Differentiation of Programs by Preprocessing or Overloading

Prof. Andreas Griewank

Technische Universitaet Dresden
Institut fuer Wissenschaftliches Rechnen

Most numerical methods for the solution of nonlinear problems require the calculation or approximation of derivatives for mathematical functions defined by evaluation (sub)programs. The chain rule based technique of automatic differentiation allows the extraction of such quantitative and qualitative sensitivity information without extra coding effort by the user and thus with much greater reliability. Moreover the computational complexities can be a priori bounded and are surprisingly low, especially with regards to the evaluation of gradients for functions in very many variables.

The extra instructions needed for the evaluation of derivatives can be generated by a preprocessor that yields and extended source code or build into new variable classes that replace the standard types real or double. The former approach has mostly been applied to Fortran sources and achieves usually better run-times than the second, which requires the overloading facilities of C++, ADA, or now Fortran 90. We will discuss the merits and drawbacks of various implementations and come to the conclusion that sensitivity analysis in a wider sense should be provided as a compiler option by programming environments for serious scientific computing.