PROJECT PART-FINANCED BY THE EUROPEAN UNION
North East South West INTERREG IIIC
MATEO
MATEO ANTASME

Automatic Differentiation Tools

Purpose of this document is to give an overview of available automatic differentiation tools and libraries.

An automatic differentiation program should be able, with minimum human intervention, to parse the source code of a program a generate a subroutine that allows the computation of the gradient, or of the Jacobian matrix, of a given function. Different tools are available free of charge. Most of them, however, can deal only with Fortan77 and/or Fortran95 language, thanks to the relative simplicity of such languages.

An automatic differentiation library uses operator overloading in order to record each and every floating point operations performed during the evaluation of a function. The data structure used to record the sequence of floating point operations if often referred to as a “tape”. When a derivative of the result with respect to some variable is required, the tape can be walked, in forward or reverse mode[1], computing the required derivative.

A (partial) list of the available tools is given in Table 1. The list of tools is taken from www.autodiff.org. Soem of them are commercial, and some of them require a one-time fee to be dowloaded.

Tool	Language	License	Source code	Method
ADF95	Fortran77/Fortran95	Custom non.profit	Yes $	Code analysis?
ADIC	C/C++	?	No	
ADIFOR	Fortran77	Custom non-profit	Yes	Code analysis?
AdiMat	MATLAB	To be defined	No	
ADOL-C	C/C++	CPL	Yes	Tape
AUTO_DERIV	Fortran77/Fortran95	Custom non.profit	Yes $	Code analysis?
CppAD	C/C++	CPL/GPL	Yes	Tape
FAD	C/C++	Teaching, non profit	Yes	Tape?
FADBAD/TADIFF	C/C++	Custom non.profit	Yes	Tape
GRESS	Fortran77	?	Yes	?
OpenAD	Fortran77/Fortran95	?	No? (EDG front end)	Code analysis
TAMC	Fortran77	Non-profit	No	Code analysis?
TAPENADE	Fortran77/Fortran95	?	No	Code analysis?
TaylUR	Fortran95	Custom non.profit	Yes $	Code analysis?
TOMLAB/MAD	MATLAB	?	$$	Tape

Table 1: Automatic differentiation tools

[1] “The forward mode computes derivatives of intermediate variables with respect to independent variables and propagates from one statement to the next according to the chain rule. The reverse mode computes derivatives of dependent variables with respect to intermediate variables and propagates from one statement to the previous statement according according to the chain rule. Thus, the reverse mode requires a reversal of the program execution” [1].

Looking at Table 1 it is clear that most of the available tools are not suited for the need of the "automatic differentiation techniques" WP of ANTASME. We need a tool that system)

- is freely downloadable and usable (MBDyn is GPL)
- comes with source code (we should not be limited to a particular enviromnet and/or operating system)
- can deal with complex C++ code
- requires minimum changes to the code

The library CppAD [3] satisfies all thees requirements. It is based on operator overloading, and requires only the change of the data type from double to CppAD<double>. This should be a non-intrusive requirement, because the MBDyn code do not use directly the "double" data type, but defines a custom "doublereal" type that can be changed for the whole code.
The library also requires a function call at the beginning and at the end of a tape session. After that, it is possible to compute the Jacobian matrix of the registered tape. Forward or reverse mode is automatically chosen in order to reduce the number of floating point operations.

Bibliography

1) http://www.autodiff.org/
2) http://www.coin-or.org/CppAD/
3) Andreas Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation SIAM, 2000.