Workshop on Correctness and Reproducibility for Climate and Weather Software
November 9-10, 2023
National Center for Atmospheric Research (NCAR), Boulder, CO

Workshop Co-chairs:
Allison Baker, National Center for Atmospheric Research
Alper Altuntas, National Center for Atmospheric Research

Organizing Committee:
Ilene Carpenter, Hewlett Packard Enterprise
Brian Dobbins, National Center for Atmospheric Research
Michael Duda, National Center for Atmospheric Research
Dorit Hammerling, Colorado School of Mines
Thomas Hauser, National Center for Atmospheric Research
Karsten Peters-von Gehlen, Deutsches Klimarechenzentrum GmbH (DKRZ)

Abstracts

Thursday Morning Session

KEYNOTE: Models, Data, and Wisdom: How do we know when to trust a climate model?
Steve Easterbrook, School of the Environment and Department of Computer Science, University of Toronto

For a computational model, correctness is a slippery concept. A model is by definition, a simplification, so assessing correctness against observational data requires subjective judgments about what counts as "good enough". And sometimes, when a model disagrees with observational data, it's the data that turns out to be wrong. In this talk, I will examine the question of model correctness from philosophical, engineering, empirical, and scientific perspectives, and I will argue it's more productive to think of a model as having a "scope of applicability" —aspects of the world where the model applies well and can be used to explain and predict. I will then give an overview of the many day-to-day practices in climate modeling labs that collectively contribute to our current understanding of the scope of applicability for climate models, and assess how, collectively, these practices give confidence in the models while simultaneously shedding light on their weaknesses.

Bio: Steve Easterbrook is the Director of the School of the Environment and Professor of Computer Science at the University of Toronto. He received his Ph.D. (1991) in Computing from Imperial College in London (UK), and joined the faculty at the School of Cognitive and Computing Science, University of Sussex. From 1995-99, he was lead scientist at NASA´s Katherine Johnson Independent Verification and Validation Facility in West Virginia, where he investigated software verification on the Space Shuttle Flight Software, the International Space Station, and the Earth Observation System. He moved to the University of Toronto in 1999. His current research is in climate informatics, where he studies how climate scientists develop computational models to improve their understanding of earth systems and climate change, and the broader question of how that knowledge is shared with other

communities. His new book, "Computing the Climate: How we know what we know about climate change", was published by Cambridge University Press in August 2023.

Component Level Regression Testing in a Hierarchical Architecture
Thomas Clune, NASA Goddard Space Flight Center

The Goddard Earth Observing System (GEOS) is an Earth system model consisting of a large suite of individual model components that can be coupled in a flexible manner to investigate a variety of Earth science issues. Specific GEOS model configurations are composed as a hierarchical collection of components based on the Earth System Modeling Framework (ESMF).

Regression testing of GEOS is currently limited to (1) full system tests that are poor at isolating specific defects and (2) a suite of unit tests which have very limited coverage. As part of our approach to improve upon the current testing situation, we have prototyped the capability to perform regression tests on individual GEOS components by leveraging and extending existing checkpoint/restart capabilities.

In our implementation, each ESMF component has 3 states: Import (what it needs to run), Export (which it needs to provide to other components), and Internal (the component state proper). By capturing, Import, EExport and Internal states for a given component during a full run of GEOS, a generic driver can then rerun the component offline and compare expected exports with those that have been saved.

The hierarchical structure of GEOS introduces an interesting wrinkle when trying to test components that in turn drive interacting child components. To fully isolate a parent component we use the approach of software mocks, in which the exports of children are also saved during the initial capture run of GEOS. Then when testing the parent component, the children components are replaced by a generic mock component that produces exports from the previously saved data and ensures that all interdependencies among children components are satisfied.

High Performance Climate and Weather Benchmark (HPCW): a framework for reproducible benchmarks
David Guibert, Center for Excellence in Performance Programming, Eviden

ESiWACE3, the third phase of the Centre of Excellence (CoE) in Simulations of Weather and Climate in Europe, is funded by the EuroHPC Joint Undertaking to provide support for the wider community of weather and climate modelling in the use of state-of-the-art supercomputers and new architectures.

We will present the High Performance Climate and Weather Benchmark (HPCW) developed in ESCAPE2, ESiWACE2 and, now, in ESiWACE3. HPCW is meant to isolate key elements in the workflow of weather and climate prediction systems to improve performance and allow a detailed performance comparison for different hardware. HPCW contains some European ESM models (Icon, Nemo, RAPS) and some mini-applications in a common framework to allow collaboration with vendors on co-design aspects and to allow standard comparisons for different architectures.

First, we will introduce the motivation and initial technical choices that lead us to develop HPCW as a CMake-based framework to be able to compile all the NWP models included on top of their own build systems and be agnostic to the software environment and scheduler of the HPC sites. We will focus on how the libraries required can be compiled by the framework itself or used from the software stack already installed by the user support.

The effort of supporting all combinations of compiler versions, vendor libraries (like MPI) and/or NWP models lead us to consider the usage of the Spack package manager. We will give an overview of using of the new SPACK recipes to build NWP models and use them to get performance metrics (like time to solution, energy to solution, flops...) on different hardware. This is a key element of one the goals of ESiWACE3 to provide a common software stack for European ESM.

Correctness Challenges in HPC and ML

Harvey Dam, Ganesh Gopalakrishnan, Department of Computer Science, University of Utah

I will begin by summarizing the ComPort project that is into its third year, as well as the CSC'23 workshop, both of which are attempts at grappling with usable notions of correctness. From both these vantages, one thing is becoming clear: the unavailability of challenge problems without well-specified and broadly applicable correctness goals is hampering progress. For instance, one talks about low precision computation or lossy data compression to help reduce data movement. Yet, metrics such as "the Physics does not change" are not sufficiently broad to help correctness researchers. Another example is that HPC benefits from ML surrogate models - but what correctness requirements are imposed on these ML models? Could those be offered as specific ML-correctness challenges? Clearly, the Climate community is very aware of these aspects and has significant experience with correctness. Would they be willing to help the correctness community with more objective challenges? I will try and propose the kinds of HPC and ML challenges that I think would be very helpful.

Reliable and reproducible Earth System Model data analysis with ESMValTool

Valeriu Predoi, Bouwe Andela, NCAS-CMS, University of Reading

ESMValTool is a software tool for analyzing data produced by Earth System Models (ESMs) in a reliable and reproducible way. It provides a large and diverse collection of “recipes” that reproduce standard, as well as state-of-the-art analyses. ESMValTool can be used for tasks ranging from monitoring continuously running ESM simulations to analysis for scientific publications such as the IPCC reports, including reproducing results from previously published scientific articles as well as allowing scientists to produce new analysis results. The tool has been developed by a heterogenous community of scientists and research software engineers in three different programming languages (Python, NCL, and R), has over 200,000 lines of code, and depends on approximately 650 other software packages. This makes software reliability and reproducibility of results rather challenging.

When we started developing version 2.0 of the tool, five years ago, we started out by adopting open science and modern software development techniques as well as best practices to ensure software quality and reproducibility, both at software, and at scientific output levels. All ESMValTool development is done in public repositories on GitHub, tests are run using CircleCI and GitHub Actions, and code quality services like Codecov and Codacy are deployed. However, we quickly realized that all these steps are not sufficient to guarantee reliability of the software and reproducible results because our contributors have very different software development practices.

Our main strategy for ensuring reliability is modular design. This comes back at various levels of the tool. We try to separate commonly used functionality from “one off” code, and make sure that commonly used functionality is covered by unit and integration tests, while we rely on regression testing for everything else. We also use comprehensive end-to-end testing for all our “recipes” before we release new versions. The computational engine framework that runs the analyses provided by ESMValTool has been detached from the scientific analyses into a separate software package, called ESMValCore. This is much smaller in size (~30,000 lines of pure Python code), is maintained by the most technically skilled members of the community, and has very good test coverage with unit,

integration, and regression tests run on the aforementioned continuous integration services. It provides a “recipe” format that allows scientists to specify the input data and how it should be pre-processed before handing it off to the actual scientific analysis script run (and developed) by the scientist. Since the ESMValCore code lives in a separate Git repository, the possibility of altering the framework to run analyses for a single use case, and thus potentially break the workflow developed by others, is much smaller. As a community, we try to use and contribute to existing libraries wherever we can, as these are typically more reliable and better maintained than custom code.

Instead of requiring bitwise identical results when regression testing the analysis scripts, we have implemented a tool that smartly handles various file formats, e.g. it compares NetCDF data files with floating point numbers in them with limited precision, and uses image comparison algorithms to avoid triggering test failures over minute changes in figures. This greatly reduces the need for ‘human testing’. ESMValTool is thus a community driven tool, with built-in robustness through modularity and a testing strategy that has been tailored to match the technical skills of its contributors.

Testing approach for porting legacy 4-mode Modal Aerosol Model (MAM4) to C++/Kokkos

Balwinder Singh, Atmospheric Sciences and Global Change Division, Pacific Northwest National Laboratory

The atmospheric component of version 4 of the Energy Exascale Earth System Model (E3SM) is designed to run on cloud-system resolving scales (i.e., x ~3 km) on GPU-enabled Exascale machines. The atmospheric component, known as EAMxx, is completely rewritten in C++, leveraging the Kokkos library to achieve performance portability across various platforms. However, it currently lacks a fully prognostic treatment of aerosols and relies on a simple prescribed approach using climatological monthly means. It is well known that the impact of aerosols on radiation, cloud properties, and precipitation can exhibit significant variations between climatological means and prognostic aerosols treatment. To address this shortcoming, the EAGLES (Enabling Aerosol-cloud interactions at GLobal convection-permitting scalES) team is actively working on the ambitious task of porting the Fortran codes of the MAM4 (4-mode Modal Aerosol Model) prognostic aerosol model to the C++/Kokkos framework. The ported MAM4, referred to as MAM4xx, will interact with other physics modules of EAMxx (the host model) to enable prognostic aerosols in EAMxx’s simulations. In this presentation, I will outline the workflow we are employing for a successful port of the legacy Fortran codes to the C++/Kokkos framework. The key emphasis will be on the end-to-end testing methods that we developed and implemented to ensure an accurate port of the Fortran codes.

Verification of the ICON model with the GT4Py dycore - challenges and insights

*Abishek Gopal**
Institute for Atmospheric and Climate Science, ETH Zurich

In this talk, we present some of the challenges in model verification during the development of the ICON atmospheric model with the GT4Py dynamical core. GT4Py is a Domain Specific Language (DSL) intended for weather and climate codes, enabling a separation of concerns between higher-level model development and lower-level architecture-specific optimizations. In the EXCLAIM project, we have been refactoring the dynamical core of the Fortran-based ICON model to call computational stencils written in the GT4Py front-end language. Due to the fast-paced development of the GT4Py framework and the ICON model, ensuring the correctness of the ICON-GT4Py dynamical core requires a verification framework that is computationally efficient and robust.

We use a tiered verification approach which ranges from runtime verification of stencil-wise computations, to post-model run system tests that include analyzing the growth of perturbations in the first 10-15 timesteps, as well as statistical testing on a grid-cell level of 10-day ensemble simulations. (Zeman & Schär, 2022). We illustrate the

value of employing multi-tiered tests during the process of debugging an error inadvertently introduced upstream to our development.

Thursday Afternoon Session

KEYNOTE: Earth system models of the future
Peter Dueben, Earth System Modelling Section, European Centre for Medium Range Weather Forecasts

The domain of Earth system modelling is currently seeing rapid changes. Models are going through a digital revolution and a significant increase in resolution caused by a drift to heterogeneous hardware in high-performance computing, machine learning is creating serious competition for physics-based Earth system models, and students do not want to learn Fortran anymore. This talk will explain the challenges and outline how Earth system modelling may look like in 10 years from now.

Bio: Peter is the Head of the Earth System Modelling Section at the European Centre for Medium Range Weather Forecasts (ECMWF) developing one of the world's leading global weather forecast models. Before, he was AI and Machine Learning Coordinator at ECMWF and University Research Fellow of the Royal Society performing research towards the use of machine learning, high-performance computing, and predictability in weather and climate simulations. Peter is coordinator of the MAELSTROM EuroHPC-Joint Undertaking project that is implementing a software/hardware co-design cycle to optimise performance and quality of machine learning applications in the area of weather and climate science, and work-package leader of the ESiWACE-3 Centre of Excellence. Before moving to ECMWF, Peter has written his PhD thesis at the Max Planck Institute for Meteorology and worked as PostDoc with Tim Palmer at the University of Oxford.

A Theory of Scientific Programming Efficacy
Michael Coblenz, Department of Computer Science, UC San DIego

Many scientists create and maintain complex software artifacts in their work. As a result, their ability to construct useful models and arrive at correct conclusions depends on their efficacy in writing software. We interviewed 25 scientists and support staff and analyzed the results, identifying six factors that contribute to scientists' abilities to write software successfully. In this talk, I will present the theory and identify potential opportunities to adapt software engineering tools and processes with the aim of making scientists more effective at writing software.

An overview of the MOM6 development cycle
Marshall Ward, Geophysical Fluid Dynamics Lab, NOAA

I will present an overview of the procedures behind the maintenance of the MOM6 ocean model source code. MOM6 has a growing community, and a responsibility to provide accurate and reproducible answers for its users. We achieve this by rigorous testing, disciplined repository management, and a collaborative process for code updates.

Floating point operations in the model are required to be bit reproducible, and I review our techniques for achieving this. I show how these methods can be extended to preserve the dimensional and rotational invariance of our solvers. Testing of these requirements are integrated into the CI system of MOM6. I then present strategies for maintaining a clean, linear history of the model's codebase. Finally, I show how GFDL's repository exists as one

of many hubs, and how we maintain a common codebase together as a community in an active research environment.

Challenges in Ensuring Reproducibility for Machine Learning Weather Model Training and Deployment
David John Gagne, Computational and Information Systems Lab, NCAR

As machine learning models in weather and climate transfer from research to operational environments, many organizations are contending with a host of questions about how to implement machine learning models within complex operational pipelines and ensure that model performance is maintained through updates to software, hardware, and supporting libraries. Major challenges include saving all of the necessary configuration and metadata to reproduce the training of a model, accounting for sources of stochastic behavior on both CPUs and GPUs, and ensuring that a model trained in Python will produce equivalent results when run in a FORTRAN-based numerical simulation. This presentation will discuss these challenges and available best practices and software to address them.

METplus: The Long and Winding Road to Unified Verification
Tara Jensen, Research Applications Lab, NCAR

With the evolution of weather and climate prediction using well established and burgeoning Earth System modeling frameworks such as NCAR's Community Earth System Model (CESM), System for Integrated Modeling of the Atmosphere (SIMA), and the National Oceanic and Atmospheric Administration (NOAA)'s Unified Forecast System (UFS), verification and evaluation activities are critical for the success of Research to Operations (R2O) across the UFS community. The enhanced Model Evaluation Tools (MET) framework (METplus) is at the core of this expanding cross-section of the community evaluation activities.

The METplus system consists of several components, including the MET, for the computation of verification statistics based on gridded forecasts and either a gridded analysis or point-based observations. The system also incorporates an analysis system for aggregating statistics and plotting graphical results. These tools are designed to be highly flexible to allow for quick adaption to meet additional evaluation and diagnostic needs. A suite of python wrappers have been implemented in METplus to facilitate a quick set-up and implementation of the system, and to enhance the pre-existing plotting capabilities. At NCAR, METplus is used most predominantly by RAL and Mesoscale and Microscale Meteorology (MMM) laboratories, but has recently been coupled with software available at the Atmospheric Composition Observation and Modeling (ACOM) lab. Efforts to integrate it into SIMA are ongoing but currently unfunded.

Nationally and internationally, all components of METplus were recently accepted for installation on the National Oceanic and Atmospheric Administration's (NOAA's) operational high-performance computing platform and is being integrated into the Environmental Modeling Center (EMC) Verification System (EVS). It is also being adopted by the United States Air Force, US Naval Research Laboratory, and Met Office of the United Kingdom and subsequent Unified Model partner organizations. METplus is also a core component in the Developmental Testbed Center (DTC)'s testing and evaluation activities, which test out innovations across many modelling system.

With its roots in evaluation of limited area tropospheric models, METplus has undergone several years of rapid development to better support evaluations on many temporal and spatial scales for a growing number of components of a coupled Earth system model. This presentation will focus on the challenges and successes of

developing the community software suite, including addressing the needs of operational partners, accepting contributions to facility R2O of evaluation methods, and refactoring for the future.

Unit Testing NCEPLIBS
Edward Hartnett, CIRES/NOAA

The NCEPLIBS are required libraries/utilities for UFS and other NOAA applications. Residing in 21 repositories, these codes are decades old and provide critical functionality to NCEP operations. The NCEPLIBS team has added unit testing to the active NCEPLIBS libraries. With very few resources, we have achieved a much higher level of unit testing for many of the NCEPLIBS codes. In this presentation we will share our process, techniques, and lessons learned.

Friday Morning Session

KEYNOTE: Lightweight Formal Methods: The What, Why, and How
John Baugh, Civil Engineering and Operations Research, North Carolina State University

Getting scientific software right is both important and challenging. We contend with long product lifespans and evolving needs, an absence of test oracles when new models are being advanced, and competing objectives of performance, maintainability, and portability. What development processes, quality assurance practices, or design strategies might lead to better, more correct software?

In this talk, I'll describe a lightweight approach to formal methods that developers can add to their toolkits today. A premise and motivating viewpoint I'll take is the central role that modeling can and should play in designing and working with complex artifacts, including scientific software. Unlike attempts at after-the-fact verification, the emphasis is on "design thinking," an inherently iterative process that can benefit from tool support. Using a declarative, lightweight formalism called Alloy, I'll show how we can frame questions and get answers to them through automatic, push-button analysis—a modeling approach that can help us gain a deeper understanding of the structure and behavior of the programs we create.

Bio: John Baugh is a professor of civil engineering and operations research at North Carolina State University. His research focuses on formal methods and verification, and the tailoring of tools and practices to support the specific needs of scientists and engineers. He works on problems in scientific computing, cyber-physical systems, mathematical optimization, and control. Current and past funding sources for his research include the National Science Foundation, US Department of Energy, Department of Homeland Security, Federal Transit Administration, US Environmental Protection Agency, and North Carolina Supercomputing Center. He received the Ph.D. from Carnegie Mellon University in 1989.

What could the next 30 years of software verification in climate science look like?
Dominic Orchard, Department of Computer Science and Technology, University of Cambridge and School of Computing, University of Kent

A significant area of computer science is devoted to developing the theory and practice of program verification. Progress over the last 30 years has yielded a plethora of tools and techniques: automated theorem provers, languages for mechanised proof, model extraction and model checking tools, program synthesisers, static analysis tools, advanced type systems, and languages with first-class verification features. Concurrently, verified

implementations of critical systems software have been produced, e.g., for compilers and operating system kernels. Yet almost none of these techniques are deployed in the context of science. In this talk, we discuss to what extent such techniques could be of use within the climate science community and what would be required to be in a position to reap any of their benefits. Included will be a discussion of recent work on targeted lightweight verification techniques for science.

Parallel reproducibility of the SHYFEM-MPI model

Francesco Carere, Italo Epicoco, Euro Mediterranean Center on Climate Change Foundation (CMCC Foundation)

SHYFEM (System of HydrodYnamic Finite Element Modules) is a hydrodynamic model used in weather & climate research. It solves the hydrodynamic equations using finite elements on an unstructured grid on both local and global scale. Development of the model started around 1985 and is ongoing; CMCC is mainly involved in the MPI-based branch (active since 2015) resulting in the validated “SHYFEM-MPI” model [Mic+22]. It is well known that numerical models suffer from round-off error due to the approximated representation of floating points moreover many architectural optimizations introduce non-determinism in numerical software, some examples are:
1. Compiler optimisation: e.g. fused multiply add, vectorisation, etc.
2. Parallelism: non-deterministic order of communication and calculation
3. Hardware differences: CPU, Cache, GPU, accelerator types
4. Software differences: e.g. settings and versions of external libraries

Even though the sources of non-determinism appear to be different, many of them can be explained by a single effect, namely the loss of the associative rule of floating point operations (FLOP). Every time a compiler optimization or parallelization leads to a different order in the evaluation of FLOP, non-deterministic behaviour is introduced into the software i.e. the outputs may not be bitwise identical. Building bitwise reproducible software introduces loss of computational performance because of obstructed compiler optimization, serialized communications, and so on. It is worth noting here that bitwise reproducibility of software does not eliminate the round-off error introduced by the approximate representation of the floating point; having bitwise reproducible software means that several and different runs are able to exactly reproduce both the model behavior, the numerical uncertainty of the input data and also the floating point approximation error. Is this exactly what we need? Bitwise reproducibility is sometimes seen as highly important for scientific advances [Gen13], the main reason being that it increases trust in published computational results by improving review capabilities. However, the fundamental component of scientific advances are the conclusions drawn from a result and the theory constructed from this, rather than the result itself [Dru09]. Can we, in this light, relax the constraint to have bitwise reproducible software for the sake of computational performance? In this work we focus on the evaluation of SHYFEM-MPI reproducibility. In our perspective, a parallel run is able to reproduce the sequential run when the outputs belong to the same interval of confidence. An optimized parallel run differs from a sequential run only in the different order of FLOP. Our analysis starts by evaluating the error due to the different order of FLOP in the sequential runs, which were executed using a deterministic and sequential version of SHYFEM-MPI, disabling all compiler optimization. We designed an ensemble experiment where each member differs only by a different order of the domain elements leading to a different order of FLOP. The confidence interval of the output distribution provide us with an estimation of the rounding error. To assess the model reproducibility we hence run an ensemble experiment using the fully optimized, parallel version of SHYFEM-MPI where each member is run with different number of cores. In our perspective, the parallel model is reproducible if the confidence intervals of the distributions obtained by both ensemble experiments overlap. In our experiment we used a regional configuration of the model in the Mediterranean Sea around the Greek island Zakinthos and used 10 members for both ensembles (the sequential and the parallel) simulating one year. The results show a strong overlap of both intervals demonstrating that SHYFEM-MPI is able to correctly reproduce the outputs. Finally, a Kolmogorov-Smirnov test

shows quantitatively the similarity of the distribution of the output obtained in parallel to the distribution of the output with grid reshuffling. A future goal could be to use this understanding of the rounding error in SHYFEM-MPI to increase the correctness of the model.

KEYNOTE: Contained Chaos: Quality Assurance for the Community Earth System Model
Dorit Hammerling, Applied Mathematics and Statistics, Colorado School of Mines

State-of-the-science climate models are valuable tools for understanding past and present climates and are particularly vital for addressing otherwise intractable questions about future climate scenarios. The National Center for Atmospheric research leads the development of the popular Community Earth System Model (CESM), which models the Earth system by simulating the major Earth system components (e.g., atmosphere, ocean, land, river, ice, etc.) and the interactions between them. These complex processes result in a model that is inherently chaotic, meaning that small perturbations can cause large effects. For this reason, ensemble methods are common in climate studies, as a collection of simulations are needed to understand and characterize this uncertainty in the climate model system. While climate scientists typically use initial condition perturbations to create ensemble spread, similar effects can result from seemingly minor changes to the hardware or software stack. This sensitivity makes quality assurance challenging, and defining "correctness" separately from bit-reproducibility is really a practical necessity. Our approach casts correctness in terms of statistical distinguishability such that the problem becomes one of making decisions under uncertainty in a high-dimensional variable space. We developed a statistical testing framework that can be thought of as hypothesis testing combined with Principal Component Analysis (PCA). One key advantage of this approach for settings with hundreds of output variables is that it not only captures changes in individual variables but the relationship between variables as well. This testing framework referred to as "Ensemble Consistency Testing" has been successfully implemented and used for the last few years, and we will provide an overview of this multi-year effort.

We are currently delving into the technical details of the PCA analysis step, which involves estimating a high dimensional covariance matrix from samples and then projecting into lower-dimensional space. This notoriously tricky problem is among other things sensitive to the ensemble size, and we are working to better describe the probabilistic properties of our testing framework and improve its robustness. As such, we are aiming to generalize our approach so it can be applied in a straightforward manner to other numerical models featuring high-dimensional spatio-temporal output.

Bio: Prof. Hammerling obtained a M.A. and PhD from the University of Michigan in Statistics and Engineering, followed by a postdoctoral fellowship at the Statistical Applied Mathematical Sciences Institute in the program for Statistical Inference for massive data. She then joined the National Center for Atmospheric Research, where she led the statistics group within the Institute for Mathematics Applied to the Geosciences and worked in the Machine Learning division before becoming an Associate Professor in Applied Mathematics and Statistics at the Colorado School of Mines in January 2019.

Methods and Tools for the Application of UF-ECT to New Climate Models
Teo Price-Broncucia, Department of Computer Science, University of Colorado Boulder

A number of previous works have developed the Ultra Fast Ensemble Consistency Test (UF-ECT) for use with the Community Earth System Model. The UF-ECT enables a user to check consistency between an accepted ensemble and a test set of CESM runs with relatively low computational cost. In this work we present a general methodology

for analyzing the output of an arbitrary earth system model in order to properly adapt the UF-ECT framework. This methodology includes inquiry into the correlation of model outputs, their variability over temporal and spatial scales, and how well they approximate normal distributions over time. These steps allow the user to identify appropriate model outputs and timescales for the test. In addition we provide steps to appropriately modify UF-ECT parameters such as ensemble size and PCA dimension.
This methodology is demonstrated using the atmospheric component of the Model Across Prediction Scales (MPAS) climate model, enabling comparison with CESM and highlighting key properties users should be aware of when applying to new models. Finally, we present a software tool that streamlines the above steps for practitioners.

Ensure the correctness and reproducibility in UFS Weather Model CI
Jun Wang, NOAA NWS/EMC

The Unified Forecast System (UFS) is a community-based, coupled, comprehensive earth modeling system. It supports both weather enterprises and is also the source system for NOAA‘s operational numerical weather prediction applications. Several UFS applications have been implemented into operation in the past several years including GFSv15, GFSv16 GEFSv12 and HAFSv1. New innovations have been integrated to the UFS weather model while developing these operational applications, it is critical to ensure that the new features are implemented correctly. Meanwhile, reproducibility under various configurations is required to support operational implementations and run the system in operation.

In the talk, we will present the capability of the reproducibility configurations in the UFS weather model including restart, threading, domain decomposition, and MPI tasks. We will also discuss the butterfly test, noise test and data compression methods in the UFS weather model. The butterfly test is used to evaluate result change due to platform porting, compiler updates and inessential code updates to ensure that no obvious errors are brought into the system. The noise test is also available to avoid artifacts that bring in unexpected behavior and can be used to evaluate the impact on model stability from a new feature. The impact of data compression on model results, especially lossy compression, will also be discussed. Some future work including using simple ML models for code updates evaluating in UFS weather model CI will also be presented.

Friday Afternoon Session

Towards Ensuring Statistical Climate Reproducibility of Earth System Models in the Exascale Age
Salil Mahajin, Computational Earth Sciences Group, Oak Ridge National Laboratory

Effective utilization of novel hybrid architectures found in near-exascale machines requires code transformations to Earth System Models that may not reproduce the original model solution bit-for-bit. Round-off level differences grow rapidly in these nonlinear and chaotic systems, making it difficult to isolate error-growth from the innocuous growth in round-off level differences. Here, we present results from the application of some classical and modern multivariate two sample equality of distribution tests to evaluate statistical reproducibility of atmosphere and ocean model components of Energy Exascale Earth System Model and the Unified Model. Baseline simulation ensembles are compared to modified ensembles – after a non-b4b change in a model component is introduced – to evaluate the null hypothesis that the two ensembles are statistically indistinguishable. To quantify the false negative rates of the tests, we conduct a formal power analysis using resampling methods with targeted suites of short simulation ensembles. Each such suite contains several perturbed ensembles, each with a progressively different climate than the baseline ensemble - obtained by perturbing the magnitude of a single model tuning

parameter in a controlled manner. The broad power analysis provides a framework to quantify the degree of differences that can be detected confidently by a given ensemble size, allowing developers to make an informed decision on an unintentional non-bit-for-bit change to the solution.

Improvements in Reproducibility Testing Through False Discovery Rate Correction
Michael Kelleher, Computational Earth Sciences Group, Oak Ridge National Laboratory

To evaluate the impact of code changes on model simulated climate in DOE's Energy Exascale Earth System Model (E3SM) a series of non-bit-for-bit tests are conducted nightly. The most computationally expensive of these is the multivariate Kolmogorov-Smirnov test (MVK) which compares two ensembles, each of 30 members, performing K-S tests on the global mean of 120 variables. A threshold for test failure (i.e. that a code change in the model has resulted in a statistically different climate) has been empirically determined based on a power analysis of E3SM version 1. To improve this such that baseline changes do not require a new power analysis study, a false discovery rate (FDR) correction has been implemented so that the test failure is determined solely by the number of variables analyzed and the chosen level of statistical significance. A new set of ensembles were generated to compare the power analysis method to an FDR corrected test. Additionally shorter ensembles (1-2 months) were conducted to investigate a reduction in computational cost of the test. The FDR corrected results indicate that most false positives are eliminated with this method with little added computational cost, while the shortened ensembles have not yet proved useful at detecting statistically different model climates.

PANEL: Correctness and verification across platforms
Moderator:
Brian Dobbins, NCAR

Panelists:
Ilene Carpenter, Hewlett Packard Enterprise
Dr. Ilene Carpenter is the Earth Sciences segment manager at Hewlett Packard Enterprise. She earned her Ph.D. in Physical Chemistry at UW-Madison and started her career at Cray Research. She has held several positions leading the environmental applications and benchmarking groups at SGI and Cray. In 2009, she left SGI to work as a computational scientist at US Dept. of Energy laboratories. After spending a year at ORNL, she moved to the Computational Sciences Center at NREL. She returned to Cray in 2018 to lead the Earth Sciences segment and continues that role at Hewlett Packard Enterprise.

Karsten Peters-von Gehlen, Deutsches Klimarechenzentrum GmbH (DKRZ)
Karsten has a background in meteorology (PhD and 2 PostDocs) and is currently leading the working group "Workflows" in the Data Management Department at DKRZ, Hamburg, Germany. Summarising, his interests relate to anything related to and required for efficient storage and handling of large-volume Earth System Model output data. To do this, keeping in touch with newest developments and ideas in the areas of data handling technology and infrastructure, like FAIR Digital Objects, data compression and seamless data handling across multiple storage tiers. Furthermore, Karsten actively engages in communication and outreach activities with scientists and students to guide concepts and developments along actual user needs.

Ganesh Gopalakrishnan, University of Utah
Ganesh Gopalakrishnan has been studying Semi-Formal and Formal Methods for HPC for many decades thanks to his past 25 PhD students (and 7 present). He helped run the 2017 DOE Correctness Summit (https://www.osti.gov/biblio/1470989) and recently the DOE/NSF Workshop on Correctness in Scientific

Computing (CSC'23 : https://pldi23.sigplan.org/home/csc-2023). He leads the X-Stack project called "ComPort" (https://comport.cs.washington.edu/).

Aaron Donahue, Lawrence Livermore National Laboratory
Dr. Aaron S. Donahue is a Climate Scientist at Lawrence Livermore National Lab (LLNL). His research interests are in improving the performance and accuracy of climate models and his recent work has been on the development of the Simple Cloud Resolving E3SM Atmosphere Model (SCREAM), a new E3SM atmosphere model that is capable of simulating the Earth at 3km resolution globally. Prior to working on SCREAM, Aaron's research focused on how the inter-processes coupling impacts atmospheric models. His work has focused on both process order and process coupling method. Prior to his position at LLNL, Aaron graduated from the University of Notre Dame in 2016 with a degree in Civil Engineering where his thesis work focused on the development of computational models to study the impact of ocean waves on coastal communities during extreme weather events such as hurricanes.