ALGORITHM TRANSFORMATION

L. B. Rall

An algorithm

$$t_i = f_i(t_1, \ldots, t_{i-1}), \quad i = 2, \ldots, n,$$

defines its *output* $t_n = f(t_1)$ as a function of its *input* t_1. This can be transformed into an algorithm for $T_n = F(T_1)$ by providing appropriate functions F_i corresponding to f_i (the *direct* or *forward method*), or in other ways.

In the case of numerical routines (i.e., computer programs), the functions f_i are arithmetic operations or given standard functions (sometimes called library or intrinsic functions).

EXAMPLES

- Single precision $\longrightarrow$ double precision
- Real $\longrightarrow$ complex
- Real $\longrightarrow$ interval
- Real vector $\longrightarrow$ gradient
- Real $\longrightarrow$ Taylor
- Real $\longrightarrow$ Fourier

Each of these algorithm transformations can be performed in forward mode by providing the appropriate arithmetic operations and standard functions.

REAL $\longrightarrow$ INTERVAL

The forward transformation gives the *united extension* $T_n = F(T_1)$ of $f(t_1)$ for $t_1 \in T_1$ (Moore, 1979). One has

$$T = \{f(t_1) | t_1 \in T_1\} \subseteq T_n,$$

but the result may not be a useful inclusion of T.

The use of *centered*, *mean value*, or more generally, *Taylor* forms may reduce the excess width of T, see (Moore, 1979), also (Rall, 1983).

REAL VECTOR $\longrightarrow$ GRADIENT

This is called automatic (or algorithmic) differentiation. In addition to forward mode, the original algorithm can be transformed into an efficient *reverse mode* algorithm (Berz, 1996). A simple explanation will appear in Reliable Computing: L. B. Rall, Computation of functions, gradients, and Jacobians. The transformed algorithm here will generally have a different number of steps than the original.

In either mode, derivatives have to be supplied for arithmetic operations and standard functions.

REAL $\longrightarrow$ TAYLOR

This is another version of automatic differentiation. Moore (1979) gives arithmetic operations and standard functions to transform the algorithm for $t_n = f(t_1)$ into an algorithm for the vector T_n of Taylor coefficients of $f(t_1)$, given the vector T_1 of Taylor coefficients of t_1.

The same idea applies to Fourier series as well as other expansions. The case of Fourier series is the subject of current work by Rall.

TRUNCATION (ROUNDING) ERROR

Rounding a real number r to a finite-precision (f.p.) number r_m is equivalent to truncation of the series expansion

$$r = b^e \times \sum_{n=1}^{\infty} d_n b^{-n}$$

to

$$r_m = b^e \times \sum_{n=1}^{m} d_n b^{-n}.$$

Similarly, rounding a Taylor series expansion to a Taylor polynomial of degree m has error

$$R_m = \frac{h^{m+1}}{(m+1)!} f^{(m+1)}(\xi),$$

where ξ lies between the expansion point x and $x + h$. This error can be bounded by an interval inclusion of $f^{(m+1)}(\xi)$. For h small, the united extension may be good enough, or the width of the error term may be reduced when monotonicity obtains (Corliss & Rall, 1999).

REFERENCES

1. M. Berz, C. Bischof, G. F. Corliss, A. Griewank (Eds.) Computational Differentiation, Techniques, Applications, and Tools, SIAM, 1996.
2. G. F. Corliss and L. B. Rall, Bounding derivative ranges, in Encyclopedia of Optimization, Ed. by P. M. Pardalos and C. A. Floudas, Kluwer, 2001.
3. R. E. Moore, Methods and Applications of Interval Analysis, SIAM, 1979.
4. L. B. Rall, Mean value and Taylor forms in interval analysis, SIAM J. Math. Analysis **14** (1983), 223–238.