Finding the Root Causes of Statistical Inconsistency in Community Earth System Model Output

Daniel Milroy[1], Dorit Hammerling[2], and Alison Baker[1]

[1]Affiliation not available
[2]National Center for Atmospheric Research

November 22, 2022

Abstract

Baker et al (2015) developed the Community Earth System Model Ensemble Consistency Test (CESM-ECT) to provide a metric for software quality assurance by determining statistical consistency between an ensemble of CESM outputs and new test runs. The test has proved useful for detecting statistical difference caused by compiler bugs and errors in physical modules. However, detection is only the necessary first step in finding the causes of statistical difference. The CESM is a vastly complex model comprised of millions of lines of code which is developed and maintained by a large community of software engineers and scientists. Any root cause analysis is correspondingly challenging. We propose a new capability for CESM-ECT: identifying the sections of code that cause statistical distinguishability. The first step is to discover CESM variables that cause CESM-ECT to classify new runs as statistically distinct, which we achieve via Randomized Logistic Regression. Next we use a tool developed to identify CESM components that define or compute the variables found in the first step. Finally, we employ the application Kernel GENerator (KGEN) created in Kim et al (2016) to detect fine-grained floating point differences. We demonstrate an example of the procedure and advance a plan to automate this process in our future work.

Finding the Root Causes of Statistical Inconsistency in Community Earth System Model Output IN41C-0055 AGU 2017

Dorit M. Hammerling[1], Daniel J. Milroy[2], and Allison H. Baker[1]

[1]National Center for Atmospheric Research and [2]University of Colorado, Boulder

Introduction

Baker et al. [1] developed the Community Earth System Model Ensemble Consistency Test (CESM-ECT) to provide a tool for software quality assurance by determining statistical consistency between an ensemble of CESM outputs and new test runs. The test has proved useful for detecting statistical difference caused by compiler bugs and errors in physical modules, but does not indicate the causes of statistical difference. The CESM is a vastly complex model comprised of millions of lines of Fortran code which is developed and maintained by a large community of software engineers and scientists. Any root cause analysis is correspondingly challenging. We propose a new capability for CESM-ECT: identifying the sections of code that cause statistical distinguishability. We focus on the first model time steps, as divergence between the ensemble and modified runs occurs rapidly, and we wish to detect differences as early as possible. Figure 1 represents our process for finding the root causes, which we detail in the following sections.

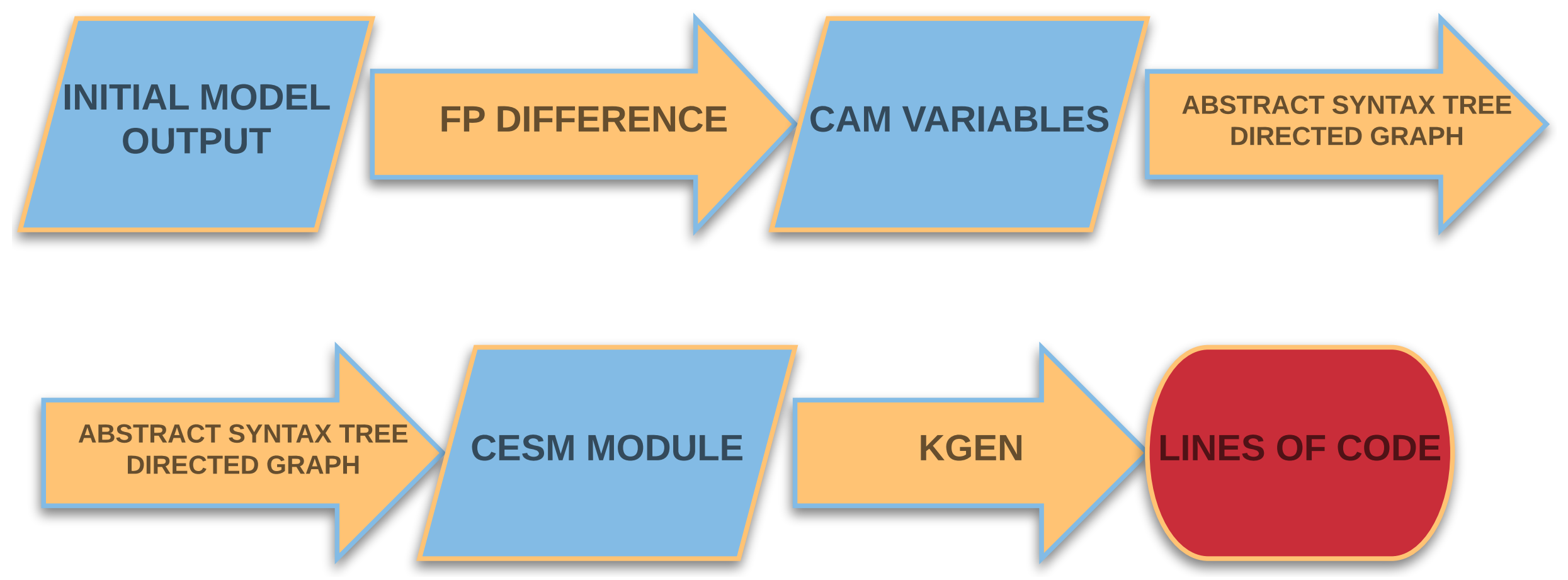


Figure 1: Workflow for identifying the root cause of statistical inconsistency in the CESM.

Overview

The CESM-ECT determines if new model output is statistically distinct from an accepted ensemble by examining principal components (PCs) of the Community Atmospheric Model (CAM) output variables. From a determination of distinguishability in terms of PCs, our first challenge is to identify the untransformed CAM variables that manifest differently between experimental and ensemble outputs. Next we perform static source code analysis to identify code sections of CESM that define, compute, or modify the variables found in the first step. We achieve this by constructing an Abstract Syntax Tree (AST) representation of the code, which we then convert into a directed graph. With graph analysis we can locate crucial code modules and subprograms. Finally, we will employ the Kernel Generator (KGen) application created in Kim et al. [2] to detect fine-grained floating point differences at runtime in the selected subprograms and identify responsible code sections.

Contact Information

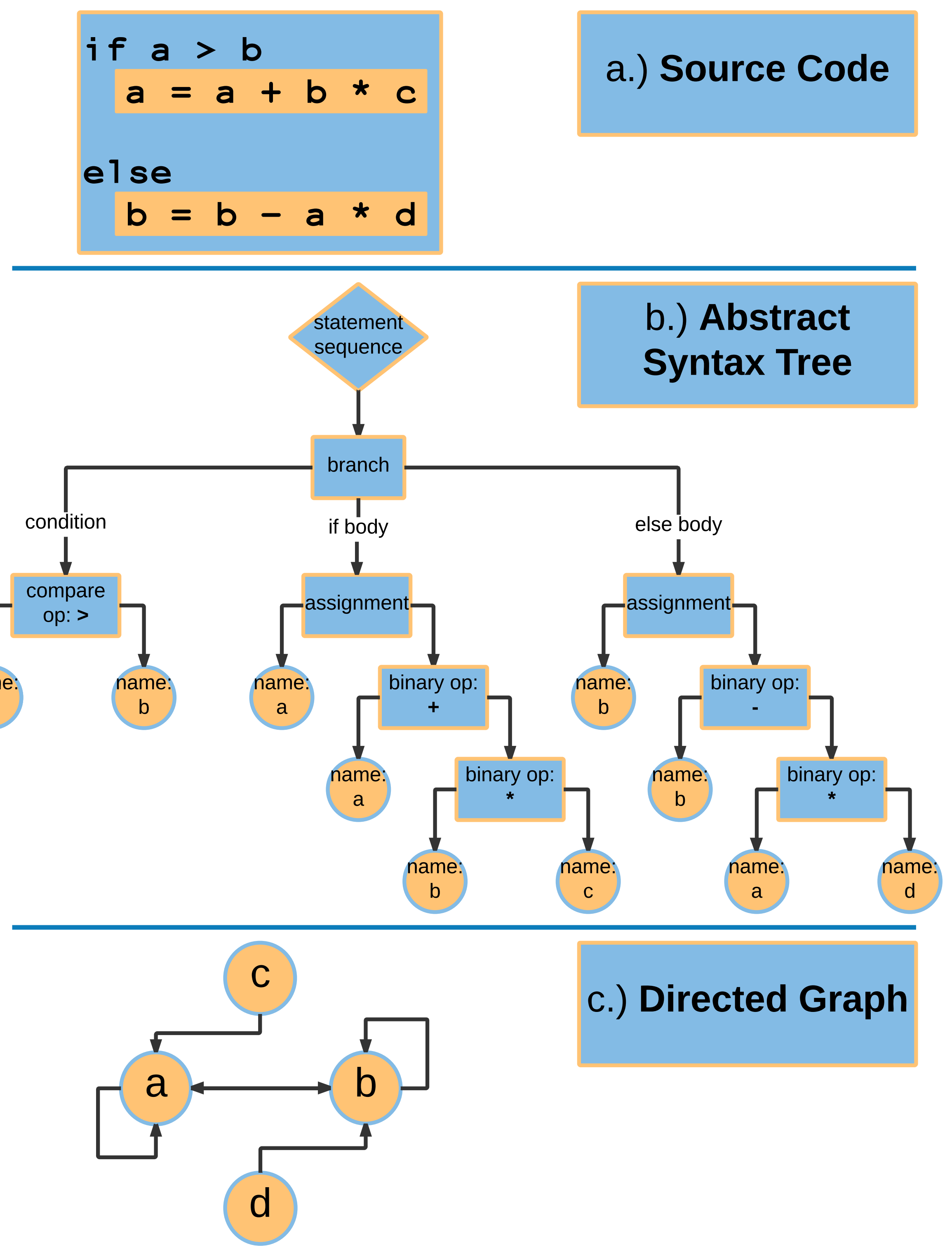
- National Center for Atmospheric Research: {dorith, abaker}@ucar.edu
- University of Colorado, Boulder: daniel.milroy@colorado.edu

Identifying CAM Variables

To find CAM variables influenced by a modification to CESM, we run the modification with 30 different $O(10^{-14})$ K perturbations to the initial atmospheric temperature field. We perform 30 runs as a sanity check to ensure that the response is the same across perturbations. Double precision floating point outputs are written for each CESM time step from 0 to 10, inclusive. We then compare the area-weighted global mean values of the ensemble with those of the experimental set for each variable at each time step and each perturbation. The comparison consists of computing whether values are equal after being rounded to the same number of significant figures. In every case we have examined, the set of different variables is the same across perturbations, so we directly compare the null perturbation of the ensemble and experiment. This simple method of value comparison can flag a large percentage of the CAM variables, in which case we may have to explore alternative ways to reduce this number for tractable analysis.

Illustration

Generating an AST is typically done to compile source code. However, we use an AST to build a graphical representation of variable relationships in the CESM.


```
if a > b
  a = a + b * c

else
  b = b - a * d
```

Figure 2: Example statement in three forms: a.) source code, b.) source code converted to an AST, and c.) AST assignment statements into a directed graph.

Test Case

The example considered here is the "CLM_ALBICE_00" experiment from [4], which CESM-ECT finds to produce statistically distinguishable output. This experiment is a modification to the CLM parameter which controls the fraction of incident solar radiation absorbed by glacial ice. We set the visible and near-infrared albedos to 0, ensuring that the ice absorbs all radiation in this range of wavelengths. Note that this modification to the CLM affects 10 CAM variables by the second time step, as can be seen in Figure 3. We proceed with the objective of identifying the statements that modify the 10 variables.

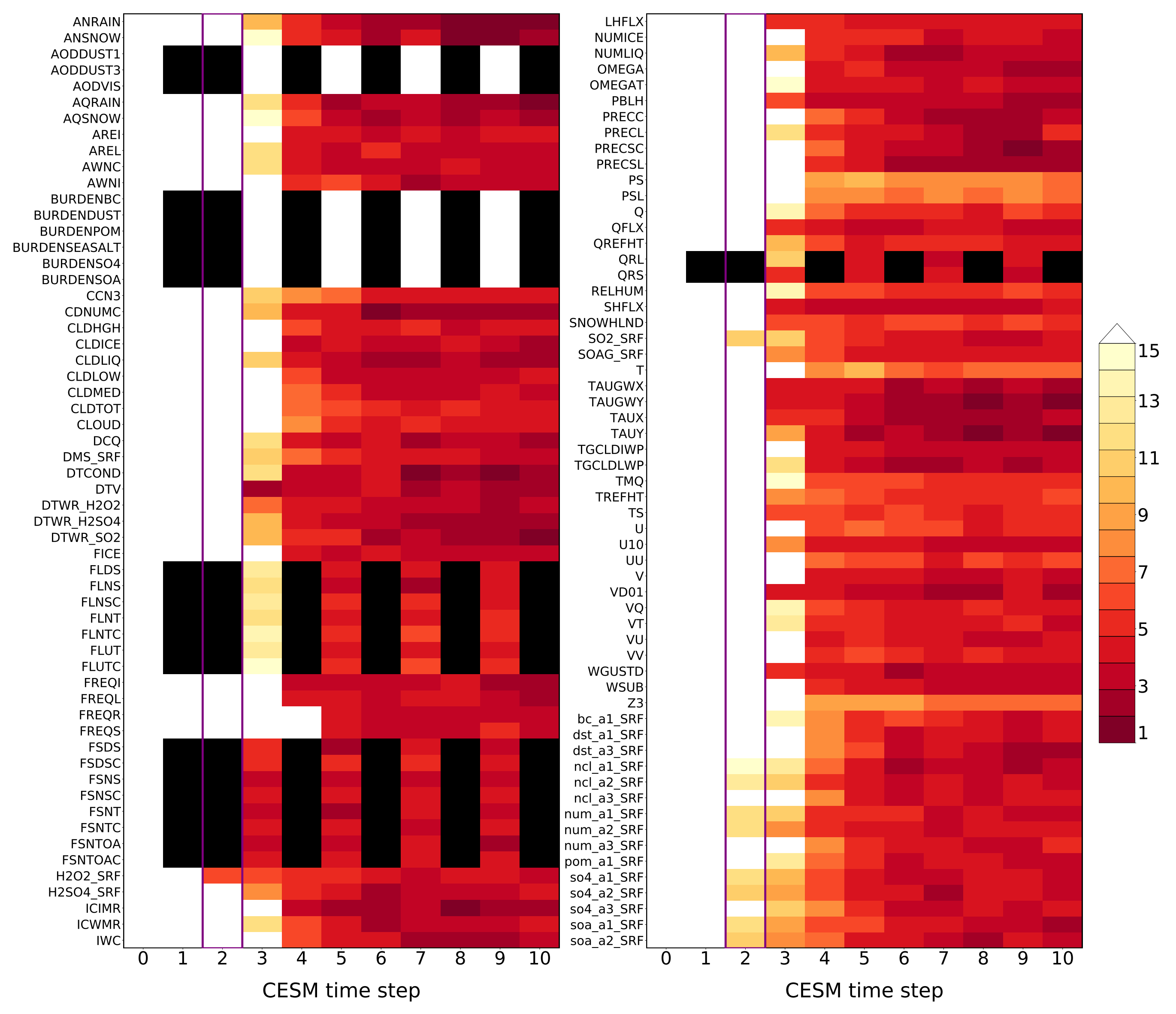


Figure 3: CLM_ALBICE_00 experiment from [4] compared to an unmodified run. The vertical labels are CAM variables. The color designates equality after being rounded to the indicated number of significant figures. A greater number indicates greater numerical similarity. Black signifies that the variables are not computed. Note that there is no difference between the unmodified and CLM_ALBICE_00 runs prior to time step 2.

Initial Findings

From the 10 CAM variables identified in time step two of Figure 3, the AST graph tool identifies all assignment statements that modify the 10 variables. It forms the induced subgraph on these edges and computes the eigenvector centrality of the subgraph. This ranks the nodes that modify the CAM output variables by relative centrality, which by extension admits an ordering of assignment locations.

AST to Directed Graph

We analyze several classes of AST statements: subroutines and functions, uses, interfaces, and assignments. We are principally interested in assignments, as they alter variable values. However, the other statement classes are of ancillary interest, as variables cannot be connected correctly without them. Assignments are encoded into a directed graph, with the assigned variable (left hand side) as the target, and the expression (right hand side variables) as the sources. See Figure 2 for a simple example. After being encoded from an AST into a directed graph, the CESM consists of approximately 130,000 nodes (variables) and 300,000 edges indicating relationships through assignment. Figure 4 is a representation of the MG1 microphysics module as a subgraph of CESM.

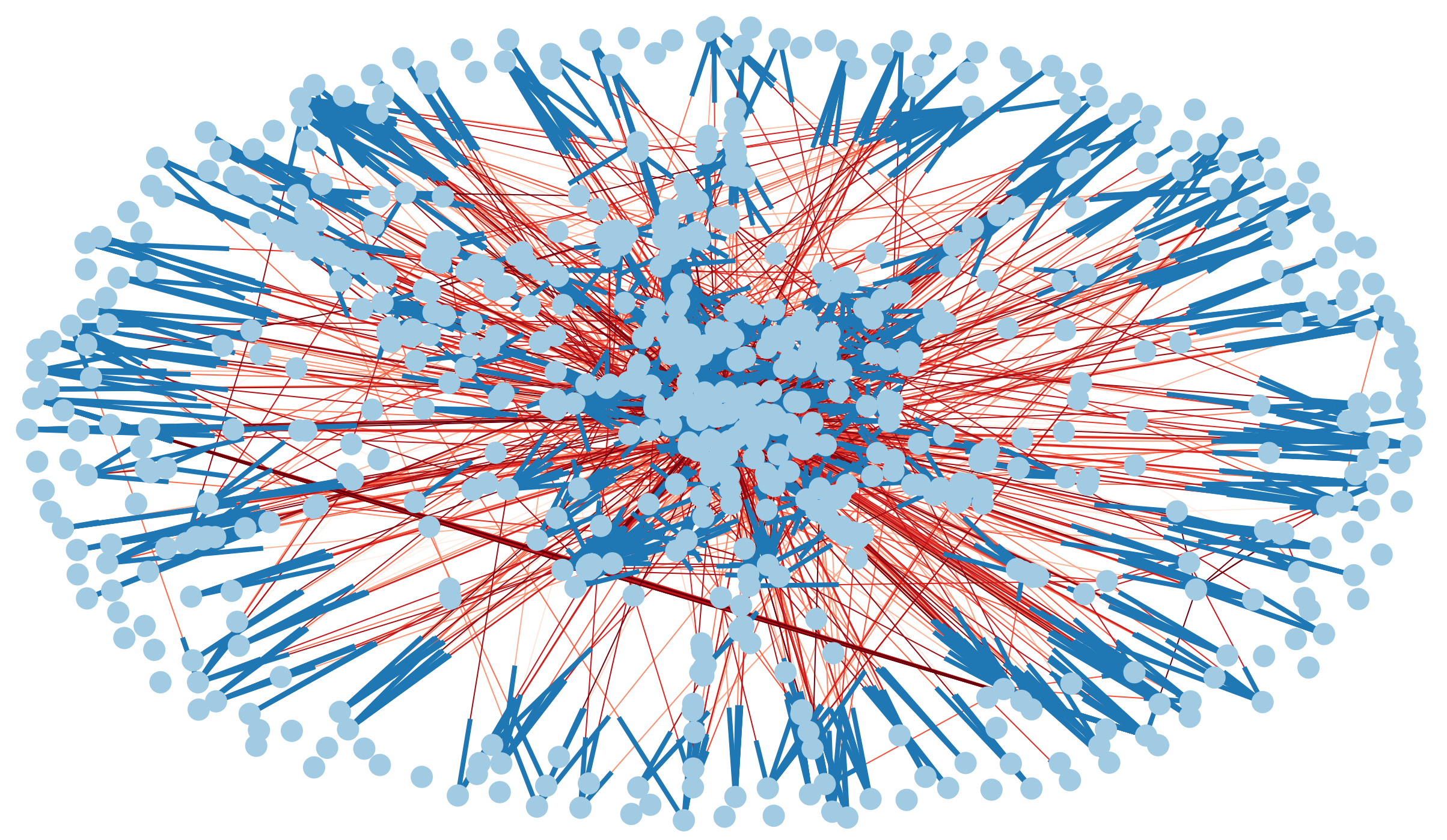


Figure 4: CAM MG1 microphysics module as a directed subgraph of CESM.

Future Work

With a list of critical code regions obtained via the procedure described above, we will use KGen capabilities to extract these sections and compile them as independent executables. KGen can then be used to identify variables whose RMS values are significantly different than those computed on an executable compiled on an accepted machine and configuration. Given four CAM output variables known to be related to statistical inconsistency on the Mira supercomputer [3], the AST digraph method correctly indicates that relevant assignments occur in the MG1 microphysics module (Figure 4). Furthermore, since Fused Multiply-Add (FMA) instructions were determined to cause the statistical distinctness, our goal is to find subsections of MG1 (and other CESM modules) that are sensitive to FMA.

References

[1] A. H. Baker, D. M. Hammerling, M. N. Levy, H. Xu, J. M. Dennis, B. E. Eaton, J. Edwards, C. Hannay, S. A. Mickelson, R. B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein, and D. Williamson. A new ensemble-based consistency test for the Community Earth System Model. *Geoscientific Model Development*, 8:2829–2840, 2015.

[2] Y. Kim, J. Dennis, C. Kerr, R. R. P. Kumar, A. Simha, A. Baker, and S. Mickelson. KGEN: A Python Tool for Automated Fortran Kernel Generation and Verification. *Procedia Computer Science*, 80(Supplement C):1450 – 1460, 2016. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA.

[3] D. J. Milroy, A. H. Baker, D. M. Hammerling, J. M. Dennis, S. A. Mickelson, and E. R. Jessup. Towards Characterizing the Variability of Statistically Consistent Community Earth System Model Simulations. *Procedia Computer Science*, 80(Supplement C):1589 – 1600, 2016. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA.

[4] D. J. Milroy, A. H. Baker, D. M. Hammerling, and E. R. Jessup. Nine time steps: ultra-fast statistical consistency testing of the Community Earth System Model (pyCECT v3.0). *Geoscientific Model Development Discussions*, 2017:1–22, 2017.