\ln(\max\{b,1\})\right) + 3b$$

for characterizing sequences with a tree as Kantorovič graph. Here b denotes the number of x_k's on which f depends. To this end, we improve the forward mode by the following modifications:

- For the temporary gradients g_j and Hessians H_j use the representations

$$g_j = \alpha_j v_j \ , \quad H_j = \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j$$

  with $\alpha_j, \beta_j, \gamma_j, \delta_j \in \mathbb{R}$, $\alpha_j, \delta_j \neq 0$, vectors v_j, w_j and matrices M_j.
- Restrict the computations to non-zero elements.
- When two temporary results are combined by a binary atomic operation, obtain the new gradient- and Hessian-representation by copying and modifying the one with the less sparse v_j. Don't copy non-leaves with one out-edge, since they are no more needed and can be overwritten.
- When applying a unary atomic operation to a non-leaf with one out-edge, obtain the new temporary result by modifying the old one.

Definition 9.1 A characterizing sequence S is called *T-sequence* iff the corresponding Kantorovič graph is a Kantorovič tree.

$\square$

Let S be a characterizing sequence for $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ of length m. Obviously, the characterizing subsequence $S_i = (\omega_s, k_s, l_s)_{s \in N_i}$ for the calculation of f_{i+n}, $1 \leq i \leq m$, can be constructed by setting

$$N_i := \bigcup_{k=1}^{i} N_i^k \tag{5}$$

with

$$N_i^1 := \{i\} \ , \quad N_i^{k+1} := \bigcup_{s \in N_i^k} \{k_s - n, l_s - n\} \setminus \{-n, \ldots, 0\} \ , \ k = 1, \ldots, i-1 \, .$$

Denote by b_i the number of x_k's used in S_i, $1 \leq i \leq m$, i.e.

$$b_i := |B_i| \ \text{ with } \ B_i := \begin{cases} \bigcup_{s \in N_i} \{k_s, l_s\} \cap \{1, \ldots, n\} & , \ 1 \leq i \leq m \\ \emptyset & , \ i \leq 0 \end{cases} \tag{6}$$

and define

$$\begin{aligned} q_i &:= b_i - \max\{b_{k_i - n}, b_{l_i - n}\} \\ p_i &:= \begin{cases} q_i & , \ \{k_i, l_i\} \cap \{1, \ldots, n\} = \emptyset \\ 0 & , \ \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

q_i can be seen as the number of fill-ins, when v_{i+n} is calculated by merging the sparser of v_{k_i}, v_{l_i} into the other. p_i counts only fill-ins, if both operands aren't x_k's.

Algorithm $\boldsymbol{SF_g}$:

$j = 1, \ldots, n$:

$\alpha_j = 1,\ v_j = e_j$

$i = 1, \ldots, m$:

$j := i + n$

I) $\omega_i \in \mathcal{C}$:

$$f_j = \omega_i,\ \ g_j = 0 = \alpha_j v_j \ \ \text{with} \ \ \alpha_j = 1,\ v_j = 0.$$

II) $\omega_i \in \mathcal{U}$:

$$f_j = \omega_i(f_{k_i}),\ \ g_j = \omega_i'(f_{k_i}) g_{k_i} = \omega_i'(f_{k_i}) \alpha_{k_i} v_{k_i} = \alpha_j v_j$$

with

1) $k_i \in \{1, \ldots, n\}$: $\alpha_j = 1,\ v_j = \omega_i'(x_{k_i})\, e_{k_i}$.

2) $k_i \notin \{1, \ldots, n\},\ \omega_i'(f_{k_i}) \neq 0$: $\alpha_j = \omega_i'(f_{k_i}) \alpha_{k_i},\ v_j = v_{k_i}$.

3) $k_i \notin \{1, \ldots, n\},\ \omega_i'(f_{k_i}) = 0$: $\alpha_j = 1,\ v_j = 0$.

III) $\omega_i \in \mathcal{B},\ \omega_i(a, b) = a \pm b$:

$$f_j = f_{k_i} \pm f_{l_i},\ \ g_j = g_{k_i} \pm g_{l_i} = \alpha_{k_i} v_{k_i} \pm \alpha_{l_i} v_{l_i} = \alpha_j v_j$$

with

1) $k_i, l_i \in \{1, \ldots, n\}$: $\alpha_j = 1,\ v_j = e_{k_i} \pm e_{l_i}$.

2) $k_i \notin \{1, \ldots, n\},\ l_i \in \{1, \ldots, n\}$: $\alpha_j = \alpha_{k_i},\ v_j = v_{k_i} \pm \dfrac{1}{\alpha_{k_i}} e_{l_i}$.

3) $l_i \notin \{1, \ldots, n\},\ k_i \in \{1, \ldots, n\}$: Analogous to 2).

4) $k_i, l_i \notin \{1, \ldots, n\},\ b_{k_i - n} \geq b_{l_i - n}$: $\alpha_j = \alpha_{k_i},\ v_j = v_{k_i} \pm \dfrac{\alpha_{l_i}}{\alpha_{k_i}} v_{l_i}$.

5) $k_i, l_i \notin \{1, \ldots, n\},\ b_{k_i - n} < b_{l_i - n}$: Analogous to 4).

IV) $\omega_i \in \mathcal{B},\ \omega_i(a, b) = a \cdot b$:

$$f_j = f_{k_i} f_{l_i},\ \ g_j = f_{l_i} g_{k_i} + f_{k_i} g_{l_i} = f_{l_i} \alpha_{k_i} v_{k_i} + f_{k_i} \alpha_{l_i} v_{l_i} = \alpha_j v_j$$

with

1) $k_i, l_i \in \{1, \ldots, n\}$: $\alpha_j = 1,\ v_j = x_{l_i} e_{k_i} + x_{k_i} e_{l_i}$.

2) $k_i \notin \{1,\ldots,n\}$, $l_i \in \{1,\ldots,n\}$, $x_{l_i} \neq 0$: $\alpha_j = x_{l_i}\alpha_{k_i}$, $v_j = v_{k_i} + \dfrac{f_{k_i}}{\alpha_j} e_{l_i}$.

3) $k_i \notin \{1,\ldots,n\}$, $l_i \in \{1,\ldots,n\}$, $x_{l_i} = 0$: $\alpha_j = 1$, $v_j = f_{k_i} e_{l_i}$.

4) $l_i \notin \{1,\ldots,n\}$, $k_i \in \{1,\ldots,n\}$, $x_{k_i} \neq 0$: Analogous to 2).

5) $l_i \notin \{1,\ldots,n\}$, $k_i \in \{1,\ldots,n\}$, $x_{k_i} = 0$: Analogous to 3).

6) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} \geq b_{l_i-n}$, $f_{l_i} \neq 0$: $\alpha_j = f_{l_i}\alpha_{k_i}$, $v_j = v_{k_i} + \dfrac{f_{k_i}\alpha_{l_i}}{\alpha_j} v_{l_i}$.

7) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} \geq b_{l_i-n}$, $f_{l_i} = 0$: $\alpha_j = 1$, $v_j = f_{k_i}\alpha_{l_i} v_{l_i}$.

8) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} \neq 0$: Analogous to 6).

9) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} = 0$: Analogous to 7).

V) $\omega_i \in \mathcal{B}$, $\omega_i(a,b) = a/b$:

$$f_j = \frac{f_{k_i}}{f_{l_i}}, \quad g_j = \frac{1}{f_{l_i}}\left(g_{k_i} - f_j g_{l_i}\right) = \frac{\alpha_{k_i}}{f_{l_i}} v_{k_i} - \frac{f_j \alpha_{l_i}}{f_{l_i}} v_{l_i} = \alpha_j v_j$$

with

1) $k_i, l_i \in \{1,\ldots,n\}$: $\alpha_j = \dfrac{1}{x_{l_i}}$, $v_j = e_{k_i} - f_j e_{l_i}$.

2) $k_i \notin \{1,\ldots,n\}$, $l_i \in \{1,\ldots,n\}$: $\alpha_j = \dfrac{\alpha_{k_i}}{x_{l_i}}$, $v_j = v_{k_i} - \dfrac{f_j}{\alpha_{k_i}} e_{l_i}$.

3) $l_i \notin \{1,\ldots,n\}$, $k_i \in \{1,\ldots,n\}$, $x_{k_i} \neq 0$: $\alpha_j = -\dfrac{f_j \alpha_{l_i}}{f_{l_i}}$, $v_j = v_{l_i} - \dfrac{1}{\alpha_{l_i} f_j} e_{k_i}$.

4) $l_i \notin \{1,\ldots,n\}$, $k_i \in \{1,\ldots,n\}$, $x_{k_i} = 0$: $\alpha_j = 1$, $v_j = \dfrac{1}{f_{l_i}} e_{k_i}$.

5) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} \geq b_{l_i-n}$: $\alpha_j = \dfrac{\alpha_{k_i}}{f_{l_i}}$, $v_j = v_{k_i} - \dfrac{\alpha_{l_i} f_j}{\alpha_{k_i}} v_{l_i}$.

6) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} \neq 0$: $\alpha_j = -\dfrac{f_j \alpha_{l_i}}{f_{l_i}}$,

$$v_j = v_{l_i} - \frac{\alpha_{k_i}}{\alpha_{l_i} f_j} v_{k_i}.$$

7) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} = 0$: $\alpha_j = \dfrac{\alpha_{k_i}}{f_{l_i}}$, $v_j = v_{k_i}$.

Algorithm SF_H can be found in the Appendix. Table 4 contains the number of atomic operations that are required in each step of algorithm SF_g (SF_H). We turn to the analysis of time complexity:

Lemma 9.2 *If* $0 < y_1 \leq z_1$, $0 < y_2 \leq z_2$ *and* $0 \leq y \leq \min\{y_1, y_2\}$, *then*

$$z_1 \ln y_1 + z_2 \ln y_2 + y \leq (z_1 + z_2) \ln\left(y + \max\{y_1, y_2\}\right)$$

Proof: Without loss of generality, assume $y_1 \le y_2$. We have to show

$$\psi(y) := (z_1 + z_2) \ln(y + y_2) - z_1 \ln y_1 - z_2 \ln y_2 - y \ge 0 \qquad \forall\, y \in [0, y_1]\,.$$

This follows immediately from

$$\psi(0) = z_1 (\ln y_2 - \ln y_1) \ge 0$$

and

$$\psi'(y) = \frac{z_1 + z_2}{y + y_2} - 1 \ge \frac{z_1 + z_2}{y_1 + y_2} - 1 \ge 0 \qquad \forall\, y \in [0, y_1]\,.$$

□

Case	Atomic operations for			Rem.
	f_j	α_j, v_j	$\beta_j, \gamma_j, \delta_j, w_j, M_j$	
I)	0	0	0	
II.1)	1	1	1	
II.2)	1	3	7	
II.3)	1	2	3	
III.1)	1	0	0	
III.2)	1	$2 - q_i$	$\le 2b_s + 5$	
III.3)	1	$2 - q_i$	$\le 2b_t + 5$	
III.4)	1	$2b_t - q_i + 1$	$\le (3b_t + 2b_s + 7)b_t + 7$	$b_s \ge b_t$
III.5)	1	$2b_s - q_i + 1$	$\le (3b_s + 2b_t + 7)b_s + 7$	$b_s < b_t$
IV.1)	1	$2 - q_i$	0	
IV.2)	1	$4 - q_i$	$\le 2b_s + 11$	
IV.3)	1	1	1	
IV.4)	1	$4 - q_i$	$\le 2b_t + 11$	
IV.5)	1	1	1	
IV.6)	1	$2b_t - q_i + 4$	$\le (3b_t + 2b_s + 7)b_t + 15$	$b_s \ge b_t$
IV.7)	1	$b_t + 2$	$\le (2b_t + b_s + 5)b_t + 4$	$b_s \ge b_t$
IV.8)	1	$2b_s - q_i + 4$	$\le (3b_s + 2b_t + 7)b_s + 15$	$b_s < b_t$
IV.9)	1	$b_s + 2$	$\le (2b_s + b_t + 5)b_s + 4$	$b_s < b_t$
V.1)	1	$3 - q_i$	3	
V.2)	1	$3 - q_i$	$\le 2b_s + 12$	
V.3)	1	$5 - q_i$	$\le 2b_t + 16$	
V.4)	1	2	$\le b_t + 3$	
V.5)	1	$2b_t - q_i + 3$	$\le (3b_t + 2b_s + 7)b_t + 18$	$b_s \ge b_t$
V.6)	1	$2b_s - q_i + 4$	$\le (3b_s + 2b_t + 7)b_s + 18$	$b_s < b_t$
V.7)	1	2	$\le (2b_t + 2)b_s$	$b_s < b_t$

Table 4: Number of atomic operations in step i of algorithm SF_g (SF_H), $s = k_i - n$, $t = l_i - n$, $j = i + n$

Lemma 9.3 *Let S be a characterizing sequence for $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ of length m. If S is a T-sequence, then*

$$b_m \leq \#(f, S) + 1\,, \tag{8}$$

$$\sum_{i=1}^{m} p_i \leq (\#(f, S) + 1) \ln(\max\{b_m, 1\}) \tag{9}$$

$$\overline{\#}(f, \nabla f, S, SF_g) \leq 7\#(f, S) - 2b_m + \sum_{i=1}^{m} p_i\,, \tag{10}$$

$$\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) \leq 5(b_m + 1)\left(7\#(f, S) - 2b_m + \sum_{i=1}^{m} p_i\right) \tag{11}$$

with b_m, p_i according to (6), (7). $\overline{\#}(f, \nabla f, S, SF_g)$ denotes the number of atomic operations needed to calculate f_{m+n} and g_{m+n} in the factorized form $g_{m+n} = \alpha_{m+n} v_{m+n}$ with algorithm SF_g. $\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H)$ is defined analogously with factorization

$$H_{m+n} = \beta_{m+n} v_{m+n} v_{m+n}^T + \gamma_{m+n} w_{m+n} w_{m+n}^T + \delta_{m+n} M_{m+n}\,.$$

Proof: We use the notations (5), (6) and (7). We proof the assertions by induction with respect to the length m of T-sequences:
If $m = 1$, then the following cases may occur:

$\underline{\omega_1 \in \mathcal{C}:}$

$b_1 = p_1 = 0$, $\#(f, S) = \overline{\#}(f, \nabla f, S, SF_g) = \overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) = 0$.

$\underline{\omega_1 \in \mathcal{U},\ k_1 \in \{1, \ldots, n\}:}$

$b_1 = 1$, $p_1 = 0$, $\#(f, S) = 1$,
$\overline{\#}(f, \nabla f, S, SF_g) = 2$, $\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) = 3$.

$\underline{\omega_1 \in \mathcal{B},\ k_1, l_1 \in \{1, \ldots, n\}:}$

$b_1 = q_1 \leq 2$, $p_1 = 0$, $\#(f, S) = 1$,
$\overline{\#}(f, \nabla f, S, SF_g) \leq 4 - q_1$, $\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) = 7 - q_1$.

In all cases (8), (9), (10) and (11) are fulfilled.

Now assume that the assertions hold for all T-sequences of length $\leq m$ and let S be a T-sequence of length $m + 1$. We have to consider the following cases:

$\underline{\omega_{m+1} \in \mathcal{U}:}$

Then $k_{m+1} = m + n$ and S_m is a T-sequence of length m defining a function $\tilde{f}$. By induction, we have

$$b_m \leq \#(\tilde{f}, S_m) + 1\,, \tag{12}$$

$$\sum_{i=1}^{m} p_i \leq \Big(\#(\tilde{f}, S_m) + 1\Big) \ln\left(\max\{b_m, 1\}\right) , \tag{13}$$

$$\overline{\#}(\tilde{f}, \nabla \tilde{f}, S_m, SF_g) \leq 7\,\#(\tilde{f}, S_m) - 2b_m + \sum_{i=1}^{m} p_i \tag{14}$$

and

$$\overline{\#}(\tilde{f}, \nabla \tilde{f}, \nabla^2 \tilde{f}, S_m, SF_H) \leq 5(b_m + 1)\left(7\,\#(\tilde{f}, S_m) - 2b_m + \sum_{i=1}^{m} p_i\right) . \tag{15}$$

Moreover,

$$\#(f, S) = \#(\tilde{f}, S_m) + 1,\ \overline{\#}(f, \nabla f, S, SF_g) \leq \overline{\#}(\tilde{f}, \nabla \tilde{f}, S_m, SF_g) + 4,$$
$$\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) \leq \overline{\#}(\tilde{f}, \nabla \tilde{f}, \nabla^2 \tilde{f}, S_m, SF_H) + 11$$
$$b_{m+1} = b_m,\ p_{m+1} = 0,$$

and therefore

$$b_{m+1} = b_m \leq \#(f, S) ,$$
$$\sum_{i=1}^{m+1} p_i = \sum_{i=1}^{m} p_i \leq \#(f, S) \ln\left(\max\{b_{m+1}, 1\}\right) ,$$

$$\begin{aligned}\overline{\#}(f, \nabla f, S, SF_g) &\leq 7(\#(f, S) - 1) - 2b_{m+1} + \sum_{i=1}^{m+1} p_i + 4 \\ &\leq 7\#(f, S) - 2b_{m+1} + \sum_{i=1}^{m+1} p_i ,\end{aligned}$$

$$\begin{aligned}\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) &\leq 5(b_m + 1)\left(7\#(f, S) - 7 - 2b_{m+1} + \sum_{i=1}^{m+1} p_i\right) + 11 \\ &\leq 5(b_{m+1} + 1)\left(7\#(f, S) - 2b_{m+1} + \sum_{i=1}^{m+1} p_i\right) .\end{aligned}$$

$\underline{\omega_{m+1} \in \mathcal{B},\ \{k_i, l_i\} \cap \{1, \ldots, n\} \neq \emptyset}$:

Then $k_{m+1} = m + n$ or $l_{m+1} = m + n$ and S_m is a T-sequence of length m defining a function $\tilde{f}$. As above, (12), (13), (14) and (15) hold, and we have

$$\#(f, S) = \#(\tilde{f}, S_m) + 1,\ \overline{\#}(f, \nabla f, S, SF_g) \leq \overline{\#}(\tilde{f}, \nabla \tilde{f}, S_m, SF_g) + 6 - q_{m+1},$$
$$\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) \leq \overline{\#}(\tilde{f}, \nabla \tilde{f}, \nabla^2 \tilde{f}, S_m, SF_H) + 2b_m + 22 - q_{m+1}$$
$$b_{m+1} = b_m + q_{m+1},\ 0 \leq q_{m+1} \leq 1,\ p_{m+1} = 0 .$$

Hence,

$$b_{m+1} \leq b_m + 1 \leq \#(f, S) + 1 ,$$
$$\sum_{i=1}^{m+1} p_i = \sum_{i=1}^{m} p_i \leq \#(f, S) \ln\left(\max\{b_{m+1}, 1\}\right) ,$$

$$\begin{aligned}\overline{\#}(f,\nabla f,S,SF_g) &\leq 7(\#(f,S)-1)-2(b_{m+1}-q_{m+1})+\sum_{i=1}^{m+1}p_i+6-q_{m+1}\\ &\leq 7\#(f,S)-2b_{m+1}+\sum_{i=1}^{m+1}p_i\,.\end{aligned}$$

With

$$5(b_{m+1}+1)(6-q_{m+1})\geq 25(b_m+1)\geq 2b_m+22-q_{m+1}$$

we obtain

$$\overline{\#}(f,\nabla f,\nabla^2 f,S,SF_H)\leq 5(b_{m+1}+1)\left(7\#(f,S)-2b_{m+1}+\sum_{i=1}^{m+1}p_i\right)\,.$$

<u>$\omega_{m+1}\in\mathcal{B},\ k_i,l_i\in\{n+1,\dots,n+m\}$:</u>

Then $\tilde{S}:=S_{k_{m+1}-n}$ and $\hat{S}:=S_{l_{m+1}-n}$ are T-sequences of length $1\leq\tilde{m},\hat{m}\leq m$ for functions $\tilde{f}$ and $\hat{f}$. By induction, with $s=k_{m+1}-n$ and $t=l_{m+1}-n$ we have

$$\begin{aligned}b_s &\leq \#(\tilde{f},\tilde{S})+1\\ b_t &\leq \#(\hat{f},\hat{S})+1\\ \sum_{i\in N_s}p_i &\leq \Big(\#(\tilde{f},\tilde{S})+1\Big)\ln\left(\max\{1,b_s\}\right)\\ \sum_{i\in N_t}p_i &\leq \Big(\#(\hat{f},\hat{S})+1\Big)\ln\left(\max\{1,b_t\}\right)\\ \overline{\#}(\tilde{f},\nabla\tilde{f},\tilde{S},SF_g) &\leq 7\#(\tilde{f},\tilde{S})-2b_s+\sum_{i\in N_s}p_i\\ \overline{\#}(\hat{f},\nabla\hat{f},\hat{S},SF_g) &\leq 7\#(\hat{f},\hat{S})-2b_t+\sum_{i\in N_t}p_i\,.\\ \overline{\#}(\tilde{f},\nabla\tilde{f},\nabla^2\tilde{f},\tilde{S},SF_H) &\leq 5(b_s+1)\left(7\#(\tilde{f},\tilde{S})-2b_s+\sum_{i\in N_s}p_i\right)\\ \overline{\#}(\hat{f},\nabla\hat{f},\nabla^2\hat{f},\hat{S},SF_H) &\leq 5(b_t+1)\left(7\#(\hat{f},\hat{S})-2b_t+\sum_{i\in N_t}p_i\right)\,.\end{aligned}$$

In addition, $m=\tilde{m}+\hat{m}$, $\#(f,S)=\#(\tilde{f},\tilde{S})+\#(\hat{f},\hat{S})+1$,

$$\begin{aligned}\overline{\#}(f,\nabla f,S,SF_g) \leq\ & \overline{\#}(\tilde{f},\nabla\tilde{f},\tilde{S},SF_g)+\overline{\#}(\hat{f},\nabla\hat{f},\hat{S},SF_g)+5\\ &+2\min\{b_s,b_t\}-q_{m+1}\,,\end{aligned}$$

$$\begin{aligned}\overline{\#}(f,\nabla f,\nabla^2 f,S,SF_H) \leq\ & \overline{\#}(\tilde{f},\nabla\tilde{f},\nabla^2\tilde{f},\tilde{S},SF_H)+\overline{\#}(\hat{f},\nabla\hat{f},\nabla^2\hat{f},\hat{S},SF_H)\\ &+23+(5b_{m+1}+9)\min\{b_s,b_t\}-q_{m+1}\end{aligned}$$

and

$$b_{m+1}=\max\{b_s,\ b_t\}+q_{m+1},\ p_{m+1}=q_{m+1}=b_{m+1}-\max\{b_s,b_t\}\,.$$

Hence,

$$b_{m+1}=|B_{m+1}|=|B_s\cup B_t|\leq b_s+b_t\leq\#(f,S)+1\,.$$

Moreover,

$$p_{m+1} \le b_s + b_t - \max\{b_s, b_t\} = \min\{b_s, b_t\} \ .$$

Thus, setting

$$y := p_{m+1},\ y_1 := \max\{1, b_s\}\ ,\ y_2 := \max\{1, b_t\}\ ,$$

$$z_1 := \#(\tilde{f}, \tilde{S}) + 1,\ z_2 := \#(\hat{f}, \hat{S}) + 1\ ,$$

we may apply Lemma 9.2:

$$\begin{aligned}
\sum_{i=1}^{m+1} p_i &= \sum_{i\in N_s} p_i + \sum_{i\in N_t} p_i + p_{m+1} \le z_1 \ln y_1 + z_2 \ln y_2 + y \\
&\le (z_1 + z_2) \ln (y + \max\{y_1, y_2\}) \\
&= (\#(f, S) + 1) \ln (p_{m+1} + \max\{1, b_s, b_t\}) \\
&= (\#(f, S) + 1) \ln (\max\{1, b_{m+1}\}) \ .
\end{aligned}$$

In the last step we have used that $b_s = b_t = 0$ implies $b_{m+1} = 0$. Finally,

$$\begin{aligned}
&\overline{\#}(f, \nabla f, S, SF_g) \le \\
&\quad \le 7\#(\tilde{f}, \tilde{S}) - 2b_s + \sum_{i\in N_s} p_i + 7\#(\hat{f}, \hat{S}) - 2b_t + \sum_{i\in N_t} p_i + 5 \\
&\qquad +2\min\{b_s, b_t\} - q_{m+1} \\
&\quad = 7\#(f, S) - 7 + \sum_{i=1}^{m+1} p_i - 2q_{m+1} - 2\left(b_s + b_t - \min\{b_s, b_t\}\right) + 5 \\
&\quad = 7\#(f, S) - 2\left(\max\{b_s, b_t\} + q_{m+1}\right) + \sum_{i=1}^{m+1} p_i - 2 \\
&\quad < 7\#(f, S) - 2b_{m+1} + \sum_{i=1}^{m+1} p_i
\end{aligned}$$

The estimate

$$\begin{aligned}
&5(b_{m+1} + 1)(5 + 2\min\{b_s, b_t\} - q_{m+1}) \ge 5(b_{m+1} + 1)(5 + \min\{b_s, b_t\}) \\
&= 25b_{m+1} + 5b_{m+1}\min\{b_s, b_t\} + 5\min\{b_s, b_t\} + 25 \\
&\ge 23 + (5b_{m+1} + 9)\min\{b_s, b_t\} - q_{m+1}
\end{aligned}$$

yields

$$\begin{aligned}
\overline{\#}(f, \nabla f, \nabla^2 f, S, SF_H) &\le 5(b_{m+1} + 1)\Bigg(7\#(\tilde{f}, \tilde{S}) - 2b_s + \sum_{i\in N_s} p_i + 7\#(\hat{f}, \hat{S}) \\
&\qquad -2b_t + \sum_{i\in N_t} p_i + 5 + 2\min\{b_s, b_t\} - q_{m+1}\Bigg) \\
&\le 5(b_{m+1} + 1)\left(7\#(f, S) - 2b_{m+1} + \sum_{i=1}^{m+1} p_i\right)
\end{aligned}$$

□

By induction, one easily verifies:

Lemma 9.4 *Let S be a characterizing sequence for $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ of length m and B_i defined according to (6). Then, during the application of algorithm SF_g, all s^{th} components of the vector v_{i+n} with $s \notin B_i$ vanish. Moreover, during the application of algorithm SF_H, all s^{th} components, $s \notin B_i$, of the vectors v_{i+n} and w_{i+n} vanish and so do all (s,t)-components of M_{i+n} with $(s,t) \notin B_i \times B_i$.*

Now we are in a position to state the main theorem:

Theorem 9.5 *Let S be a characterizing sequence for $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ of length m. If S is a T-sequence, then*

$$\#(f, \nabla f, S, SF_g) \leq 7\#(f,S) - b + (\#(f,S)+1)\ln(\max\{b,1\})$$

and

$$\begin{aligned}\#(f, \nabla f, \nabla^2 f, S, SF_H) \leq\ & 5(b+1)\Big(7\#(f,S) - \frac{3}{2}b \\ & + (\#(f,S)+1)\ln(\max\{b,1\})\Big) + 3b\end{aligned}$$

Here $b = \left| \bigcup_{i=1}^{m} \{k_i, l_i\} \cap \{1,\ldots,n\} \right|$ denotes the number of x_k's on which f depends.

Proof: Due to Lemma 9.4, the computation of $\nabla f(x) = \alpha_{m+n} v_{m+n}$ costs at most $b = b_m$ multiplications. Thus,

$$\#(f, \nabla f, S, SF_g) \leq \overline{\#}(f, \nabla f, S, SF_g) + b\,.$$

Computing

$$\nabla^2 f(x) = \beta_{m+n} v_{m+n} v_{m+n}^T + \gamma_{m+n} w_{m+n} w_{m+n}^T + \delta_{m+n} M_{m+n}$$

requires $\leq 5\dfrac{b(b+1)}{2} + 2b$ operations. Now the assertions follow immediately by combining the results of Lemma 9.3.

□

Remark 9.6 In many cases one would like to compute the directonal derivative $\nabla^2 f(x)d$, $d \in \mathbb{R}^n \setminus \{0\}$, instead of the whole Hessian. The representations

$$g_j = \alpha_j v_j\ ,\ \alpha_j \neq 0\ ,\quad H_j d = \beta_j v_j + \gamma_j w_j + \delta_j z_j\ ,\ \delta_j \neq 0\ ,\ \ g_j^T d = \varepsilon_j\,.$$

allow efficient update formulas for the scalars $\alpha_j, \beta_j, \gamma_j, \delta_j, \varepsilon_j$ and the vectors v_j, w_j, z_j similar to those in algorithm SF_H.

□

10. Applications

In general, algorithm SF_g (and SF_H) develops its full efficiency only if it is applied to a T-sequence. We will put forward some arguments why it is nevertheless valuable. First of all, the results of Theorem 9.5 are interesting from a theoretical point of view, because they show that in special cases the forward mode is almost as fast as the reverse mode. There are classes of functions (e.g. Horner schemes) that are evaluable by T-sequences without additional costs and would require too much workspace when a reverse mode is applied. The algorithms SF_g, SF_H are directly applicable with almost optimal complexities $O(\ln(n)\#(f,S))$ and $O(\ln(n)n\#(f,S))$, which are in general better than the complexities $O((1+\ln(\#(f,S)))\#(f,S))$ and $O((1+\ln(\#(f,S)))n\#(f,S))$ of Griewank's recursive reverse mode. The numerical results in Section 13 even show typical costs of $\leq 4\#(f,S)$ and $\leq 2\,n\#(f,S)$, respectively. Further, the SF-algorithms are very efficient general purpose implementations of the forward mode and achieve the complexity of Theorem 9.5 in all subtrees of the Kantorovič graph.

Another application of the new algorithms consists in shortening characterizing sequences before applying the reverse mode. Every characterizing sequence for a function $f : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ can be split up as follows:

$$t_1 = \tau_1(x)\ ,\quad i = 2,\ldots,l:\ t_i = \tau_i(x,t_1,\ldots,t_{i-1})$$

where the functions

$$\tau_i : D_i \subset \mathbb{R}^{n+i-1} \longrightarrow \mathbb{R}$$

are given by T-sequences S_{τ_i}. In most cases, the function f is already formulated in this fashion. Analogously to Section 6, we define

$$\Phi_i : y \in D_i \longmapsto \begin{pmatrix} y \\ \tau_i(y) \end{pmatrix} \in \mathbb{R}^{n+i},\ i = 1,\ldots,l-1\ ,\quad \Phi_l : y \in D_l \longmapsto \tau_l(y) \in \mathbb{R}$$

and

$$T_0 := x\ ,\quad T_i := \begin{pmatrix} T_{i-1} \\ t_i \end{pmatrix} = \Phi_i \circ \cdots \circ \Phi_1(x)\ ,\quad i = 1,\ldots,l-1\,.$$

Then

$$\nabla f(x)^T = D\Phi_l(T_{l-1})\, D\Phi_{l-1}(T_{l-2}) \cdots D\Phi_1(x)\,.$$

Each Jacobian $D\Phi_i$ has the structure

$$D\Phi_i(y) = \begin{pmatrix} I_{n+i-1} \\ \nabla\tau_i(y)^T \end{pmatrix}.$$

Now we may apply a *reduced reverse mode* RR_g as follows:

1) forward calculation of the t_i, $i = 1,\ldots,l-1$:

   $t_1 := \tau_1(x)\ ,\quad i = 2,\ldots,l-1:\ t_i = \tau_i(x,t_1,\ldots,t_{i-1})$

   Here, the functions τ_i are evaluated using S_{τ_i}.

2) Compute $f(x) = t_l = \tau_l(T_{l-1})$ and $v^l := \nabla\tau_l(T_{l-1})$ by applying algorithm SF_g to S_{τ_l}.

3) Reverse steps:

   $i = l-1,\ldots,1$:

a) Compute $\nabla\tau_i(T_{i-1})$ by applying algorithm SF_g to S_{τ_i}.

b) Compute $v^i := v^{i+1}\begin{pmatrix} I_{n+i-1} \\ \nabla\tau_i(T_{i-1})^T \end{pmatrix}$

4) $\nabla f(x)^T = v^1$.

Note that the computation of v^i in 3 b) requires at most $2\,b_{\tau_i}$ operations, where b_{τ_i} denotes the number of non-zeroes in $\nabla\tau_i(T_{i-1})$. Since Lemma 9.3 and 9.4 imply $b_{\tau_i} \le \#(\tau_i, S_{\tau_i}) + 1$, we get with

$$\#(f, S) = \sum_{i=1}^{l} \#(\tau_i, S_{\tau_i})$$

in the case $\#(\tau_i, S_{\tau_i}) \ge 1$, $i = 1, \dots, l$, the estimate

$$\begin{aligned}
\#(f, \nabla f, S, RR_g) &\le \sum_{i=1}^{l-1} \#(\tau_i, S_{\tau_i}) + \sum_{i=1}^{l} \#(\tau_i, \nabla\tau_i, S_{\tau_i}, SF_g) + 2\sum_{i=1}^{l-1} b_{\tau_i} \\
&\le 3\sum_{i=1}^{l-1} \#(\tau_i, S_{\tau_i}) + \sum_{i=1}^{l} \#(\tau_i, \nabla\tau_i, S_{\tau_i}, SF_g) + 2(l-1) \\
&\le \left(3 + \max_{1\le i\le l} \frac{\#(\tau_i, \nabla\tau_i, S_{\tau_i}, SF_g)}{\#(\tau_i, S_{\tau_i})}\right) \#(f, S) + 2(l-1)\,.
\end{aligned}$$

Theorem 9.5 gives an upper bound for the maximum in the last expression. An algorithm for the computation of the k^{th} row of the Hessian can be obtained by considering the sequence

$$\begin{gathered}
t_1 = \tau_1(x)\ , \quad t_1^k = \tau_1^k(x) \\
i = 2, \dots, l: \quad t_i = \tau_i(x, t_1, \dots, t_{i-1})\ , \quad t_i^k = \tau_i^k(x, t_1, \dots, t_{i-1}, t_1^k, \dots, t_{i-1}^k)
\end{gathered}$$

with $\tau_1^k(x) = \partial_{x_k}\tau_1(x)$ and

$$\tau_i^k : (x, y, z) \in \mathbb{R}^n \times \mathbb{R}^{i-1} \times \mathbb{R}^{i-1} \longmapsto \nabla\tau_i(x, y)^T \begin{pmatrix} e_k \\ z \end{pmatrix}\ , \quad i = 2, \dots, l\,,$$

which simultanously computes $f(x) = t_l$ and $\partial_{x_k} f(x) = t_l^k$. To carry out a reverse mode, we need in each step $i+1 \to i$ the gradients $\nabla\tau_i(T_{i-1})$ and $\nabla\tau_i^k(T_{i-1}, T_{i-1}^k)$, where $T_i^k = \left(t_1^k, \dots, t_i^k\right)^T$. They can be computed by applying a modification of algorithm SF_H in the sense of Remark 9.6 to S_{τ_i} and by using the identity

$$\nabla\tau_i^k(x, y, z) = \begin{pmatrix} \nabla^2\tau_i(x, y)\begin{pmatrix} e_k \\ z \end{pmatrix} \\ \nabla_y\tau_i(x, y) \end{pmatrix}\,.$$

11. Technical Details

An efficient implementation of algorithm SF_g and SF_H requires a data structure for vectors that allows fast execution of the following operations:

- Set to zero.
- Determine the number of nonzero components.
- Check, whether a component is nonzero.
- Access all nonzero components.
- Create a new nonzero component.

By nonzero we mean the possibility of being nonzero, because the component was attached during the algorithm.

A vector $v \in \mathbb{R}^d$ is represented by

$$b^v, m^v \in \mathbb{N}\,,\ c^v \in \mathbb{N}^d,\ d^v \in \mathbb{N}^d,\ v^v \in \mathbb{R}^d\,.$$

b^v is the number of nonzeros, $\left(c_j^v\right)_{1\le j\le b^v}$ contains the indices of possibly nonzero components in arbitrary order and $d_i^v = m^v$ means that v_i is possibly nonzero, i.e. $i \in \left\{c_j^v\,;\ 1 \le j \le b^v\right\}$, with $v_i = v_i^v$.

A workspace of those vectors v is managed by a free-list. At the beginning, all vectors d^v and the b^v's are set to zero, the marks m^v are initialized to one. Then the above mentioned operations can be implemented as follows:

Set to zero: $m^v := m^v + 1$, $b^v := 0$

Number of nonzeros: b^v

Check whether component i is nonzero: $d_i^v \stackrel{?}{=} m^v$

Access to all nonzero components: $v_{c_j^v}^v$, $j = 1, \ldots, b^v$

Create a component $v_j = a$: $b_v := b_v + 1$, $c_{b^v}^v := j$, $v_j^v := a$, $d_j^v := m^v$

The matrix M can be represented in the same way. If the implementations of algorithm SF_g and SF_H are based on the above data structures and the assumptions of Theorem 9.5 are met, a dominating part of the runtime is spent to perform atomic operations. This is confirmed by the numerical results presented in Section 13.

12. Parallelization

The key to an efficient parallelization of the forward mode F_g is the observation that in each step $i-1 \to i$ the components of g_{i+n} can be computed concurrently if f_{k_i}, f_{l_i} and f_{i+n} are known. So, on a multiprocessor machine each processor can be entrusted with the evaluation of $f(x)$ and some partial derivatives $\partial_{x_k} f(x)$. Similarily, in algorithm F_H the components of H_{i+n} can be computed concurrently from f_{k_i}, f_{l_i}, f_{i+n} and g_{k_i}, g_{l_i}, g_{i+n}. Thus, each processor calculates $f(x)$, $\nabla f(x)$ and a part of the Hessian $\nabla^2 f(x)$. This method has been successfully implemented on a transputer workstation.
If one doesn't exploit the eventual concurrencies in the Kantorovič graph, for methods with better complexity only the parallelization of the Hessian seems to be promising. The reverse mode R_H consists of n characterizing sequences for

$f(x)$ and $\partial_{x_k} f(x)$, which can be interpreted and backward-differentiated fully in parallel. Algorithm SF_H can be parallelized in the same way as F_H.

13. Numerical Results

In the following examples $T(f)$ denotes the execution time to compute $f(x)$ on a HP 712/80 workstation. $T(f, \nabla f)$ and $T(f, \nabla f, \nabla^2 f)$ denote the execution times of algorithm SF_g and SF_H, respectively.

Example 13.1 $f : A \in \mathbb{R}^{d\times d} \longrightarrow \det A$, $d \geq 2$, evaluated by Laplace's expansion rule, i.e. $\#(f, S) \in [5n!/2 - 2, en! - 2[$.

$n = d^2$	Algorithm SF_g			Algorithm SF_H	
d	$T(f)$	$T(f,\nabla f)$	$\frac{T(f,\nabla f)}{T(f)}$	$T(f,\nabla f,\nabla^2 f)$	$\frac{T(f,\nabla f,\nabla^2 f)}{T(f)}$
6	2.67	8.41	3.15	26.7	$10.0 \approx 0.28n$
7	19.3	58.5	3.03	196	$10.2 \approx 0.21n$
8	150	472	3.15	1550	$10.3 \approx 0.16n$
9	1350	4250	3.15	14400	$10.7 \approx 0.13n$
10	13500	42900	3.18	143000	$10.6 \approx 0.11n$

Table 5: Execution times for Example 13.1 in 10^{-3} seconds

Example 13.2

$$f(x) = \sum_{k=1}^{K} \left(\frac{y_k - \left(1 + \sum_{j=1}^{n_1} u_j^k x_j\right) \left(\sum_{j=n_1+1}^{n_2} u_j^k x_j\right)}{\sum_{j=n_2+1}^{n} u_j^k x_j} \right)^2$$

with given data y_k, u_j^k.

K=500			Algorithm SF_g			Algorithm SF_H	
n_1	n_2	n	$T(f)$	$T(f,\nabla f)$	$\frac{T(f,\nabla f)}{T(f)}$	$T(f,\nabla f,\nabla^2 f)$	$\frac{T(f,\nabla f,\nabla^2 f)}{T(f)}$
5	7	9	0.016	0.046	2.88	0.19	$11.9 \approx 1.32n$
8	14	18	0.027	0.077	2.85	0.49	$18.1 \approx 1.01n$
16	28	36	0.050	0.14	2.80	1.50	$30.0 \approx 0.83n$
32	56	72	0.094	0.27	2.87	5.39	$57.3 \approx 0.80n$
64	112	144	0.18	0.53	2.94	21.5	$119 \approx 0.83n$
128	224	288	0.36	1.05	2.92	89.6	$249 \approx 0.86n$
256	448	576	0.72	2.09	2.90	354	$492 \approx 0.85n$
512	896	1152	1.37	4.22	3.08	1402	$1023 \approx 0.89n$
1024	1792	2304	2.66	8.73	3.28	–	–
4096	7168	9216	11.0	36.2	3.29	–	–
16384	28672	36864	43.2	144.1	3.34	–	–

Table 6: Execution times for Example 13.2 in seconds

Example 13.3 A worst case example can be obtained by evaluating a balanced Kantorovič tree of depth d with cyclic distribution of the n independent variables to the leaves and divisions as binary operations. The characterizing sequence is obtained by recursive descent. In the second part of the table, the constant increment of $T(f,\nabla f)$ by about 0.5 seconds when doubling n shows the logarithmic dependence on n.

		Algorithm SF_g			Algorithm SF_H	
d	n	$T(f)$	$T(f,\nabla f)$	$\frac{T(f,\nabla f)}{T(f)}$	$T(f,\nabla f,\nabla^2 f)$	$\frac{T(f,\nabla f,\nabla^2 f)}{T(f)}$
17	64	0.212	0.757	3.57	22.19	$105 \approx 1.64n$
17	128	0.212	0.814	3.84	45.4	$214 \approx 1.67n$
17	256	0.212	0.823	3.88	92.5	$436 \approx 1.70n$
17	512	0.212	0.890	4.20	183	$863 \approx 1.69n$
20	256	1.66	6.55	3.95	–	–
20	512	1.66	7.06	4.25	–	–
20	1024	1.66	7.56	4.55	–	–
20	2048	1.66	8.16	4.92	–	–
20	4096	1.66	8.67	5.22	–	–
20	8192	1.66	9.17	5.52	–	–

Table 7: Execution times for Example 13.3 in seconds

Acknowledgments

We wish to thank Dr. Herbert Fischer for his helpful suggestions and Michael Rempter for proof reading.

Appendix

Algorithm SF_H:

$j = 1, \ldots, n$:

$\alpha_j = 1,\ v_j = e_j,$

$\beta_j = \gamma_j = 0,\ \delta_j = 1,\ w_j = 0,\ M_j = 0.$

$i = 1, \ldots, m$:

$j := i + n$

<u>I) $\omega_i \in \mathcal{C}$:</u>

$$f_j = \omega_i,\quad g_j = 0 = \alpha_j v_j,\quad H_j = 0 = \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j$$

with

$\alpha_j = 1,\ v_j = 0,$

$\beta_j = \gamma_j = 0,\ \delta_j = 1,\ w_j = 0,\ M_j = 0.$

II) $\omega_i \in \mathcal{U}$:

$$\begin{aligned} f_j &= \omega_i(f_{k_i})\,, \quad g_j = \omega_i'(f_{k_i})g_{k_i} = \omega_i'(f_{k_i})\alpha_{k_i}v_{k_i} = \alpha_j v_j \\ H_j &= \omega_i'(f_{k_i})H_{k_i} + \omega_i''(f_{k_i})g_{k_i}g_{k_i}^T \\ &= (\omega_i''(f_{k_i})\alpha_{k_i}^2 + \omega_i'(f_{k_i})\beta_{k_i})v_{k_i}v_{k_i}^T + \omega_i'(f_{k_i})\gamma_{k_i}w_{k_i}w_{k_i}^T + \omega_i'(f_{k_i})\delta_{k_i}M_{k_i} \\ &= \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j \end{aligned}$$

with

1) $k_i \in \{1,\ldots,n\}$:

   $\alpha_j = 1,\ v_j = \omega_i'(x_{k_i})\,e_{k_i},$

   $\beta_j = 0,\ \delta_j = 1,\ \gamma_j = \omega_i''(x_{k_i}),\ w_j = e_{k_i},\ M_j = 0.$

2) $k_i \notin \{1,\ldots,n\},\ \omega_i'(f_{k_i}) \neq 0$:

   $\alpha_j = \omega_i'(f_{k_i})\alpha_{k_i},\ v_j = v_{k_i},$

   $\beta_j = (\omega_i''(f_{k_i})\alpha_{k_i}^2 + \omega_i'(f_{k_i})\beta_{k_i}),\ \gamma_j = \omega_i'(f_{k_i})\gamma_{k_i},\ \delta_j = \omega_i'(f_{k_i})\delta_{k_i},$

   $w_j = w_{k_i},\ M_j = M_{k_i}.$

3) $k_i \notin \{1,\ldots,n\},\ \omega_i'(f_{k_i}) = 0$:

   $\alpha_j = 1,\ v_j = 0,$

   $\beta_j = 0,\ \delta_j = 1,\ \gamma_j = \omega_i''(f_{k_i})\alpha_{k_i}^2,\ w_j = v_{k_i},\ M_j = 0.$

III) $\omega_i \in \mathcal{B},\ \omega_i(a,b) = a \pm b$:

$$\begin{aligned} f_j &= f_{k_i} \pm f_{l_i}\,, \quad g_j = g_{k_i} \pm g_{l_i} = \alpha_{k_i}v_{k_i} \pm \alpha_{l_i}v_{l_i} = \alpha_j v_j \\ H_j &= H_{k_i} \pm H_{l_i} = \beta_{k_i}v_{k_i}v_{k_i}^T + \gamma_{k_i}w_{k_i}w_{k_i}^T + \delta_{k_i}M_{k_i} \\ &\quad \pm \left(\beta_{l_i}v_{l_i}v_{l_i}^T + \gamma_{l_i}w_{l_i}w_{l_i}^T + \delta_{l_i}M_{l_i}\right) = \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j \end{aligned}$$

with

1) $k_i, l_i \in \{1,\ldots,n\}$:

   $\alpha_j = 1,\ v_j = e_{k_i} \pm e_{l_i},$

   $\beta_j = \gamma_j = 0,\ \delta_j = 1,\ w_j = 0,\ M_j = 0.$

2) $k_i \notin \{1,\ldots,n\},\ l_i \in \{1,\ldots,n\}$:

   $\alpha_j = \alpha_{k_i},\ v_j = v_{k_i} \pm \dfrac{1}{\alpha_{k_i}}e_{l_i},$

   $\beta_j = \beta_{k_i},\ \gamma_j = \gamma_{k_i},\ \delta_j = \delta_{k_i},\ w_j = w_{k_i},$

   $M_j = M_{k_i} - \dfrac{\beta_{k_i}}{\alpha_{k_i}\delta_{k_i}}\left(\dfrac{1}{\alpha_{k_i}}e_{l_i}e_{l_i}^T \pm \left(v_{k_i}e_{l_i}^T + e_{l_i}v_{k_i}^T\right)\right)$

3) $l_i \notin \{1,\ldots,n\},\ k_i \in \{1,\ldots,n\}$:

   Analogous to 2).

4) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} \geq b_{l_i - n}$:

$$\alpha_j = \alpha_{k_i},\ v_j = v_{k_i} \pm \frac{\alpha_{l_i}}{\alpha_{k_i}} v_{l_i},$$

$$\beta_j = \beta_{k_i},\ \gamma_j = \gamma_{k_i},\ \delta_j = \delta_{k_i},\ w_j = w_{k_i},$$

$$\begin{aligned} M_j = M_{k_i} \pm \frac{1}{\delta_{k_i}} \Bigg(\left(\beta_{l_i} \mp \beta_{k_i} \left(\frac{\alpha_{l_i}}{\alpha_{k_i}} \right)^2 \right) v_{l_i} v_{l_i}^T + \gamma_{l_i} w_{l_i} w_{l_i}^T + \delta_{l_i} M_{l_i} \\ - \beta_{k_i} \frac{\alpha_{l_i}}{\alpha_{k_i}} \left(v_{k_i} v_{l_i}^T + v_{l_i} v_{k_i}^T \right) \Bigg) \end{aligned}$$

5) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} < b_{l_i - n}$:

Analogous to 4).

IV) $\omega_i \in \mathcal{B}$, $\omega_i(a, b) = a \cdot b$:

$$\begin{aligned} f_j &= f_{k_i} f_{l_i}\,, \quad g_j = f_{l_i} g_{k_i} + f_{k_i} g_{l_i} = f_{l_i} \alpha_{k_i} v_{k_i} + f_{k_i} \alpha_{l_i} v_{l_i} = \alpha_j v_j \\ H_j &= f_{l_i} H_{k_i} + f_{k_i} H_{l_i} + g_{k_i} g_{l_i}^T + g_{l_i} g_{k_i}^T \\ &= f_{l_i} \beta_{k_i} v_{k_i} v_{k_i}^T + f_{l_i} \gamma_{k_i} w_{k_i} w_{k_i}^T + f_{l_i} \delta_{k_i} M_{k_i} + f_{k_i} \beta_{l_i} v_{l_i} v_{l_i}^T + f_{k_i} \gamma_{l_i} w_{l_i} w_{l_i}^T \\ &\quad + f_{k_i} \delta_{l_i} M_{l_i} + \alpha_{k_i} \alpha_{l_i} \left(v_{k_i} v_{l_i}^T + v_{l_i} v_{k_i}^T \right) = \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j \end{aligned}$$

with

1) $k_i, l_i \in \{1, \ldots, n\}$:

$$\alpha_j = 1,\ v_j = x_{l_i} e_{k_i} + x_{k_i} e_{l_i},$$

$$\beta_j = \gamma_j = 0,\ \delta_j = 1,\ w_j = 0,\ M_j = e_{k_i} e_{l_i}^T + e_{l_i} e_{k_i}^T.$$

2) $k_i \notin \{1, \ldots, n\}$, $l_i \in \{1, \ldots, n\}$, $x_{l_i} \neq 0$:

$$\alpha_j = x_{l_i} \alpha_{k_i},\ v_j = v_{k_i} + \frac{f_{k_i}}{\alpha_j} e_{l_i},$$

$$\beta_j = x_{l_i} \beta_{k_i},\ \gamma_j = x_{l_i} \gamma_{k_i},\ \delta_j = x_{l_i} \delta_{k_i},\ w_j = w_{k_i},$$

$$M_j = M_{k_i} + \frac{1}{\delta_j} \left(\left(\alpha_{k_i} - \beta_j \frac{f_{k_i}}{\alpha_j} \right) \left(v_{k_i} e_{l_i}^T + e_{l_i} v_{k_i}^T \right) - \beta_j \left(\frac{f_{k_i}}{\alpha_j} \right)^2 e_{l_i} e_{l_i}^T \right)$$

3) $k_i \notin \{1, \ldots, n\}$, $l_i \in \{1, \ldots, n\}$, $x_{l_i} = 0$:

$$\alpha_j = 1,\ v_j = f_{k_i} e_{l_i},$$

$$\beta_j = \gamma_j = 0,\ \delta_j = \alpha_{k_i},\ w_j = 0,\ M_j = v_{k_i} e_{l_i}^T + e_{l_i} v_{k_i}^T$$

4) $l_i \notin \{1, \ldots, n\}$, $k_i \in \{1, \ldots, n\}$, $x_{k_i} \neq 0$:

Analogous to 2).

5) $l_i \notin \{1, \ldots, n\}$, $k_i \in \{1, \ldots, n\}$, $x_{k_i} = 0$:

Analogous to 3).

6) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} \geq b_{l_i - n}$, $f_{l_i} \neq 0$:

$$\alpha_j = f_{l_i}\alpha_{k_i},\ v_j = v_{k_i} + \frac{f_{k_i}\alpha_{l_i}}{\alpha_j} v_{l_i},$$

$$\beta_j = f_{l_i}\beta_{k_i},\ \gamma_j = f_{l_i}\gamma_{k_i},\ \delta_j = f_{l_i}\delta_{k_i},\ w_j = w_{k_i},$$

$$M_j = M_{k_i} + \frac{1}{\delta_j}\left(\left(f_{k_i}\beta_{l_i} - \beta_j\left(\frac{f_{k_i}\alpha_{l_i}}{\alpha_j}\right)^2\right) v_{l_i}v_{l_i}^T + f_{k_i}\gamma_{l_i}w_{l_i}w_{l_i}^T \right.$$
$$\left. + f_{k_i}\delta_{l_i}M_{l_i} + \left(\alpha_{k_i}\alpha_{l_i} - \beta_j\frac{f_{k_i}\alpha_{l_i}}{\alpha_j}\right)\left(v_{k_i}v_{l_i}^T + v_{l_i}v_{k_i}^T\right)\right)$$

7) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} \geq b_{l_i-n}$, $f_{l_i} = 0$:

$$\alpha_j = 1,\ v_j = f_{k_i}\alpha_{l_i}v_{l_i},$$
$$\beta_j = 0,\ \gamma_j = f_{k_i}\gamma_{l_i},\ \delta_j = 1,\ w_j = w_{l_i},$$
$$M_j = f_{k_i}\beta_{l_i}v_{l_i}v_{l_i}^T + f_{k_i}\delta_{l_i}M_{l_i} + \alpha_{k_i}\alpha_{l_i}\left(v_{k_i}v_{l_i}^T + v_{l_i}v_{k_i}^T\right)$$

8) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} \neq 0$:

Analogous to 6).

9) $k_i, l_i \notin \{1,\ldots,n\}$, $b_{k_i-n} < b_{l_i-n}$, $f_{k_i} = 0$:

Analogous to 7).

V) $\omega_i \in \mathcal{B}$, $\omega_i(a,b) = a/b$:

$$f_j = \frac{f_{k_i}}{f_{l_i}},\quad g_j = \frac{1}{f_{l_i}}(g_{k_i} - f_j g_{l_i}) = \frac{\alpha_{k_i}}{f_{l_i}}v_{k_i} - \frac{f_j\alpha_{l_i}}{f_{l_i}}v_{l_i} = \alpha_j v_j$$
$$\begin{aligned} H_j &= \frac{1}{f_{l_i}}\left(H_{k_i} - f_j H_{l_i} - g_j g_{l_i}^T - g_{l_i}g_j^T\right) \\ &= \frac{\beta_{k_i}}{f_{l_i}}v_{k_i}v_{k_i}^T + \frac{\gamma_{k_i}}{f_{l_i}}w_{k_i}w_{k_i}^T + \frac{\delta_{k_i}}{f_{l_i}}M_{k_i} - \frac{f_j\beta_{l_i}}{f_{l_i}}v_{l_i}v_{l_i}^T - \frac{f_j\gamma_{l_i}}{f_{l_i}}w_{l_i}w_{l_i}^T \\ &\quad - \frac{f_j\delta_{l_i}}{f_{l_i}}M_{l_i} - \frac{\alpha_{l_i}\alpha_j}{f_{l_i}}\left(v_j v_{l_i}^T + v_{l_i}v_j^T\right) = \beta_j v_j v_j^T + \gamma_j w_j w_j^T + \delta_j M_j \end{aligned}$$

with

1) $k_i, l_i \in \{1,\ldots,n\}$:

$$\alpha_j = \frac{1}{x_{l_i}},\ v_j = e_{k_i} - f_j e_{l_i},$$
$$\beta_j = \gamma_j = 0,\ \delta_j = \alpha_j^2,\ w_j = 0,\ M_j = -e_{k_i}e_{l_i}^T - e_{l_i}e_{k_i}^T + 2f_j e_{l_i}e_{l_i}^T.$$

2) $k_i \notin \{1,\ldots,n\}$, $l_i \in \{1,\ldots,n\}$:

$$\alpha_j = \frac{\alpha_{k_i}}{x_{l_i}},\ v_j = v_{k_i} - \frac{f_j}{\alpha_{k_i}}e_{l_i},$$
$$\beta_j = \frac{\beta_{k_i}}{x_{l_i}},\ \gamma_j = \frac{\gamma_{k_i}}{x_{l_i}},\ \delta_j = \frac{\delta_{k_i}}{x_{l_i}},\ w_j = w_{k_i},$$

$$M_j = M_{k_i} + \frac{1}{\delta_j}\left(\left(\beta_j \frac{f_j}{\alpha_{k_i}} - \frac{\alpha_j}{x_{l_i}}\right)(v_{k_i} e_{l_i}^T + e_{l_i} v_{k_i}^T) - \frac{f_j}{\alpha_{k_i}}\left(\beta_j \frac{f_j}{\alpha_{k_i}} - 2\frac{\alpha_j}{x_{l_i}}\right) e_{l_i} e_{l_i}^T\right)$$

3) $l_i \notin \{1, \ldots, n\}$, $k_i \in \{1, \ldots, n\}$, $x_{k_i} \neq 0$:

$$\alpha_j = -\frac{f_j \alpha_{l_i}}{f_{l_i}},\ v_j = v_{l_i} - \frac{1}{\alpha_{l_i} f_j} e_{k_i},$$

$$\beta_j = -2\frac{\alpha_{l_i}\alpha_j}{f_{l_i}} - \frac{f_j \beta_{l_i}}{f_{l_i}},\ \gamma_j = -\frac{f_j \gamma_{l_i}}{f_{l_i}},\ \delta_j = -\frac{f_j \delta_{l_i}}{f_{l_i}},\ w_j = w_{l_i},$$

$$M_j = M_{l_i} + \frac{1}{\delta_j}\left(\frac{1}{\alpha_{l_i} f_j}\left(\beta_j + \frac{\alpha_{l_i}\alpha_j}{f_{l_i}}\right)(e_{k_i} v_{l_i}^T + v_{l_i} e_{k_i}^T) - \left(\frac{1}{\alpha_{l_i} f_j}\right)^2 \beta_j e_{k_i} e_{k_i}^T\right)$$

4) $l_i \notin \{1, \ldots, n\}$, $k_i \in \{1, \ldots, n\}$, $x_{k_i} = 0$:

$$\alpha_j = 1,\ v_j = \frac{1}{f_{l_i}} e_{k_i},$$

$$\beta_j = \gamma_j = 0,\ \delta_j = -\frac{\alpha_{l_i}}{f_{l_i}^2},\ w_j = 0,\ M_j = e_{k_i} v_{l_i}^T + v_{l_i} e_{k_i}^T$$

5) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} \geq b_{l_i - n}$:

$$\alpha_j = \frac{\alpha_{k_i}}{f_{l_i}},\ v_j = v_{k_i} - \frac{\alpha_{l_i} f_j}{\alpha_{k_i}} v_{l_i},$$

$$\beta_j = \frac{\beta_{k_i}}{f_{l_i}},\ \gamma_j = \frac{\gamma_{k_i}}{f_{l_i}},\ \delta_j = \frac{\delta_{k_i}}{f_{l_i}},\ w_j = w_{k_i},$$

$$M_j = M_{k_i} + \frac{1}{\delta_j}\left(\left(-\frac{f_j \beta_{l_i}}{f_{l_i}} - \frac{\alpha_{l_i} f_j}{\alpha_{k_i}}\left(\beta_j \frac{\alpha_{l_i} f_j}{\alpha_{k_i}} - 2\,\frac{\alpha_{l_i}\alpha_j}{f_{l_i}}\right)\right) v_{l_i} v_{l_i}^T - \frac{f_j \gamma_{l_i}}{f_{l_i}} w_{l_i} w_{l_i}^T - \frac{f_j \delta_{l_i}}{f_{l_i}} M_{l_i} + \left(\beta_j \frac{\alpha_{l_i} f_j}{\alpha_{k_i}} - \frac{\alpha_{l_i}\alpha_j}{f_{l_i}}\right)(v_{k_i} v_{l_i}^T + v_{l_i} v_{k_i}^T)\right)$$

6) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} < b_{l_i - n}$, $f_{k_i} \neq 0$:

$$\alpha_j = -\frac{f_j \alpha_{l_i}}{f_{l_i}},\ v_j = v_{l_i} - \frac{\alpha_{k_i}}{\alpha_{l_i} f_j} v_{k_i},$$

$$\beta_j = -\frac{f_j \beta_{l_i}}{f_{l_i}} - 2\frac{\alpha_{l_i}\alpha_j}{f_{l_i}},\ \gamma_j = -\frac{f_j \gamma_{l_i}}{f_{l_i}},\ \delta_j = -\frac{f_j \delta_{l_i}}{f_{l_i}},\ w_j = w_{l_i},$$

$$M_j = M_{l_i} + \frac{1}{\delta_j}\left(\left(\frac{\beta_{k_i}}{f_{l_i}} - \beta_j \left(\frac{\alpha_{k_i}}{\alpha_{l_i} f_j}\right)^2\right) v_{k_i} v_{k_i}^T + \frac{\gamma_{k_i}}{f_{l_i}} w_{k_i} w_{k_i}^T + \frac{\delta_{k_i}}{f_{l_i}} M_{k_i} + \frac{\alpha_{k_i}}{\alpha_{l_i} f_j}\left(\beta_j + \frac{\alpha_{l_i}\alpha_j}{f_{l_i}}\right)(v_{k_i} v_{l_i}^T + v_{l_i} v_{k_i}^T)\right)$$

7) $k_i, l_i \notin \{1, \ldots, n\}$, $b_{k_i - n} < b_{l_i - n}$, $f_{k_i} = 0$:

$$\alpha_j = \frac{\alpha_{k_i}}{f_{l_i}},\ v_j = v_{k_i},$$

$$\beta_j = \frac{\beta_{k_i}}{f_{l_i}},\ \gamma_j = \frac{\gamma_{k_i}}{f_{l_i}},\ \delta_j = \frac{\delta_{k_i}}{f_{l_i}},\ w_j = w_{k_i},$$
$$M_j = M_{k_i} - \frac{\alpha_{l_i}\alpha_j}{\delta_j f_{l_i}}\left(v_{k_i}v_{l_i}^T + v_{l_i}v_{k_i}^T\right)$$

References

[1] W. Baur, V. Strassen: *The Complexity of Partial Derivatives.* Theor. Comp. Sci. 22, 1983, 317-330.

[2] H. Fischer: *Automatisches Differenzieren.* In: Wissenschaftliches Rechnen (J. Herzberger ed.), Akademie Verlag, Berlin, 1995.

[3] H. Fischer, H. Warsitz: *Complexity Investigations Concerning the Derivative of a Rational Function.* Technical Report 138, Inst. f. Angew. Math. u. Stat., TU München, 1995.

[4] A. Griewank: *Achieving Logarithmic Growth of Temporal and Spacial Complexity in Reverse Automatic Differentiation.* Optimiz. Meth. Softw. 1, 1992, 35-54.

[5] A. Griewank, G. F. Corliss (eds.): *Automatic Differentiation of Algorithms: Theory, Implementations and Applications.* SIAM, Philadelphia, 1991.

[6] M. Iri: *History of Automatic Differentiation and Rounding Error Estimation.* In: [5].

[7] L. B. Rall: *Automatic Differentiation: Techniques and Applications.* Lect. Notes Comp. Sci. 120, Springer-Verlag, Berlin, 1981.