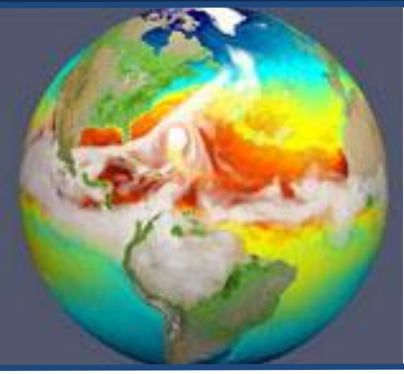
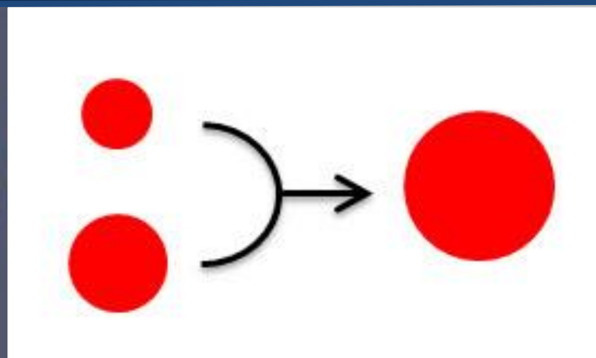
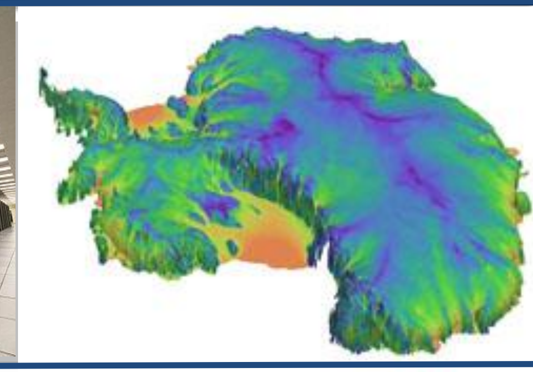
Exceptional service in the national interest

Sandia National Laboratories







Verification and Testing Infrastructure and Demonstrations

Irina Tezaur[1], Hui Wan[2], Andreas Wilke[3], Dick Easter[2], Jian Sun[2], Jason Sarich[3], Kai Zhang[2], Luke van Roekel[4], LeAnn Conlon[4], Jamil Gafur[4], Rachel Scanza[2], Lance Rayborn[2], Richard Easter[2], Vince Larson[5]

[1] SNL, [2] PNNL, [3] ANL, [4] LANL, [5] U Wisconsin

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2019-13937C

Motivation

This talk is on the **tools/workflow** created under the **CMDV-SM verification subtask**, which aims to create/enable a culture of **testing/verification** in the E3SM.

- **Purpose of verification**: instill ***confidence*** in numerical simulations
  - Demonstrate that simulations represent the intended mathematical model rather than ***numerical artifacts*** or ***coding bugs***
  - Test against ***analytic*** or ***trusted*** solutions
  - Confirm ***convergence*** of algorithms at theoretical rates
  - Detect changes over time (***regression***)
- **Status quo in (much of) E3SM**
  - Verification & validation not sufficiently ***distinguished***
  - Mostly focus on ***validation*** (matching observations)
  - Most developers do ***some verification***
  - Usually ***limited coverage*** and not formalized
  - Usually not isolated or localized – always ***case-based***
  - ***Not preserved*** to re-confirm correctness after modifications

Our efforts under CMDV-SM

Sandia National Laboratories

- Work with model developers to ***define/formulate*** appropriate ***verification (unit and unit-like) tests*** for their models

- ***Create workflow*** and ***corresponding infrastructure*** for performing verification and presenting verification results
  - cmdv-test-runner: python-based tool for running verification tests.
  - cron/Jenkins: tools for automating running of tests nightly, weekly, etc.
  - CDash: web-based software server for displaying/storing testing results.
  - Jupyter (notebooks): tool for writing documentation for verification tests and post-processing results.

- Create ***concrete demonstrations*** of the above testing/ documentation infrastructure on MAM and CLUBB, Ocean Mixing, etc.

Jenkins

jupyter

Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
  - Requires ***interest/involvement*** from ***component developers.***
  - Requires knowledge of ***what is in the code.***
  - Requires understanding of ***mathematical concepts***, e.g. convergence.

Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
    - Requires ***interest/involvement*** from ***component developers.***
    - Requires knowledge of ***what is in the code.***
    - Requires understanding of ***mathematical concepts***, e.g. convergence.
- Ideally tests would be written as code is ***being developed*** (e.g., SCREAM)

Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
  - Requires ***interest/involvement*** from ***component developers.***
  - Requires knowledge of ***what is in the code.***
  - Requires understanding of ***mathematical concepts***, e.g. convergence.
- Ideally tests would be written as code is ***being developed*** (e.g., SCREAM)
- Tests should be ***small*** (subroutine, kernel, small set of kernels).

Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
  - Requires ***interest/involvement*** from ***component developers.***
  - Requires knowledge of ***what is in the code.***
  - Requires understanding of ***mathematical concepts***, e.g. convergence.
- Ideally tests would be written as code is ***being developed*** (e.g., SCREAM)
- Tests should be ***small*** (subroutine, kernel, small set of kernels).
- Examples of what **IS** a verification test:
  - Check ***mathematical properties*** (convergence rate, divergence free, conservation of mass, etc.), compare to theory.

Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
  - Requires ***interest/involvement*** from ***component developers.***
  - Requires knowledge of ***what is in the code.***
  - Requires understanding of ***mathematical concepts***, e.g. convergence.
- Ideally tests would be written as code is ***being developed*** (e.g., SCREAM)
- Tests should be ***small*** (subroutine, kernel, small set of kernels).
- Examples of what **IS** a verification test:
  - Check ***mathematical properties*** (convergence rate, divergence free, conservation of mass, etc.), compare to theory.
- Examples of what **IS NOT** a verification test:
  - Perform a run and compare to ***observational data*** (this is ***validation***!).
  - ***Arbitrary parameter tunings*** to match expected data/solution.

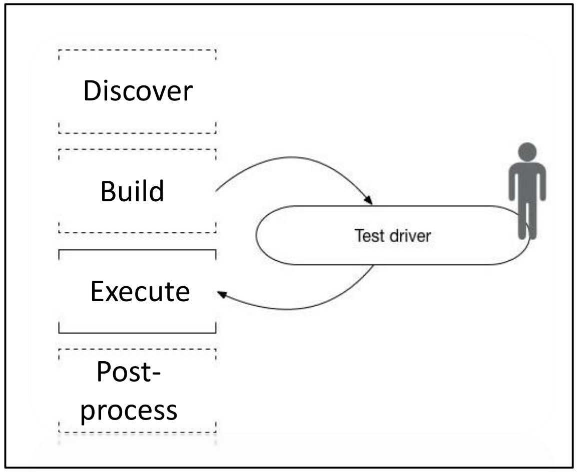
Verification test formulation

- Formulating a verification test can present a number of ***challenges***:
  - Requires ***interest/involvement*** from ***component developers.***
  - Requires knowledge of ***what is in the code.***
  - Requires understanding of ***mathematical concepts***, e.g. convergence.
- Ideally tests would be written as code is ***being developed*** (e.g., SCREAM)
- Tests should be ***small*** (subroutine, kernel, small set of kernels).
- Examples of what **IS** a verification test:
  - Check ***mathematical properties*** (convergence rate, divergence free, conservation of mass, etc.), compare to theory.
- Examples of what **IS NOT** a verification test:
  - Perform a run and compare to ***observational data*** (this is ***validation***!).
  - ***Arbitrary parameter tunings*** to match expected data/solution.
- Once test is formulated, ***test driver*** must be created, which can be done by ***hand*** or using available ***tools***, e.g. kgen.

Running the tests: cmdv-test-runner

Sandia National Laboratories

- Python-based tool to discover, build, run, post-process verification tests
  - ➢ Discovers tests in current directory
  - ➢ Compiles and runs tests according to workflow file
  - ➢ Reports results



One step workflow executing one command:
./cmdv-test-runner --test mam_box.verification.test.yaml

```yaml
cmdvVersion: v1.0
class: Test-Workflow

label: MAM BOX verification config

steps:
  build:
    run:
      baseCommand: [
      'cp ../src/cambox_config.cpp.in.gfortran ../src/cambox_config.cpp.in' , ' ; ' ,
      'cp ../src/cambox_config.make.in.gfortran ../src/cambox_config.make.in' , ' ; ' ,
      'cp ../src/coag_al_driver.F90 ../src/driver.F90' , ' ; ' ,
      'make' , '-C' , '../src'  , " ; " ,
      'echo compilation success' ]
    in: {}
    out: {}

  run:
    run:
      baseCommand: [ '../src/dd.x' ]
    in: {}
    out:
      rate:
        type: File
        glob: coag_rate.out

  postprocessing:
    run: # workflow or tool or baseCommand
      baseCommand: [ 'nb2html' , '../src/Coagulation.ipynb ' ]
    in: {}
    out: {}
```

Sample input yaml file:
mam_box.verification.test.yaml

➢ Can be executed within Docker container.

Documentation and examples can be found here:

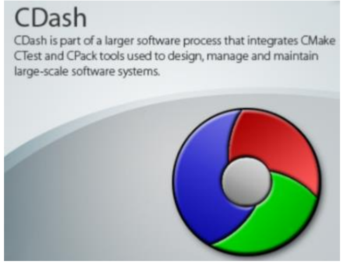
https://github.com/E3SM-Project/CMDV-testing/wiki

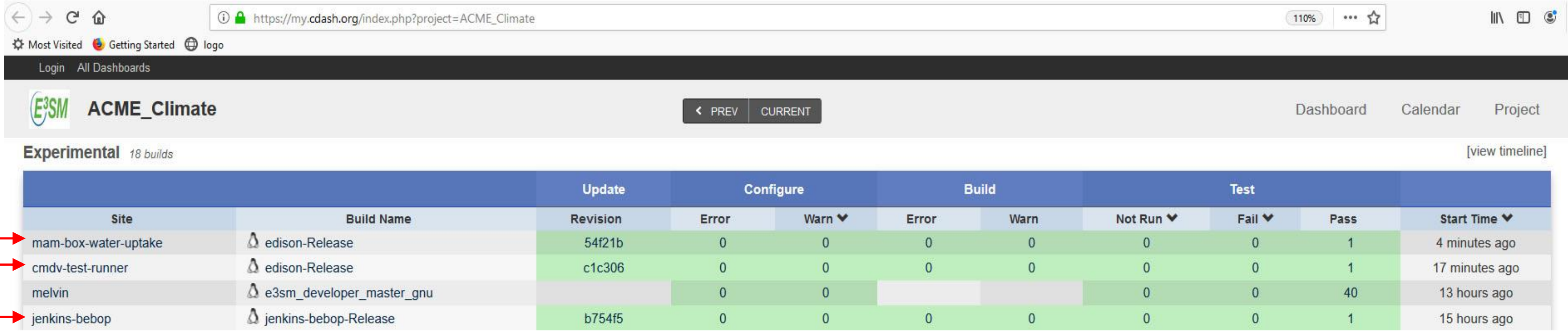
Automation of test execution (cron/Jenkins) and results archival (CDash)

- Execution of tests can be ***automated*** using cron or Jenkins.

- We have created ***cron/Jenkins jobs*** on NERSC, ANL, SNL machines that run cmdv-test-runner (self-tests, MAM water uptake tests) nightly and post results to the [ACME_Climate CDash site.](https://my.cdash.org/index.php?project=ACME_Climate)

  - **<u>Future work</u>**: extend workflow to automatically run and post-to-CDash additional test results.





		Update	Configure		Build		Test			
Site	Build Name	Revision	Error	Warn	Error	Warn	Not Run	Fail	Pass	Start Time
mam-box-water-uptake	edison-Release	54f21b	0	0	0	0	0	0	1	4 minutes ago
cmdv-test-runner	edison-Release	c1c306	0	0	0	0	0	0	1	17 minutes ago
melvin	e3sm_developer_master_gnu		0	0			0	0	40	13 hours ago
jenkins-bebop	jenkins-bebop-Release	b754f5	0	0	0	0	0	0	1	15 hours ago

Automation of test execution (cron/Jenkins) and results archival (CDash)

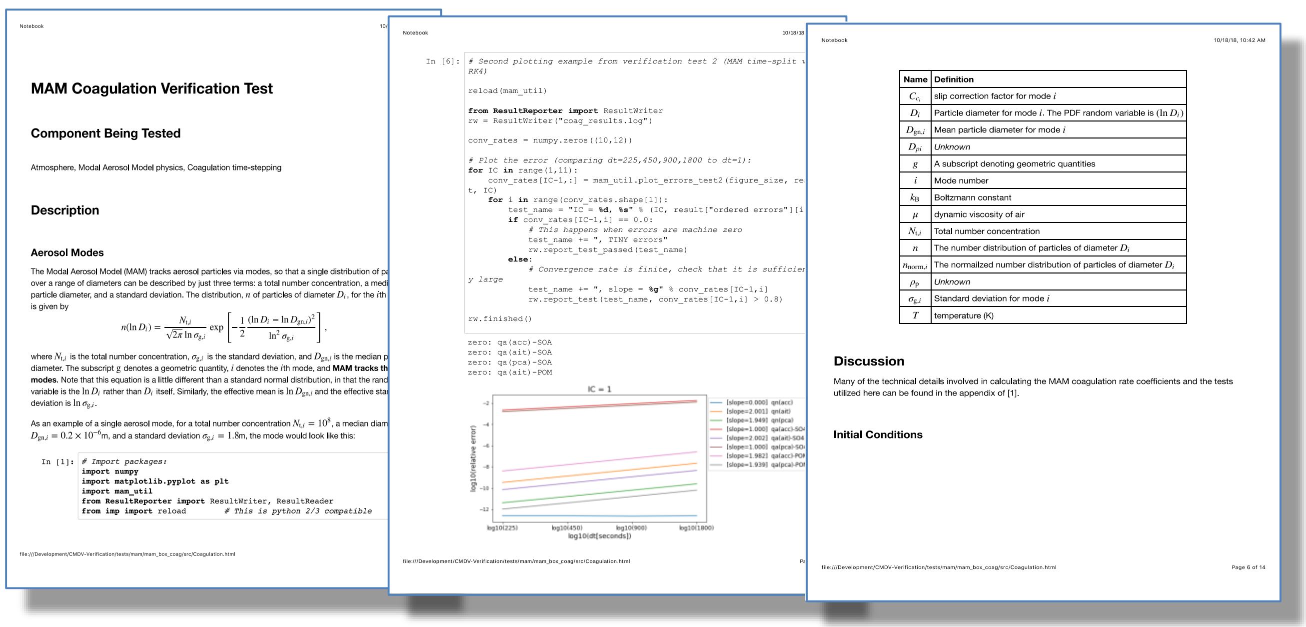
https://my.cdash.org/index.php?project=ACME_Climate

Documentation/post-processing: Jupyter notebooks

- Common tool for ***writing documentation*** and ***post-processing/re-generating*** verification results.

- ***Documentation*** to new/existing users of various E3SM components *and* ***template*** for writing verification/unit tests.

- The following can be ***embedded*** within a Jupyter notebook:
  - ➢ LaTex
  - ➢ Python
  - ➢ Matlab
  - ➢ Julia
  - ➢ …

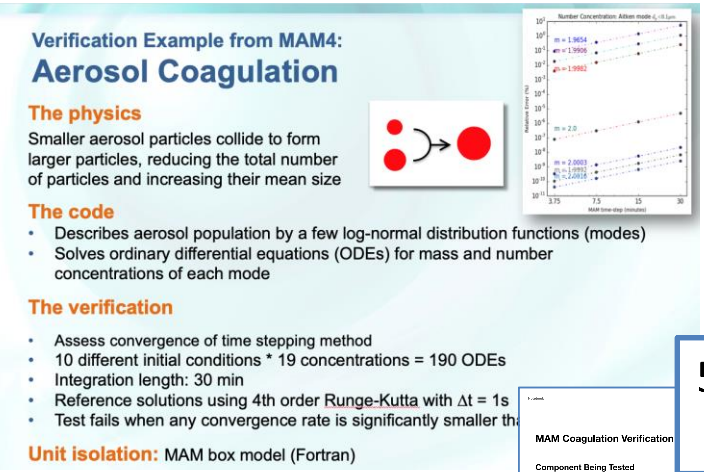


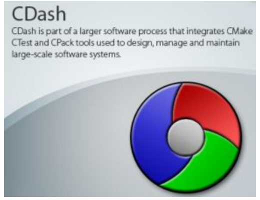
Documentation/post-processing: Jupyter notebooks

- HTML versions of Jupyter notebooks are linked from ***E3SM github.io*** page: https://e3sm-project.github.io/CMDV-testing/
- We have Jupyter notebooks for the ***following components***:
  - *HOMME:* Shallow Water TC1
  - *Albany Land-Ice (ALI):* FO Stokes MMS TC
  - *MPAS-Ocean*: comparison b/w cmix and PALM LES runs
  - *Modal Aerosol Model (MAM):* water uptake, condensation, coagulation
  - *Cloud Layers Unified By Binormals (CLUBB):* clipping in atm. physics
  - *Misc Unit Tests:* subroutines in global_verif_summary.F90
- Jupyter source code can be found in the ***CMDV-Verification repo*** https://github.com/E3SM-Project/CMDV-verification
  - ***Binder*** support coming soon.
- **Future work**: add on-the-fly creation of (some) of the Jupyter notebooks as a part of automated verification workflow.

End-to-end verification workflow

1. Formulate test
2. Identify target code
3. Create test driver

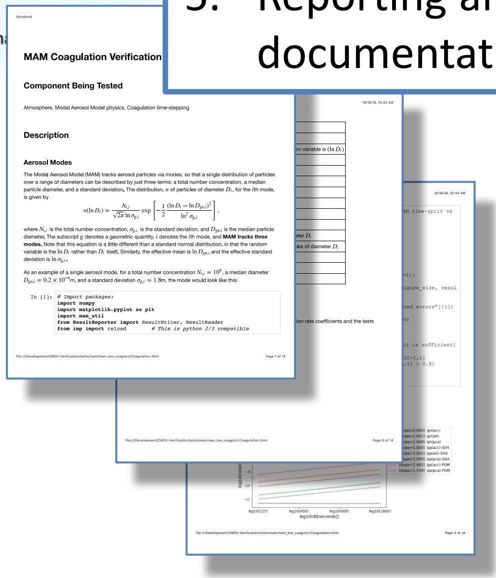




4. Automate setup and execution

```
#! /bin/csh
module load intel
make clean
cp coag_b1_driver.F90 driver.F90
make
echo compilation finished...
echo running code
#cd test_output  ; rm dd.x  ; ln -s ../dd.x .
./dd.x
echo test compiled here
#cd ../ ; cp test_output/coag_delNum_delMass.out .
echo main output is coag_delNum_delMass.out
echo now running python script to generate graphics and
R2. /share/apps/python/anaconda3/bin/python
coag_dq_da.py
echo convergence diagrams saved.
```

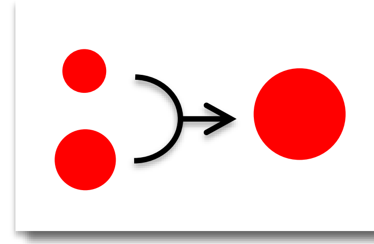
5. Reporting and documentation



Verification example from MAM: Aerosol Coagulation

R. Scanza, R. Easter (PNNL)

https://e3sm-project.github.io/CMDV-testing/Coagulation.html



The physics

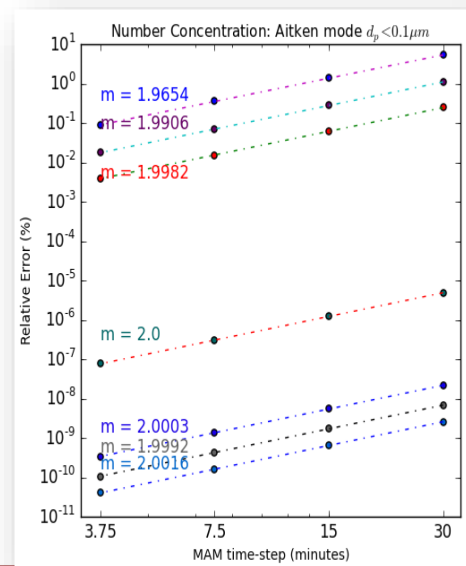
- Smaller aerosol particles collide to form larger particles, reducing the total number of particles and increasing their mean size

The code

- Describes aerosol population by a few log-normal distribution functions (modes)
- Solves ordinary differential equations (ODEs) for mass and number concentrations of each mode

The verification

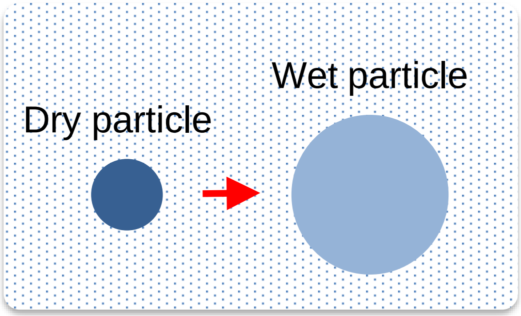
- Assess convergence of time stepping method
- 10 different initial conditions * 19 concentrations = 190 ODEs
- Integration length: 30 min
- Reference solutions using 4th order Runge-Kutta with Δt = 1s
- Test fails when convergence rates is significantly < expected



Unit isolation: MAM box model (Fortran)

Verification example from MAM: Aerosol Water Update

J. Sun, K. Zhang, H. Wan (PNNL)



The physics

- Aerosol particles absorb water and grow in size

The code

- Assumes quartic relationship between air humidity and particle wet radius
- Calculates wet radius (a root of quartic function) using analytical expression
- Expects 1 real-and-physical root

The verification

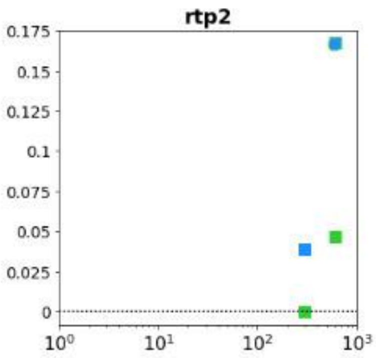
- Verifies the analytical expression and its implementation
- Calculates wet radius for 576 different combinations of humidity, particle composition, and dry radius
- Reference solutions using root-finding by bisection
- Test fails if relative difference between any pair of MAM and reference solutions exceeds 1E-15 (machine precision)

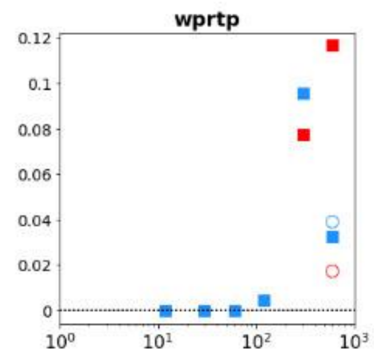
Unit isolation: MAM box model (Fortran)

https://e3sm-project.github.io/CMDV-testing/wateruptake/verification.html

Verification example from CLUBB: Clipping in Stand-Alone Model

L. Rayborn (PNNL)





The physics

- CLUBB (Cloud Layers Unified By Binormals) is a parameterization of clouds and turbulence for the representation of cloud macrophysics, shallow convection, and turbulence.

The code

- Solution “clipping” is introduced to avoid non-physical quantities (e.g., negative densities).
- Clipping is in general non-conservative, but there are conservative variants (e.g., hole filling)

The verification

- Quantify magnitude of clipping terms in CLUBB's 13 prognostic eqns. using BOMEX TC.
- Evaluate various clipping schemes in CLUBB.
- Magnitude of clipping term for each scheme is compared to max physical term in each budget; test fails if magnitude > threshold.

Unit isolation: clipping BOMEX unit test.

https://e3sm-project.github.io/CMDV-testing/CLUBB_clipping_test.html

Verification example from MPAS-O: K-Profile Parameterization

L. Van Roekel, L. Conlon, J. Gafur (LANL)

Sandia National Laboratories

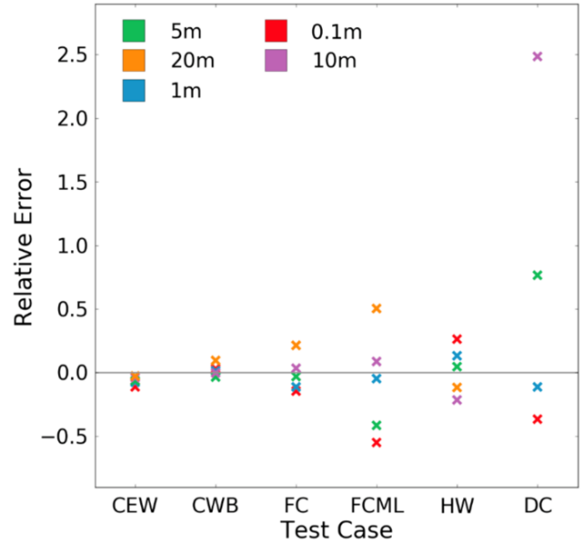
https://e3sm-project.github.io/CMDV-testing/palm_cvmix_compare.html

The physics

- The K-Profile Parameterization (KPP) represents small scale (below the grid scale) vertical turbulent fluxes of heat, salt, and momentum in the ocean.

The code

- KPP does not contain any prognostic equations, but utilizes diagnostic equations and scaling relationships to parameterize turbulent fluxes.
- KPP is known to not exhibit convergence with vertical resolution, but is robust to time step variation.
- No known analytic solutions for comparison in most cases.



Relative error he across test cases and resolution

The verification

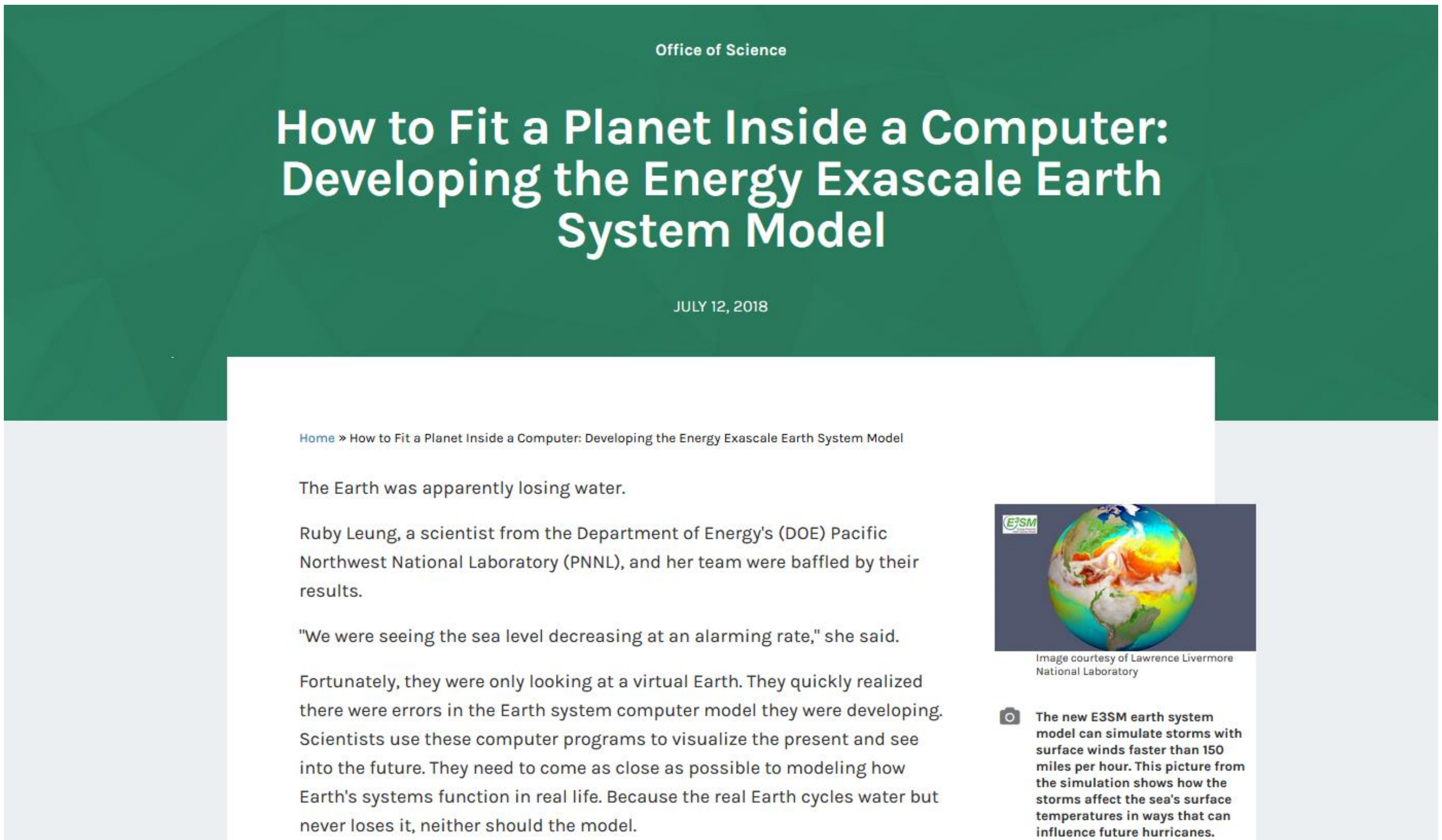
- Utilize a large eddy simulation as a baseline to verify the temperature tendency and entrainment depth (figure) returned by KPP, critical quantities for the ocean simulation.

Unit isolation: CVMix module and MPAS-Ocean vertical mixing interface.

Success story #1

*A **critical issue** with water conservation in E3SM was **uncovered**!*

https://www.energy.gov/science/articles/how-fit-planet-inside-computer-developing-energy-exascale-earth



Office of Science

How to Fit a Planet Inside a Computer: Developing the Energy Exascale Earth System Model

JULY 12, 2018

Home » How to Fit a Planet Inside a Computer: Developing the Energy Exascale Earth System Model

The Earth was apparently losing water.

Ruby Leung, a scientist from the Department of Energy's (DOE) Pacific Northwest National Laboratory (PNNL), and her team were baffled by their results.

"We were seeing the sea level decreasing at an alarming rate," she said.

Fortunately, they were only looking at a virtual Earth. They quickly realized there were errors in the Earth system computer model they were developing. Scientists use these computer programs to visualize the present and see into the future. They need to come as close as possible to modeling how Earth's systems function in real life. Because the real Earth cycles water but never loses it, neither should the model.

Image courtesy of Lawrence Livermore National Laboratory

The new E3SM earth system model can simulate storms with surface winds faster than 150 miles per hour. This picture from the simulation shows how the storms affect the sea's surface temperatures in ways that can influence future hurricanes.

Success story #2

Sandia National Laboratories

Issues *in the implementation and use of* ***nudging**** *in E3SM were* ***uncovered!***

Caution:
Potential Misuse of Nudging in the Atmosphere Model of E3SMv0 and v1

Kai Zhang, Jian Sun, Hui Wan
PNNL
January, 2019

Target audience of the slides

E3SM developers and users who used nudging in CAM5 and are using similar scripts and forcing data to conduct nudged atmospheric simulations with E3SMv0 and v1

A more ***flexible implementation*** *was* ***developed.***

Impact of nudging strategy on the climate representativeness and hindcast skill of constrained EAMv1 simulations

Jian Sun[1], Kai Zhang[1], Hui Wan[1], Po-Lun Ma[1], Qi Tang[2], Shixuan Zhang[1]

[1]Pacific Northwest National Laboratory, Richland, WA, USA
[2]Lawrence Livermore National Laboratory, Livermore, CA, USA

Journal article accepted by JAMES

*Data assimilation technique used in sensitivity studies/validation of EAM simulations.

Ongoing & future work

Expand testing framework:

- ***Evaluate*** and ***integrate*** or support for kgen, ctest
- Explore using ***E3SM configuration/software environment*** to setup/tests on supported machines

Outreach and documentation:

- Expand testing ***documentation***, hold cmdv-test-runner ***tutorial*** (TBD).
- Deploy testing ***workflow on other projects***, e.g.,
  - ❑ RRTMCP (radiative transport)
  - ❑ Land model (FATES)
  - ❑ DEMSI
  - ❑ SCREAM
  - ❑ EAGLES
  - ❑ Other components?

Recommendations for future success:

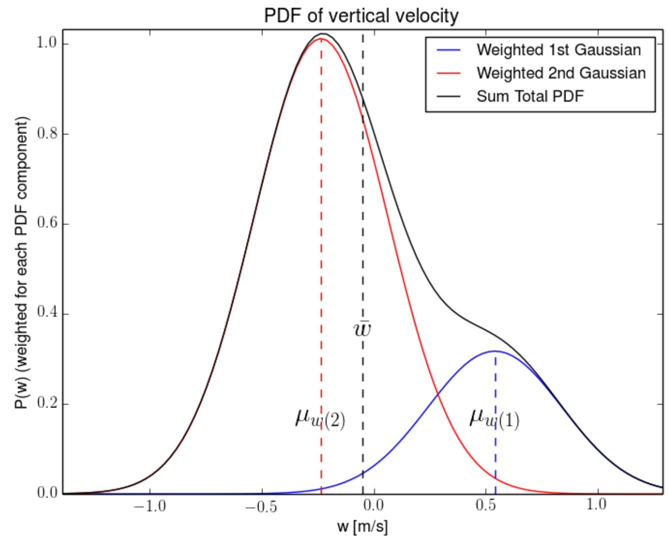
- Requiring ***common testing tools*** (e.g., ctest) would simplify workflow.
- ***Documentation & verification*** should be done as code is developed.
- ***Liaison*** from each targeted component/project is critical to creating meaningful tests!

Starting CY20: monthly ***Verification Interest Group*** concall (POC: Hui Wan)

Backup Slides

Verification example from CLUBB: Turbulence Closure

V. Larsen (U Wisconsin)



The physics

- CLUBB's subgrid PDF is used to diagnose cloud fraction within a grid box.
- CLUBB's PDF shape is a double Gaussian (sum of 2 Gaussian components).

The code

- Parameterizes the mean and variance of *each Gaussian component* in terms of the mean and variance over the *entire grid box*.

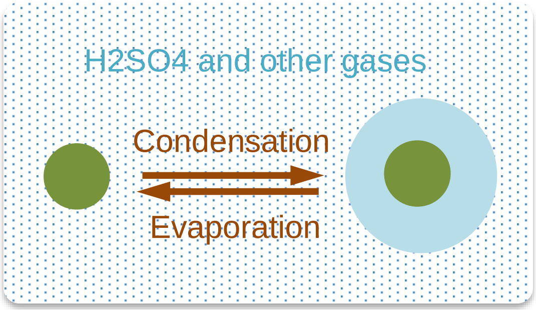
The verification

- Tests whether the aforementioned algebra is correct.
- More specifically, for a variety of selected inputs, the test uses CLUBB code to parameterize the component means and variances, and then checks whether the original grid mean and variance can be recovered.

Unit isolation: CLUBB PDF subroutine (Fortran).

Verification example from MAM: Condensation and evaporation

J. Sun, R. Easter, K. Zhang, H. Wan (PNNL)



The physics

- Gas phase chemical species condense to or evaporate from aerosol particles and change their sizes

The code

- Solves ordinary differential equations (ODEs) for mass concentrations of related gases and aerosol species

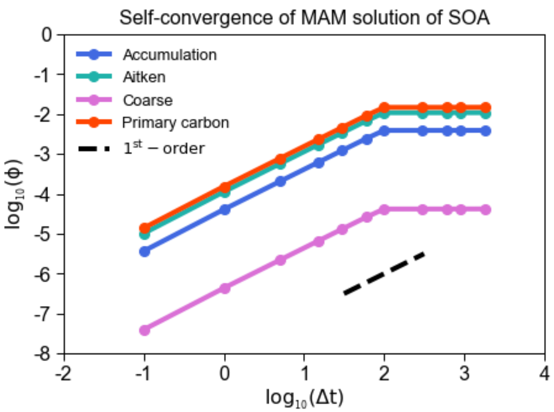
The verification

Multiple tests for the numerical methods

- Gauss-Hermite Quadrature for Mass Transfer Coefficients
- Time integration of ODEs

Additional tests for code implementation

- E.g., Impact of rounded or non-standard values for various parameters



See also poster by Sun et al.

Unit isolation: MAM box model (Fortran)

https://e3sm-project.github.io/CMDV-testing/condensation/gauss_hermite_accuracy/verification.html