June 21, 2006 **Summary**

Nuclear Science and Technology Division (94)

Automatic Differentiation to Couple SCALE Modules Using GRESS 90—Part I: Methodology

J. E. Horwedel

Oak Ridge National Laboratory*
P.O. Box 2008, MS-6170
Oak Ridge, TN 37831-6170 USA
Telephone: (865) 576-4420
Fax: (865) 576-3513
E-mail: horwedelje@ornl.gov

Submitted to the
American Nuclear Society
2006 Winter Meeting, "Ensuring the Future in Times of Change: Nonproliferation and Security"
November 12–16, 2006
Albuquerque, NM

* Managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy.

Automatic Differentiation to Couple SCALE Modules Using GRESS 90—Part I: Methodology

James E. Horwedel

Oak Ridge National Laboratory,[*] P.O. Box 2008, Bldg. 5700, Oak Ridge, TN 37831-6170, horwedelje@ornl.gov

INTRODUCTION

Sensitivity analysis is a necessary component in the evaluation of computer simulation models. Because computational simulation often involves two or more computer programs run in sequence, the calculation of sensitivities using analytic methods requires a code coupling procedure to propagate sensitivities through the entire code sequence. This summary describes a code coupling methodology used in conjunction with automatic differentiation to develop new versions of existing Fortran 90 programs by adding the capability to calculate sensitivities analytically even when the computational simulation is run as a sequence of computer programs. The code coupling method was successfully used to couple SCALE [1] modules CENTRM [2] and PMC [3]. The application of this code coupling methodology to CENTRM and PMC is described in a companion paper.[4]

BACKGROUND INFORMATION

Because programmed equations can be differentiated analytically, sensitivities can be precisely defined and calculated using automatic differentiation. [5-9] Furthermore, when computational simulations are implemented as a sequence of computer codes, the automatic differentiation approach can be extended to automate the calculation of sensitivities for the entire sequence.

In the 1980s, the Gradient Enhanced Software System (GRESS) [8,9] was developed at Oak Ridge National Laboratory to automate the implementation of sensitivity analysis methods into existing Fortran 77 programs. More recently GRESS was upgraded to allow processing of Fortran 90 programs. The new version of GRESS is named GRESS 90.

An automated code coupling methodology implemented in GRESS 90 extends the automatic differentiation approach to couple a sequence of computer programs. The code coupling procedure involves writing derivatives calculated in one code to a transfer file, along with information identifying parameters of interest for sensitivity calculations. The next code in the sequence reads the transfer file and initializes derivative information needed to calculate derivatives and sensitivities with respect to parameters identified in the first code.

A MATHEMATICAL MODEL OF A COMPUTER PROGRAM

In a Fortran program, calculated left-hand-side variables are a function of previously defined left-hand-side variables and data, either through mathematical operations or read statements. This relationship can be expressed as

$$\bar{y} <= \bar{f}(\bar{y}), \tag{1}$$

where the symbol <= indicates a value assignment (i.e., store) operation. The components of the column vector, $\bar{y}$, are all the terms on the left-hand-side of real number replacement statements. The column vector, $\bar{f}$, represents the right-hand-side mathematical operations. The vector, $\bar{y}$, includes both calculated results and input data. Read statements are treated in the same manner as setting a variable equal to a constant.

In a Fortran program, a symbol cannot explicitly depend on itself. When a Fortran variable is redefined, mathematically, it is not the same variable. In the statement, $X <= X + 5.0$, the X on the left and the X on the right represent two different locations in the solution vector, $\bar{y}$. Mathematically, the equation can be thought of as $X_2 = X_1 + 5.0$. Therefore to represent Eq. (1) mathematically, the dependence of a variable on itself must be considered explicitly. If we define

$$\frac{dy_i}{dy_i} = 1, \text{ for all i,} \tag{2}$$

then Eq. (1) can be rewritten as

$$\bar{y} = \bar{f}(\bar{y}). \tag{3}$$

Differentiating Eq. (3) with respect to $\bar{y}$ yields

$$\frac{d\bar{y}}{d\bar{y}} = \frac{\partial \bar{f}}{\partial \bar{y}} \cdot \frac{d\bar{y}}{d\bar{y}} + I, \tag{4}$$

[*] Managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.

where the identity matrix, *I*, provides the explicit dependence of a variable on itself necessary to make Eq. (4) meaningful. Eq. (4) can be rearranged, such that

$$\left[I - \frac{\partial \overline{f}}{\partial \overline{y}} \right] \cdot \frac{d\overline{y}}{d\overline{y}} = I \,. \tag{5}$$

Eq. (5) can be represented in a more compact form as

$$AY' = I \,, \tag{6}$$

where

$$A = \left[I - \frac{\partial \overline{f}}{\partial \overline{y}} \right]$$

and

$$Y' = \frac{d\overline{y}}{d\overline{y}} \quad .$$

Because Fortran equations are solved in a sequential fashion, Fortran variables are dependent on previously defined variables. Therefore,

$$\frac{\partial f_i}{\partial y_j} = 0 \text{ , for } \mathrm{j} \geq \mathrm{i} \,,$$

such that the matrix, $\frac{\partial \overline{f}}{\partial \overline{y}}$, is a lower triangular matrix with zeros on and above the diagonal. This matrix can easily be solved by application of the chain rule of calculus as each row is determined.

GRESS 90 FORWARD CHAINING OPTION

The GRESS 90 forward chaining option is utilized to calculate and report sensitivities with respect to a subset of the input data. The method used is to resolve Eq. (6) for selected columns in the matrix, Y', by forward substitution in memory as each Fortran statement is executed. A fully resolved column in Y' represents the derivatives of every real variable with respect to the user-selected variable or parameter associated with that column. Because selected columns in Y' are resolved by forward substitution in memory, the *A* matrix is never saved. At any given point during execution, the user can retrieve the total first-order derivatives of a calculated variable with respect to all the declared parameters. The steps used to process a code with GRESS 90 are illustrated in Fig. 1.

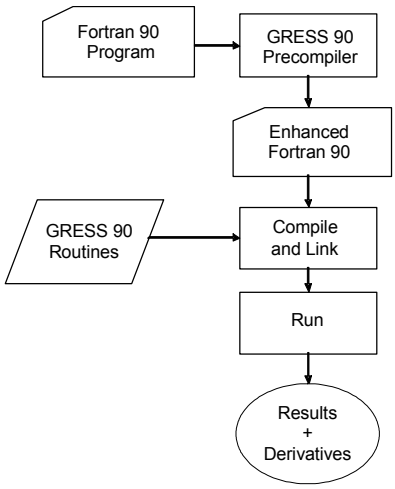

Fig. 1. Flowchart showing the processing steps for using GRESS 90.

GRESS 90 CODE COUPLING

When a model is implemented as a sequence of computer codes, the code coupling method, illustrated by the flowchart in Fig. 2, is used to calculate derivatives with respect to input parameters. For example, assume there are three codes, named A, B, and C, respectively. As shown in Fig. 2, both A and B are run first. The input to codes A and B includes sets of parameters of interest, P_A and P_B, respectively. The results from A and B (i.e., R_A and R_B, respectively) are read as input by code C. The results of interest for derivative calculation, R_C, are output from code C. The independent variables of interest are the inputs P_A and P_B.

The derivatives of results from A and B with respect to input parameters can be represented as

$$\frac{d\overline{R_A}}{d\overline{P_A}}$$

and

$$\frac{d\overline{R_B}}{d\overline{P_B}} \,.$$

Via application of the chain rule, the derivatives of code C with respect to input parameters can be represented as

$$\frac{d\overline{R_C}}{d\overline{P}} = \frac{\partial \overline{R_C}}{\partial \overline{R_A}} \cdot \frac{d\overline{R_A}}{d\overline{P}} + \frac{\partial \overline{R_C}}{\partial \overline{R_B}} \cdot \frac{d\overline{R_B}}{d\overline{P}} + \frac{\partial \overline{R_C}}{\partial \overline{P}} \,,$$

where $\overline{P} = \overline{P_A} \cup \overline{P_B}$.

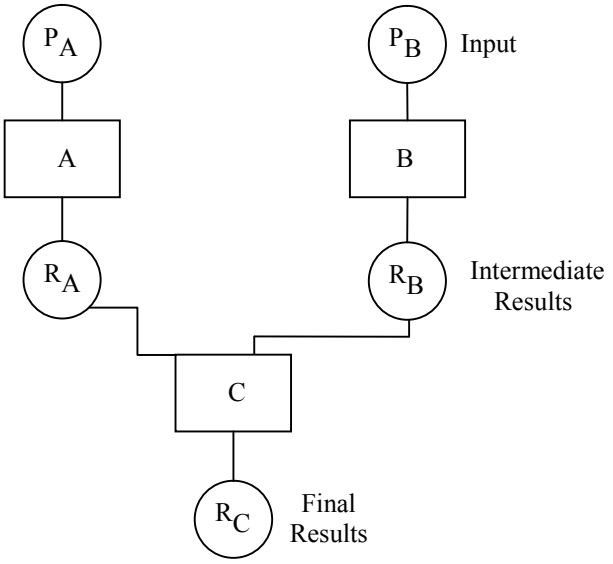

Fig. 2. Flowchart illustrating the GRESS 90 code coupling methodology.

The results that are calculated in codes A and B and then read by C are referred to as transfer variables. The parameters of interest from codes A and B are referred to as transfer parameters in C. The user inserts subroutine calls in code C to identify transfer parameters and transfer variables. When code C is executed, derivatives of floating point variables with respect to parameters defined in codes A and B are calculated and may be reported or used for sensitivity calculations.

CONCLUSION

The automated GRESS 90 procedure with code coupling was successfully used to add sensitivity capability to SCALE programs CENTRM and PMC.[4] Results demonstrate that the GRESS 90 code coupling methodology can be successfully applied to existing Fortran 90 programs to create new versions of those programs enhanced for sensitivity calculations. Because computational simulations often involve two or more programs run in sequence, the automated code coupling methodology using GRESS 90 is a significant new capability for calculating sensitivities in such simulations.

REFERENCES

1. *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5, Vols. I-III, April 2005. Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-725.
2. M. L. WILLIAMS, M. ASGARI, D. F. HOLLENBACH, "CENTRM: A One-Dimensional Neutron Transport Code for Computing Pointwise Energy Spectra," in *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5, Vol. II, Sect. F19 (April 2005). Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-725.
3. M. L. WILLIAMS, D. F. HOLLENBACH, "PMC: A Program to Produce Multigroup Cross Sections Using Pointwise Energy Spectra from CENTRM," in *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5, Vol. II, Sect. F19 (April 2005). Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-725.
4. B. T. REARDEN, "Application of Automatic Differentiation to Coupled SCALE Modules Using GRESS 90—Part 2," submitted to the American Nuclear Society 2006 Winter Meeting, November 12–16, 2006, Albuquerque, N.M.
5. C. BISCHOF, A. CARLE, A. MAUER, "Adifor 2.0: Automatic Differentiation of Fortran 77 Programs," *IEEE Computational Science & Engineering*, **3**(3), 18, (1996).
6. K. KUBOTA, "PADRE2—Fortran Precompiler for Automatic Differentiation and Estimates of Rounding Error," in *Computational Differentiation: Techniques, Applications, and Tools*, SIAM, Philadelphia, pp. 367–374 (1996).
7. R. GIERING, T. KAMINSKI, "Applying TAF to Generate Efficient Derivative Code of Fortran 77-95 Programs," *PAMM*, **2**(1), 54–57 (2003).
8. J. E. HORWEDEL, "GRESS, A Preprocessor for Sensitivity Analysis of Fortran Programs," in *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, A. GRIEWANK, Ed., SIAM, Philadelphia, pp. 243–250 (1991).
9. J. E. HORWEDEL, *GRESS Version 2.0 User's Manual,* ORNL/TM-11951, Oak Ridge National Laboratory, Oak Ridge, Tenn. (November 1991).