# Differentiating Fixed Point Iterations with ADOL-C

S. Schlenkrich[1] and A. Walther[1]

**Abstract:** Automatic Differentiation (AD) based on a sequential internal function representation requires an amount of memory proportional to the size of the computational graph. For iterative processes of uniform complexity this means that the memory requirement is proportional to the number of iterations. Especially for fixed point iterations of the form

$$x_{k+1} = F(x_k, u) \text{ with } x_k, x_{k+1} \in \mathbb{R}^m \text{ and } u \in \mathbb{R}^n$$

this is not efficient, since it neglects any structure of the problem.

The method of Reverse Accumulation, introduced by B. Christianson (1994) allows for linear converging iterations $\lim_{k\to\infty} x_k = x_*$ the iterative computation of the gradient

$$\bar{u}^T = \bar{x}^T \frac{dx_*}{du}.$$

Here $\bar{x} \in \mathbb{R}^m$ describes a given weight vector and $\bar{u} \in \mathbb{R}^n$ is the vector of the resulting sensitivities or the gradient, respectively. The iteration of the gradient converges with the same rate as the fixed point iteration itself. The memory requirement for this method is independent of the number of iterations and with this of the desired accuracy.

We apply the concept of Reverse Accumulation within the AD tool ADOL-C to compute gradients of fixed point iterations. Results for the large scale applications will be presented. Furthermore we give the concept to incorporate differentiation of fixed point iterations into ADOL-C to decrease the memory requirement and therefore increase the range of applications.

[1] Institute for Scientific Computing, Technische Universität Dresden
Zellescher Weg 12–14, 01069 Dresden, Germany
*[schlenk,awalther]@math.tu-dresden.de*