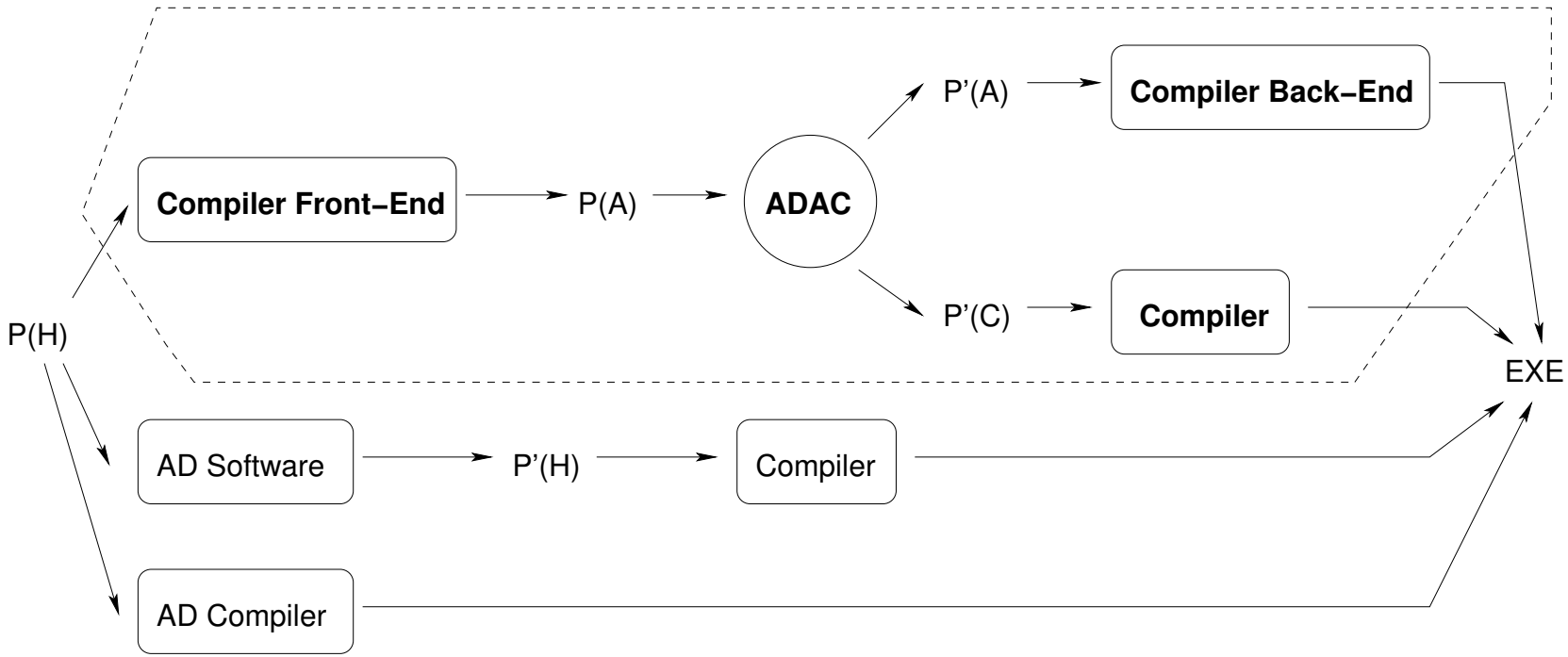
Automatic Differentiation of Assembler Code

Uwe Naumann and Dmitrij Gendler

Department of Computer Science
RWTH Aachen University, Aachen, Germany
{naumann,gendler}@stce.rwth-aachen.de

Contents

Context


P(H)
Compiler Front–End
P(A)
ADAC
P'(A)
Compiler Back–End
P'(C)
Compiler
EXE
AD Software
P'(H)
Compiler
AD Compiler

Functionality

For a given assembler routine that implements a vector function

$$F : \boldsymbol{R}^{n_a+n_p} \longrightarrow \boldsymbol{R}^{m_a+m_p}$$

as

$$\begin{pmatrix} \mathbf{y}_a \\ \mathbf{y}_p \end{pmatrix} = F(\mathbf{x}_a, \mathbf{x}_p) \quad ,$$

where $\mathbf{x}_a \in \boldsymbol{R}^{n_a}$ ($\mathbf{x}_p \in \boldsymbol{R}^{n_p}$) are the active (passive) inputs and $\mathbf{y}_a \in \boldsymbol{R}^{m_a}$ ($\mathbf{y}_p \in \boldsymbol{R}^{m_p}$) are the active (passive) outputs, the ADAC compiler generates a new assembler routine that implements the tangent-linear function $\dot{F} : \boldsymbol{R}^{2\cdot n_a+n_p} \longrightarrow \boldsymbol{R}^{2\cdot m_a+m_p}$ as

$$\begin{pmatrix} \mathbf{y}_a \\ \dot{\mathbf{y}}_a \\ \mathbf{y}_p \end{pmatrix} = \dot{F}(\mathbf{x}_a, \dot{\mathbf{x}}_a, \mathbf{x}_p) \quad .$$

Availability and Applicability

ADAC is open-source software. It is available for downloading from our Internet site

`http://www.stce.rwth-aachen.de/ADAC/` .

The current version of ADAC has been created and tested successfully under Linux in the following configuration:

- Fedora Core release 2 (Tettnang)
- gcc (GCC) 3.3.3 20040412 (Red Hat Linux 3.3.3-7) .

Test Problems

We provide test sets for the following five problems from the MINPACK-2 test problem collection

- Chebyshev quadrature
- coating thickness standardization
- enzyme reaction
- flow in a driven cavity
- solid fuel ignition

Two scripts are provided to build ADAC and to run a test with the corresponding input problem. Refer to the website for more specific instructions.

ADAC and CompAD-II

- second-order adjoints by forward over reverse mode
- tangent-linear version of adjoint assembler code by ADAC
- alternatively tangent-linear back-end C code (possibly single-pass?)