Singletrack - Diagnostics Working Group v0

Overview

The success of the singletrack modeling system will depend crucially on the flexibility, utility and extensible of its diagnostic capabilities. Currently the diagnostic workflow among the modeling systems and NCAR labs differs markedly in terms of data formats, software and analysis. Even within individual modeling systems the validation and analysis of model output is not fully standardized and is often ad-hoc.

The consensus is that we cannot create a complete comprehensive set of diagnostics with a generality to cover all current and potential future applications. However, we can implement a flexible framework and library of tools meant to accommodate as wide as possible set of applications. These will serve as a core set of standard data manipulation (e.g., post-processing tools) and diagnostics while retaining sufficient extensibility and flexibility for additional diagnostics.

The Singletrack diagnostics working group aims to prioritize a set of proposed requirements, document current capabilities, propose an overarching workflow, and determine how we can move forward in the short, medium and long term.

1. **Summary of requirements:** The most basic requirements for diagnostic capabilities and associated workflows target the ability to work across current modeling, and data production and processing standards. They will ensure the most efficient implementation, maintenance, testing and dissemination of current and future diagnostics capabilities.
    a. Common data format, software and workflow across applications
    b. Maintenance of core diagnostic capabilities
    c. Use of freely available software for at least the core diagnostics and newly developed capabilities
    d. Version controlled software
    e. Well defined and robust testing procedures
    f. Efficient use of computational resources for large volume, high workload diagnostic packages through software parallelism
    g. CIME compliant infrastructure where relevant
    h. Subject to governance procedures (working groups, oversight etc.)
    i. Educational diagnostic capabilities (stripped down/specific)
    j. Compatibility with diagnostics from other components (ocean, land etc.)

2. **Current capabilities:** There exists a number of standard analysis packages across the different modeling communities that are utilized, in many cases, in a quasi-operational manner

a. Weather forecast assessment (standard surface and upper-air observation station and gridded analysis comparisons including skill measure) over ranges out to a few days (e.eg. MET package)
b. Verification methods (mostly encompassed in MET package)
c. Mean climate assessment (CAM, CAM-CHEM, WACCM, WACCM-X - has no current recognized 'package')
d. Model variability assessment (CVDP, WRF)
e. Visualization Vapor/Paraview/Visit, capabilities for interactive usage and parallel
f. Nudging tendencies
g. Pre-simulations processing diagnostics (WRF)
h. Diagnostics related to cycled data assimilation (short-range forecast, e.g. 1, 3 or 6 hr analysis increments based on DA)
i. Post-processor "forward models" for comparison with radar and satellite observations

3. **In-line Diagnostics**
Diagnostic output that occurs during simulation run-time rather than at the end is increasingly important for several reasons. With the high cost (and high risk) of frontier simulations, it is crucial to ensure wherever possible the success of the integration during integration and before it completes. Targeted in-line diagnostics would allow failure to be determined at the earliest stage and wastage of compute time and people time could be reduced. In addition ever increasing standard diagnostic output at the end of runtime, makes its production less viable in an increasingly resource limited world (e.g., spinning disc), and ongoing diagnostics may need to be calculated with temporary data that is subsequently discarded. Examples of in-line diagnostics would be
    a. Max, min, mean, stdev of surface T, Td, winds, etc., e.g. daily
    b. Accumulated diagnostics for radiation (TOA and surface), energy and water budgets, etc.
    c. Possible need for online graphical outputs as a diagnostics (high-frequency animations, time series) - also allows for real-time evaluation of long runs
    d. Outputting application-driven diagnostics like diffuse/direct solar flux, radar reflectivity, hub-height winds, storm parameters (helicity, max updraft, etc.)

4. **Workflow Paradigms:** There are a number of different paradigms that may be employed for the model.
    a. Top-down: All software is rigorously tested to the same standard and has to closely follow the recommended software design and workflow.
    b. Bottom-up: We provide just the loosest of workflow guidance and interface with the model output. Then any individual software can be adapted to produce diagnostics, no matter the language.
    c. Priority-first: We require strict standards, in terms of workflow and software for a priority set of diagnostics and then enforce looser standards for secondary diagnostics, providing basic software interfaces.

5. **Software**
   It's generally expected that the main packages will be written in freely available software. Options include ncl, python, grads, nco, VAPOR, c++,java among others. This will avoid licensing options associated with popular geophysical software packages such as idl, matlab or mathematica. Parallel compute capabilities will be key to high volume, large file number output simulations. Code has to work on different machines and be easily portable. One current software package that may be a good workflow baseline to use or emulate is Pangeo. Pangeo is a software infrastructure that enables diagnostic capabilities that conform to more advanced industry-level software, employs levels of parallelism necessary for big data problems required for cutting edge science in singletrack and provides reliable workflows to ensure reliable scientific reproducibility.

6. **Computation/Portability/Multiple Simulations/Parallelism**
   One of the most important aspects to anticipate might be the future computing environments. This will be key to the high parallel processing of output data that will be crucial for frontier simulations, likely producing frequent high volume output on fine resolution grids and for long simulation run periods. Current compression strategies under develop in CISL will need to be employed, and further aggressive strategies (e.g., lossly) for lower value output.

7. **Observations/Validation/Comparison Datasets**
   In many ways this could be the most challenging aspect of the diagnostics workflow, ie., how should we incorporate the rich and vast array of observation types. Their characteristics can vary in a multitude of ways. For current CESM global longer term climate simulations many of the observational datasets used for comparison and validation are curated and documented as part of the climatedataguide.net activity. A similar infrastructure, with common metadata standards and curation across current modeling activities could be employed as part of a similar activity. This could come under the umbrella of the newly formed CISL Digital Asset Services Hub (DASH).

8. **Educational**
   An important NCAR and NSF requirement for singletrack is the support of educational tools and software to young scientists (e.g., K-12, undergraduates and graduates) for the purposes of learning about earth system science. Singletrack should consider the pros and cons of many of the current activities. This should include tools that emphasize more simplified model frameworks to learn about modeling, particularly since the singletrack as a whole (and it's codebase) could seem a daunting task for new users or young learners.

9. **Software version control**
   There has been a long standing practice of utilizing version control software for maintaining model codes (e.g., subversion and more recently git). This should certainly

be extended to diagnostic data processing and analysis software. This should enable a more unified support methodology among model and diagnostic code, with community code review and maintenance of code quality through code review practices and coding guidelines, crucially including bug reporting.

10. **Documentation**

    In a similar manner as documentation methodologies for running the singletrack model, there will be a requirement for providing documentation on the numerous diagnostics capabilities. Documentation should also be a priority for new users who either wish to include a pre-existing diagnostic capability as part of singletrack, but also for providing software (based on recommendations from this group) of a new diagnostic analysis.

11. **Diagnostic Curation**

    Given the ever increasing requirements in scientific journals for reproducibility of results and workflow traceability an automated capability for the curation of diagnostics and diagnostic software is required. This should include the creation of, for example, a doi that would provide curation of the whole workflow in a single location (not just diagnostics) with end-to-end information regarding model code bases, parameter choices, value-added diagnostic output and diagnostic scripts. Recent efforts for CESM that employ automated run submission and database capabilities could provide a primer for workflows that enable this type of curation.

Next Steps

- Provide an inventory of current model diagnostics packages and capabilities on a two-tiered basis: priority standard packages and community contributions
- Interface with the infrastructure working group to ensure that diagnostic requirements are considered at the design and implementation phases of new singletrack infrastructure
- Determine relevant diagnostics capabilities to be implemented as part of the singletrack frontier science applications, and their software framework
- Implement potential diagnostic workflow capabilities with a pilot project to interface the MET package with CESM unified component (atm, land, ocean, ice) diagnostic workflow
- Request community input on singletrack diagnostic expectations, including the extent to which PI or community diagnostics should be supported and what the software constraints (if any) should be encouraged or enforced. This could take the form of and email or web survey.