ANALYSIS OF CROSS-COMPILER COMPATIBILITY FOR A CROP GROWTH SIMULATION MODEL USING C++ AND FORTRAN

ANALISI DI COMPATIBILITA DEL CROSS-COMPILATORE PER UN MODELLO DI SIMULAZIONE DELLA CRESCITA DELLE COLTURE UNTILIZZANDO C++ E FORTRAN

Kwang Soo Kim[1,2,*], Haneul Jeong[1]

[1] Affiliation – Department of Plant Science, Seoul National University, 151-742, Seoul (Korea)
[2] Affiliation – National Center For AgroMeteorology, 151-742, Seoul, (Korea)
*luxkwang@snu.ac.kr

Abstract

Crop growth models have been used to simulate biophysical processes for a variety of crops. These crop models are often operational under a limited number of computer operating systems. Those crop models are written in Fortran, which could cause inconsistency between simulation results under different computing environments. For future development, open source paradigm would benefit crop growth models. Nevertheless, the issues associated with model compatibility across a range of computing environments would make it challenging to seize the great opportunity.

We used DSSAT-CSM to examine differences in simulation result by the simulation model built using computer programming languages including C++ and Fortran. The results showed that the simulation models based on C++ had better cross-compiler compatibility than the model based on Fortran. This suggested that crop growth model written in C++ would facilitate open-source software development for crop growth models.

Keywords: crop growth model, open source, cross-compiler.
Parole chiave: modelli di crescita, open source, cross-compiler.

Introduction

A variety of crop growth models has been used for simulation of biophysical processes between soil, crop, and the atmosphere. These crop models are often limited to a small number of computer operating systems, e.g., Microsoft Windows. Open-source paradigm provides great opportunity for development of a crop growth model. However, most of crop growth models are written in Fortran, which could have inconsistency in simulation depending on computing and programming environments. Crop models written in more advanced programming languages would have better consistency irrespective of computing environments. We examined cross-compiler compatibility between crop models built using two different computer programming languages.

Materials and Methods

DSSAT-CSM (the Decsion Support System for Agrotechnological Transfer – Cropping System Model) version 4.02 was used to compare simulation results depending on compilers and programming languages. In the present study, outputs from rice growth simulation were analyzed. A total of 39 simulations from five experiment configuration files were conducted for each combination of computer programming languages and compiler.

To compile the original version of DSSAT-CSM, both proprietary and open-source versions of compilers were used under a Linux operating system. To compile Fortran code of DSSAT-CSM, gfortran compiler (ver. 4.5.1: gcc.gnu.org), which is an open-source compiler within GNU compiler collection. Intel Fortran compiler (Intel Corporation, Santa Clara, CA) were also used to compile the Fortran code of DSSAT-CSM.

To examine cross-compiler compatibility in C++, DSSAT-CSM was reconstructed replacing Fortran codes in DSSAT-CSM with C++ codes. Repetition block between DO and GOTO statement in Fortran code was rewritten using for statement in C++. Modules for taking input parameters into DSSAT-CSM was redesigned and implemented in order to minimize file input and output systems.

C++ code of DSSAT-CSM was compiled using both C++ compiler (g++) within the GNU compiler collection and Intel C++ compiler. The compilation was performed under OpenSUSE (ver. 11.4), which is one of Linux operating systems. Simulation results were analyzed by DSSAT-CSM model built from computer programming languages and compilers. Because a large number of variables are included in the DSSAT-CSM, compatibility between models under different computing environments were evaluated based on the variables analyzed in Thorp et al. (2012). These variables include anthesis date (ADAT), maturity date (MDAT), maximum leaf area index (LAIX), canopy weight at maturity (CWAM), cumulative evapotranspiration at maturity (ETCM). ADAT and MDAT were used as indicators for the numerical accuracy of crop phenology. Comparison between values of LAIX and CWAM indicated the accuracy in leaf area development and CWAM between models. ETCM was used as an indicator of numerical accuracy in water balance simulation.

At first, compatibility between Fortran and C++ versions of DSSAT-CSM was examined using open source compilers. Then, cross-compiler compatibility was examined for both Fortran and C++. Differences in variables between models were quantified using mean difference (MD) and mean absolute difference (MAD) between models built using different compilers as follows:

$$MD \;\; = (Var_m - Var_n) / N \text{ and} \qquad \text{(eq. 1)}$$

$$MAD = | Var_m - Var_n| / N, \qquad \text{(eq. 2)}$$

where Var and N indicate simulation outputs of a variable and total number of treatment in simulations, respectively. m and n represent individual compilers used to compile DSSAT-CSM. t-value was also calculated to provide a simple indication of degree of agreement between simulation outputs.

Results and Discussion (Times New Roman 10 Bold)

C++ version of DSSAT-CSM model had similar simulation outputs to Fortran version of the model. However, simulated phenology dates, e.g., ADAT and MDAT, were different in one experiment configuration. Thus, analysis of simulation outputs were performed for two groups depending on whether phonological date agreed between models built using C++ and Fortran or not.

Simulation outputs of C++ version of DSSAT-CSM were identical for five variables between open-source compiler and proprietary compiler (Tab. 1). In contrast, the values of CWAM and LAIX differ between the models built using Fortran compilers. For example, the DSSAT-CSM model built using Intel Fortran compiler tends to have smaller CWAM values than that using GNU fortran compiler (data not shown).

Under experimental conditions where phonological dates were identical between Fortran and C++ version of DSSAT-CSM models, simulation outputs for CWAM had smaller MAD between the Fortran models built from GNU Fortran compiler and the C++ version of the model. On the other hand, MAD of the LAIX output was less between the Fortran version of model built using Intel Fortran compiler and the C++ version of the model.

In the case of experiment configuration where phonological dates were different between Fortran and C versions of DSSAT-CSM model, C++ version of the model had slower phonological changes than Fortran version of the model. Between Fortran version of the models, LAIX was considerably low and CWAM was identical between two models.

Considering fact that phonological dates differ by experiment conditions between C++ version and Fortran version of the DSSAT-CSM model, there was discrepancy between C++ code and Fortran code, which require further examination of source code for C++ version model.

Our results suggested that it would be preferable to use the C++ version of the DSSAT-CSM model when DSSAT-CSM model is developed under open-source paradigm because simulated outputs were identical irrespective of C++ compilers. Still, more validation and debug are needed for the C++ version of the DSSAT-CSM model because there was difference in simulated phonological dates.

Tab. 1 - Mean absolute differences between DSSAT-CSM models built using open-source and proprietary compilers.
Tab. 1 - Media delle differenze assolute tra i modelli DSSAT-CSM costruiti usando compilatori open-source o proprietari.

	icpc/[a] g++[b]	ifort/ gfortran	g++/ ifort	g++/ gofrtran
Same[c]	(n=35)			
ADAT	0	0	0	0
MDAT	0	0	0	0
LAIX	0	4.8E-04	3.1E-04	4.0E-04
CWAM	0	1.9E+00	3.1E+00	2.7E+00
ETCM	0	0	0	0
Different	(n=4)			
ADAT	0	0	2.5E+00	2.5E+00
MDAT	0	0	4.0E+00	4.0E+00
LAIX	0	5.7E-07	1.6E-01	1.6E-01
CWAM	0	0	4.5E+02	4.5E+02
ETCM	0	0	2.1E+02	2.1E+02

a icpc and ifort represent proprietary Intel C++ compiler and fortran compiler, respectively.
b. g++ and gfortran represent open-source GNU C++ compiler and fortran compiler, respectively.
c. “Same” and “Different” represents experiment configurations in which phonological date was same and different between simulation using the model based on C++ and Fortran codes, respectively.

Conclusions

The DSSAT-CSM model rewritten in C++ is better adapted to diverse programming environments and potentially computing environments. Simulation outputs of variables that indicated crop phonology, leaf area development, and water balance were identical in the C++ version of the model although it was compiled using both open-source or proprietary compilers. Currently, the C++ version of DSSAT-CSM model needs more validation and debug. Once validation and debugging of the C++ version of the model are completed, it is preferable to use the model as a base for open-source development of the DSSAT-CSM.

References

Thorp K.R., White, J.W., Porter, C.H., Hoogenboom, G., Nearing G.S., French A.N. 2012. Methodology to evaluate the performance of simulation models for alternative compiler and operating system configurations. Computers and Electronics in Agriculture. 81: 62-71.