Introduction to Noah-MP component model in NOAA’s UFS

Ufuk Turuncoglu and Michael Barlage

25 May 2023, Noah-MP Workshop & Tutorial

This work is supported by the NOAA Joint Technology Transfer Initiative (JTTI)
NA21OAR4590167: Advancing Land Modeling Infrastructure in the UFS for Hierarchical Model Development

Project Goals and Objectives

- The project will deliver **new land modeling infrastructure** software that enables running component-based land surface models (LSMs) within the UFS.

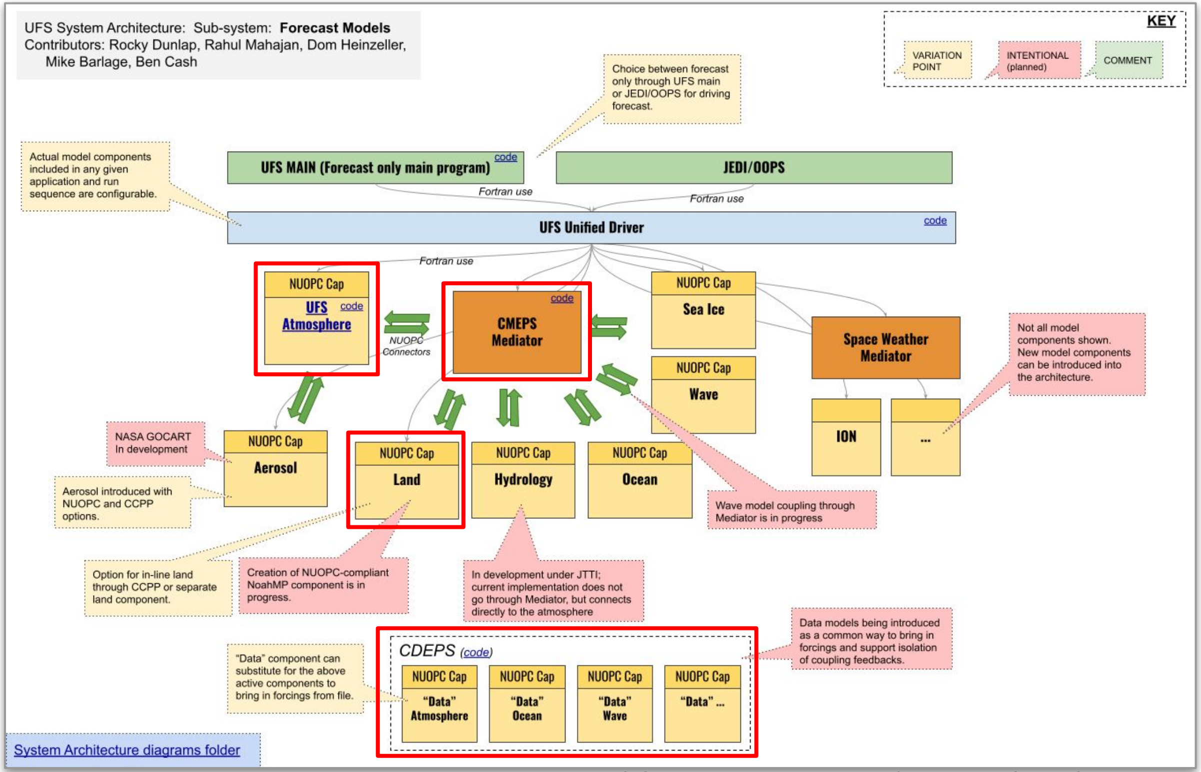
Project Title: "Advancing Land Modeling Infrastructure in the UFS for Hierarchical Model Development" **PIs:** Ufuk Turuncoglu (previous PIs Rocky Dunlap and Mariana Vertenstein) **EMC Collaborator:** Mike Barlage Funded by FY21 Joint Technology Transfer Initiative (JTTI) Program

Obj 1	The Noah-MP parameterization will be "wrapped" as a NUOPC-based model component, and will be demonstrated to run as an independent component under the UFS Driver in different configurations.
Obj 2	The Noah-MP component will be configured to run "one-way coupled" with atmospheric forcings provided by the Community Data Models for Earth Prediction Systems (CDEPS).
Obj 3	The Noah-MP component will be configured to run "two-way coupled" to the active atmosphere model through the Community Mediator for Earth Prediction Systems (CMEPS).

- **All developments will be integrated into the ufs-weather-model community repository** and made available for use by the UFS community of developers.

NOAA's Unified Forecast System (UFS)

- The UFS is a community-based, coupled, comprehensive Earth modeling system.



Forecast Models diagram developed by UFS System Architecture and Infrastructure Cross-Cutting Team.

Unified modeling requires a modular and flexible approach to building complex coupled systems.

Uses ESMF library for coupling.

Example UFS configurations:

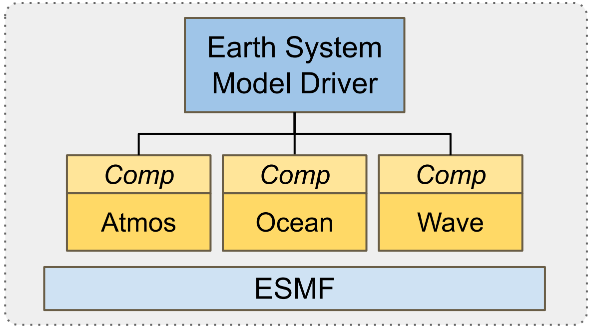
UFS GFS:
FV3ATM-MOM6-CICE6-WW3-GOCART-CMEPS

UFS Hurricane (HAFS):
FV3ATM-HYCOM-WW3-CMEPS

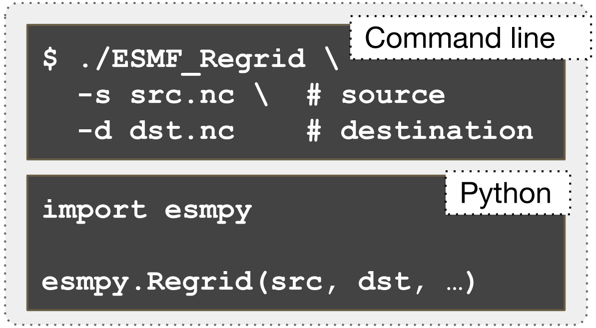
UFS RRFS:
FV3ATM

Introduction to ESMF

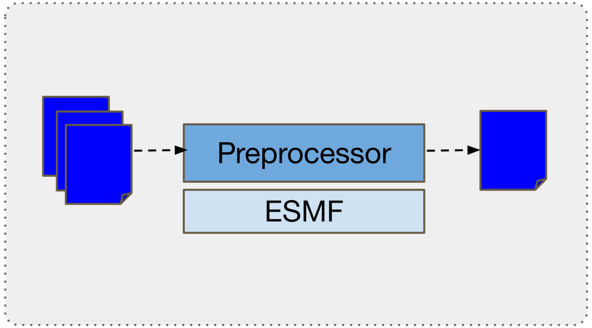
- The **E**arth **S**ystem **M**odeling **F**ramework (ESMF) is parallelizable high-performance software infrastructure used in coupled Earth science applications.
- It is open source software: https://github.com/esmf-org/esmf
- Different ways to use ESMF?



Coupling infrastructure
in a modeling system
(includes the NUOPC Layer)



***Offline tool for grid remapping** and*
interpolation weight generation
(command line and Python)



***Library** used to construct custom*
*tools, such as **preprocessor***
coprocessing** and **postprocessor

Introduction to NUOPC Layer

- The **N**ational **U**nified **O**perational **P**rediction **C**apability (NUOPC) is a software layer on top of ESMF that provides “out of the box” capabilities for constructing coupled models.
- Noah-MP under UFS Weather Model is one of the such components

Driver

Driver

A Driver has one or more child components and is responsible for coordinating their initialization sequence and driving them through a customizable run sequence.

Connector

Connector

Connector

A Connector performs standard communication operations, in parallel, between other components, such as grid remapping and redistribution of data. Connectors have a built-in field matching algorithm based on standard names.

Model

Model

A Model “cap” wraps a geophysical model code with standard initialization and run methods so it can be plugged into a Driver.

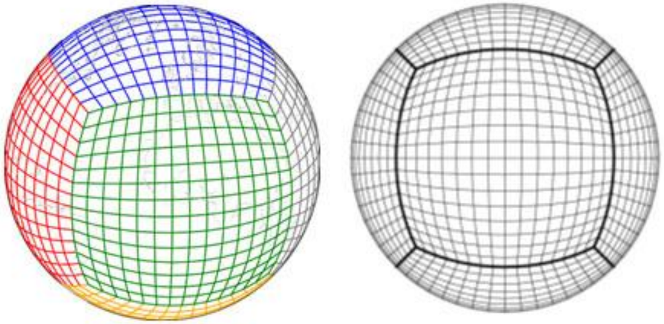
Mediator

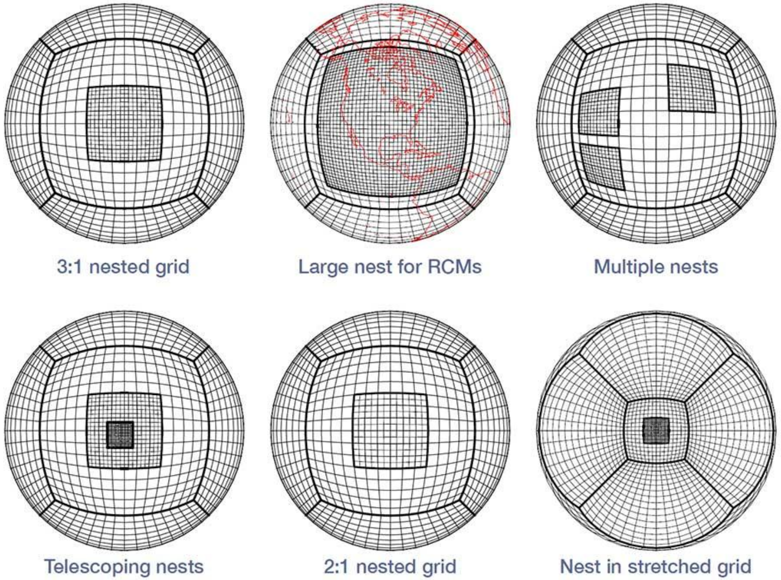
Mediator

A Mediator contains custom coupling code such as flux calculations, accumulation/averaging, and merging of fields among several components.

FV3ATM: The atmospheric component of the NOAA's UFS

- Sub-components of FV3ATM
  - FV3: Finite-Volume Cubed-Sphere Dynamical Core
  - The Common Community Physics Package (CCPP)
    - Framework and Physics
  - UFS stochastic physics
  - I/O and coupler code (ESMF)

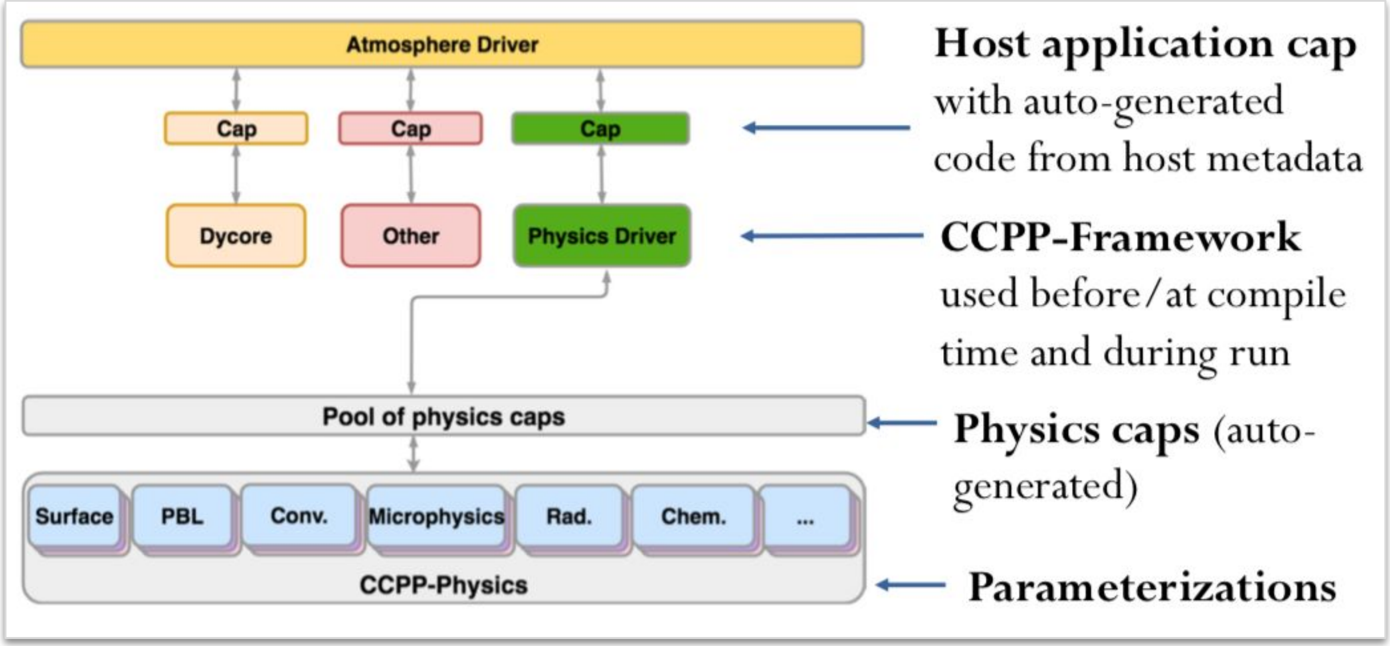




https://www.gfdl.noaa.gov/fv3/fv3-grids/

CCPP: The Common Community Physics Package

- There are two distinct parts to the CCPP: a library of physical parameterizations (CCPP-Physics) that conforms to selected standards and an infrastructure (CCPP-Framework) that enables connecting the physics to a host model.



Noah-MP ESMF/NUOPC “cap”

- Since Noah-MP model does not include
  - I/O capability for reading static information, writing model history/restart files
  - namelist handling (configuring model) etc.
- It requires to have additional layer (sometimes called as “driver”) to force/drive the model such as *HRLDAS*, *UFS-Land-Driver* and *CCPP driver*
- As a part of this project, we created another application layer using ESMF/NUOPC library to interact with other model components. This layer is commonly called as “cap”.
- The “cap” mainly provides capability for I/O (through use of ESMF multi-tile I/O support), namelist handling and interaction with other model components through the use of ESMF import and export states

Noah-MP ESMF/NUOPC “cap” ...

- The Noah-MP model with ESMF/NUOPC “cap” can be found in following repository
  - Main repository: https://github.com/NOAA-EMC/noahmp (*develop*)
  - The development fork: https://github.com/esmf-org/noahmp (*feature/nuopc_cap*)
    - Includes ERA5 forced configuration and fully coupled FV3+LND
  - Documentation: https://noaa-emc.github.io/noahmp/
- Limitations
  - Only tested with global cubed-sphere grid but there is no any restriction to extend implementation to cover regional cases and regular lat-lon grid
  - It uses FV3/CCPP version of the model. There is a plan to extend the implementation to use restructured model code at authoritative repository.

Code Structure

```
.
├── cmake
│   ├── FindESMF.cmake
│   └── FindNetCDF.cmake
├── CMakeLists.txt
```

Build system for standalone model and creates *libnoahmp.a*.

```
├── drivers
│   ├── ccpp
│   │   ├── funcphys.f90
│   │   ├── machine.F
│   │   ├── namelist_soilveg.f
│   │   ├── noahmpdrv.F90
│   │   ├── noahmpdrv.meta
│   │   ├── noahmp_tables.f90
│   │   ├── physcons.F90
│   │   ├── set_soilveg.f
│   │   ├── sfc_diff.f
│   │   └── update.sh
```

CCPP Physics related files.

```
│   ├── nuopc
│   │   ├── docs
│   │   │   ├── Makefile
│   │   │   ├── README.md
│   │   │   └── source
│   │   │       ├── BuildingAndRunning.rst
│   │   │       ├── ComponentTesting.rst
│   │   │       ├── conf.py
│   │   │       ├── DriverAndCoupling.rst
│   │   │       ├── index.rst
│   │   │       └── references.bib
│   │   ├── lnd_comp_domain.F90
│   │   ├── lnd_comp_driver.F90
│   │   ├── lnd_comp_import_export.F90
│   │   ├── lnd_comp_io.F90
│   │   ├── lnd_comp_kind.F90
│   │   ├── lnd_comp_nuopc.F90
│   │   ├── lnd_comp_shr.F90
│   │   └── lnd_comp_types.F90
```

ESMF/NUOPC cap files and documentation source (Sphinx)

```
│   └── wrf
│       └── module_sf_noahmpdrv.F
```

```
├── .git
├── .github
│   └── workflows
│       ├── data
│       │   ├── C96.initial.tile1.nc
│       │   ├── C96.initial.tile2.nc
│       │   ├── C96.initial.tile3.nc
│       │   ├── C96.initial.tile4.nc
│       │   ├── C96.initial.tile5.nc
│       │   ├── C96.initial.tile6.nc
│       │   ├── oro_data.tile1.nc
│       │   ├── oro_data.tile2.nc
│       │   ├── oro_data.tile3.nc
│       │   ├── oro_data.tile4.nc
│       │   ├── oro_data.tile5.nc
│       │   └── oro_data.tile6.nc
│       ├── datm_noahmp.yaml
│       └── tests
│           ├── test_datm_lnd
│           │   ├── datm.yaml
│           │   └── lnd.yaml
│           └── test_datm_lnd.yaml
```

Files for testing (DATM+LND) with GitHub action - [nuopc-comp-testing]

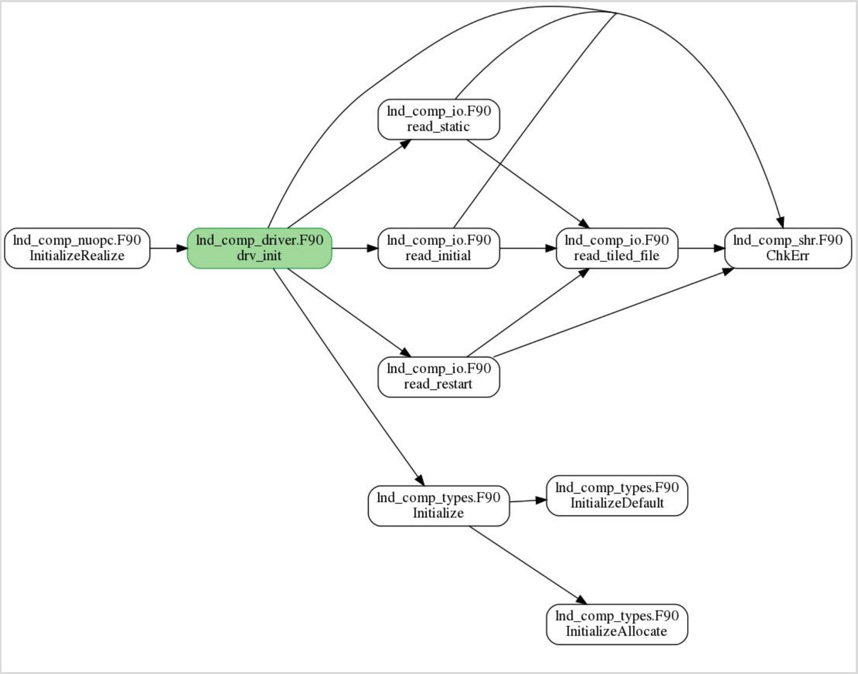
```
├── parameters
│   ├── GENPARM.TBL
│   ├── MPTABLE.TBL
│   └── SOILPARM.TBL
```

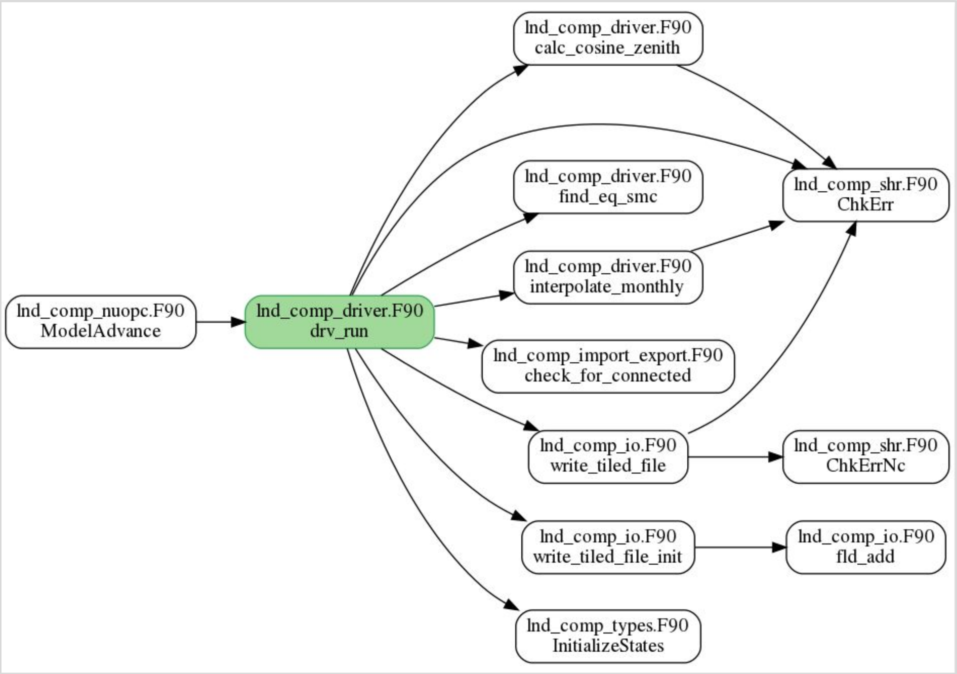
Parameter tables

```
├── README.md
└── src
    ├── module_sf_noahmp_glacier.F90
    └── module_sf_noahmplsm.F90
```

Model code

Details of ESMF/NUOPC “cap”





In the initialization phase, it reads initial conditions, static information and restart files (if it is required) from multi-tile files.

In the run phase, it prepares required data structures, calls *noahmpdrv()* - not shown in the diagram - and write output.

Import/Export Fields - DATM/Noah-MP

- Noah-MP is forced with data atmosphere: GSWP3 or ERA5

Import

Std. Name	Alias	Unit	Receiver
Sa_z	inst_height_lowest	m	lnd
Sa_tbot	inst_temp_height_lowest	K	lnd
Sa_pslv	inst_pres_height_surface	Pa	lnd
Sa_pbot	inst_pres_height_lowest	Pa	lnd
Sa_shum	inst_spec_humid_height_lowest	kg kg-1	lnd
Sa_u	inst_zonal_wind_height_lowest	m s-1	lnd
Sa_v	inst_merid_wind_height_lowest	m s-1	lnd
Sa_wspd	inst_wind_speed_height_lowest	m s-1	lnd
Faxa_swdn	mean_down_sw_flx	W m-2	lnd
Faxa_lwdn	mean_down_lw_flx	W m-2	lnd
Faxa_rain	mean_prec_rate	kg m-2 s-1	lnd

- There is no any limitation to use other data atmosphere modes supported by CDEPS. Some other data modes are;
  - CFSR
  - GFS
  - CFS …

- For the coupled model, the interpolation capability is mainly provided by ESMF library.
  - The default interpolation type is defined in CMEPS mediator. In this case, *1st Order Conservative interpolation* is used for active model coupling and *1st Order Conservative (with nearest-neighbour)* for coupling with data components.

Import/Export Fields - FV3/Noah-MP fully-coupled

- Note: side-by-side configuration does not send fields to FV3 (cpllnd2atm = .false.)

Import

Std. Name	Alias	Unit	Receiver
Sa_z	inst_height_lowest	m	lnd
Sa_ta	inst_temp_height_lowest_from_phys	K	lnd
Sa_pslv	inst_pres_height_surface	Pa	lnd
Sa_qa	inst_spec_humid_height_lowest_from_phys	kg kg-1	lnd
Sa_ua	inst_zonal_wind_height_lowest_from_phys	m s-1	lnd
Sa_va	inst_merid_wind_height_lowest_from_phys	m s-1	lnd
Faxa_swdn	mean_down_sw_flx	W m-2	lnd
Faxa_lwdn	mean_down_lw_flx	W m-2	lnd
Faxa_swnet	mean_net_sw_flx	W m-2	lnd
Faxa_rain	mean_prec_rate	kg m-2 s-1	lnd
Sa_prsl	inst_pres_height_lowest_from_phys	Pa	lnd
vfrac		0-1	lnd
Faxa_snow	mean_fprec_rate	kg m-2 s-1	lnd
Faxa_rainc	mean_prec_rate_conv	kg m-2 s-1	lnd
Sa_tskn	inst_temp_height_surface	K	lnd
Sa_exner	inst_exner_function_height_lowest	0-1	lnd
Sa_ustar	surface_friction_velocity	m s-1	lnd
zorl		cm	lnd

Export

Std. Name	Alias	Unit	Receiver
Sl_lfrac	land_fraction	0-1	mediator
Fall_lat	mean_laten_heat_flx_lnd	kg kg-1 m s-1	atm
Fall_sen	mean_sensi_heat_flx_lnd	K m s-1	atm
Fall_evap	mean_potential_laten_heat_flx_lnd	W m-2	atm
Fall_gflx	mean_upward_heat_flux_lnd	W m-2	atm
Fall_roff	mean_runoff_rate_lnd	kg m-2 s-1	atm
Fall_soff	mean_subsurface_runoff_rate_lnd	kg m-2 s-1	atm
Sl_sfrac	mean_snow_area_fraction_lnd	0-1	atm
Sl_tref	inst_temp_height2m_lnd	K	atm
Sl_qref	inst_spec_humid_height2m_lnd	kg kg-1	atm
Sl_q	inst_spec_humid_lnd	kg kg-1	atm
Sl_cmm	inst_drag_wind_speed_for_momentum	m s-1	atm
Sl_chh	inst_drag_mass_flux_for_heat_and_moisture	kg m-2 s-1	atm
Sl_zvfun	inst_func_of_roughness_length_and_vfrac	0-1	atm

Land Component Related Namelist Options

- Noah-MP configuration parameters: *nems.configure* and in *LND_attributes::* group

Option	Description	Possible Values
mosaic_file	The path and name of the mosaic grid file	any, INPUT/C96_mosaic.nc
input_dir	The directory that stores initial conditions, static information and grid related files	any
ic_type	Indicates the source of the initial conditions (custom - C96.initial.tile[1-6].nc and sfc - sfc_data.tile[1-6].nc)	custom, sfc (default)
layout	Defines decompositions in each direction on each tile (i.e. 3:8 for C96).	any number pair separated with double column
num_soil_levels	Number of soil levels used by NoahMP Land Model	any number
forcing_height	Height of the atmospheric forcing in meters. Used to overwrite component provided one.	any number
soil_level_thickness	Thickness of the soil levels. Needs to be consistent with num_soil_levels	Any list of number separated with double column
soil_level_nodes	Depths of the node points for each soil level.Needs to be consistent with num_soil_levels	Any list of number separated with double column
dynamic_vegetation_option	Options for dynamic vegetation	1 (off), 2 (on)
canopy_stomatal_resistance_option	Canopy stomatal resistance	1 (ball-berry), 2 (jarvis)
soil_wetness_option	Options for soil moisture factor for stomatal resistance	1 (noah), 2 (clm), 3 (ssib)
runoff_option	Options for runoff and groundwater	1 (simgm), 2 (simtop), 3 (schaake96), 4 (bats)
surface_exchange_option	Options for surface layer drag coefficient (ch and cm)	1 (m-o),2 (chen97)
supercooled_soilwater_option	Options for supercooled liquid water	1(ny06), 2 (koren99)
frozen_soil_adjust_option	Options for frozen soil permeability	1 (ny06), 2 (koren99)
radiative_transfer_option	Options for radiation transfer	1 (gap=f(3d,cosz), 2 (gap=0), 3 (gap=1-fveg)
snow_albedo_option	Options for snow surface albedo	1 (bats), 2 (class)
precip_partition_option	Options for rainfall & snowfall	1 (jordan91), 2 (bats), 3 (noah)
soil_temp_lower_bdy_option	Options for lower boundary of soil temperature	1 (zero-flux), 2 (noah)
soil_temp_time_scheme_option	Options for surface resistance	1->sakaguchi/zeng; 2->seller; 3->mod sellers; 4->1+snow
glacier_option	Options for glacier treatment (fixed to 2 in noahmpdrv)	(1->phase change; 2->simple)
output_freq	Options for output frequency in seconds	any number
restart_freq	Options for restatr frequency in seconds (negative for disabling)	any number
do_mynnedmf	Option for MYNN-EDMF (default is false)	true or false
do_mynnsfclay	Option for MYNN surface layer scheme (default is false)	true or false
soil_type_category	Option for soil type	0 (Zobler - 9 category), 1 (STATSGO - 19 category), 2 (STAS-RUC - 19 category)
veg_type_category	Option for source of vegetation data	0 (USGS), 1 (IGBP), 2 (UMD), 3 (NLCD40), 4 (USGS-RUC), 5 (MODI-RUC)
initial_emiss	Option for initial surface lw emissivity in fraction (default value is 0.95)	any number between 0-1
initial_albedo	Option for initial mean surface albedo (value is default 0.2)	any number between 0-1
has_export	Option to enable export fields (default value is true)	true or false
calc_snet	Option to calculate shortwave radiation internally (default value is false)	true or false

Additional configuration options in FV3 side:

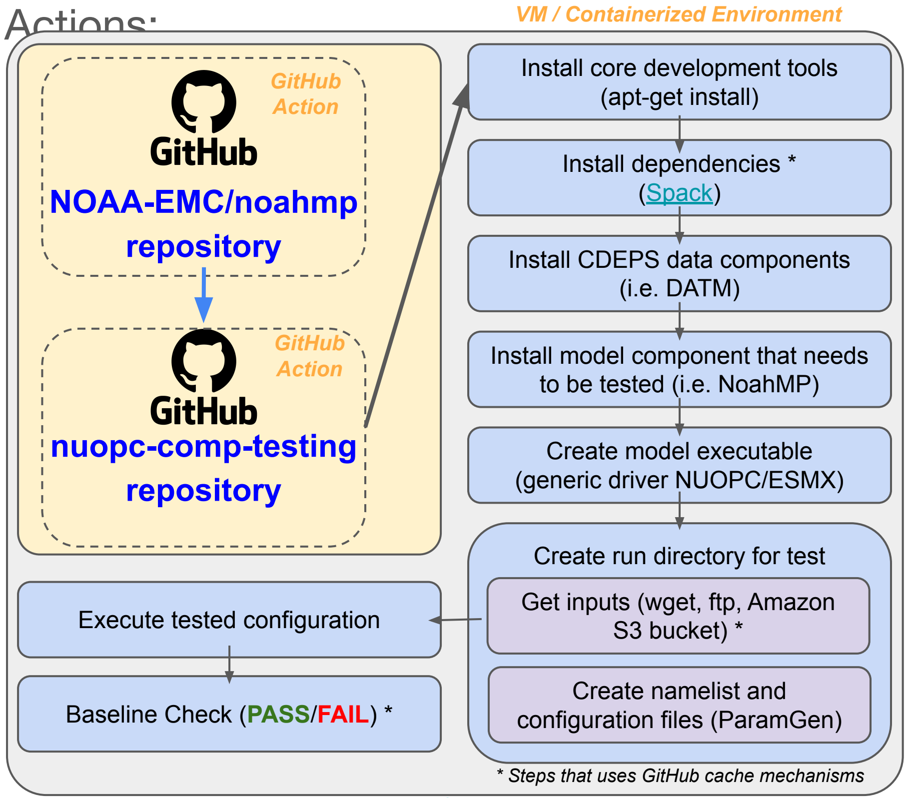
input.nml and *gfs_physics_nml* section

cpllnd - atm->lnd

cplInd2atm - lnd->atm

Automated Testing System

- New hierarchical testing capabilities for NoahMP component model via GitHub Actions:



https://github.com/esmf-org/nuopc-comp-testing

- It runs very simple DATM+Noah-MP configuration in every push or PR to Noah-MP repository
- GSWP3 forced C96 (~1° global) land configuration
- Check results against the baseline to find possible and unexpected baseline changes

Limitations

- Only tested with global cubed-sphere grid
  - *grid_spec.nc* and *oro_data.tile*.nc* files are used to define the global domain, mask information and land fraction.
  - It requires additional work to support regional domains
    - single tile regional FV3 configurations
    - Regular lat-lon grids (global and regional) can be also supported by reading in ESMF mesh file but this requires additional development and testing
- I/O layer also needs to be extended to work with other grids and configurations
  - Current implementation uses ESMF for I/O (based on PIO2 library)
- Need to develop set of tools to generate files that store static information and initial conditions for other grid types and regional applications

Noah-MP Component Model under UFS

- Supported configurations:
  - *DATM+LND* (incl. restart capability)
    - Tested with GSWP3 and ERA5 input provided by CDEPS data components
  - *FV3+LND side-by-side* configuration (no feedback from land)
    - FV3/CCPP/Noah-MP also runs
    - Aims to compare the results coming from component model and CCPP
  - *FV3+LND fully coupled* (incl. restart capability)
    - Fully active atmosphere coupled two-way with land component
    - Supports also running land component in higher resolution (i.e. C384 (~0.25° global)

NUOPC Run Sequence and CCPP Suite File

```
# run sequence for fully coupled case

runSeq::
@720
   MED med_phases_prep_atm
   MED -> ATM :remapMethod=redist
   ATM
   ATM -> MED :remapMethod=redist
   MED med_phases_post_atm
   MED med_phases_prep_lnd
   MED -> LND :remapMethod=redist
   LND
   LND -> MED :remapMethod=redist
   MED med_phases_post_lnd
   MED med_phases_restart_write
   MED med_phases_history_write
@
::
```

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="FV3_GFS_v17_p8" version="1">
  <group name="time_vary">...</group>
  <group name="radiation">...</group>
  <group name="physics">
    <subcycle loop="1">...</subcycle>
    <subcycle loop="2">
      <scheme>sfc_diff</scheme>
      <scheme>GFS_surface_loop_control_part1</scheme>
      <scheme>sfc_nst_pre</scheme>
      <scheme>sfc_nst</scheme>
      <scheme>sfc_nst_post</scheme>
      <scheme>noahmpdrv</scheme>
      <scheme>sfc_land</scheme>
      <scheme>sfc_sice</scheme>
      <scheme>GFS_surface_loop_control_part2</scheme>
    </subcycle>
    <subcycle loop="1">
      <scheme>GFS_surface_composites_post</scheme>
      ...
    </subcycle>
   </group>
  <group name="stochastics">...</group>
</suite>
```

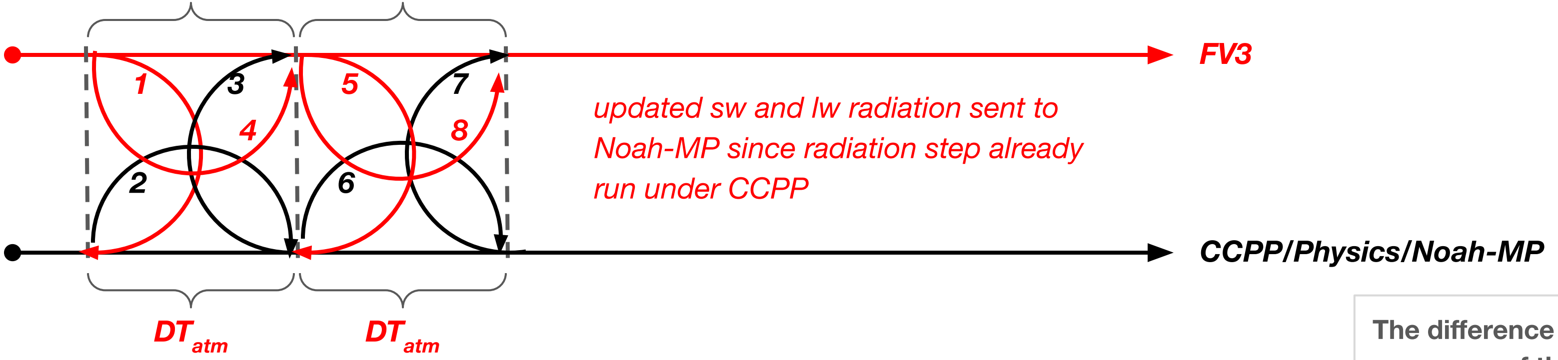
skips if cpllnd2atm = .true.

runs if cpllnd2atm = .true.

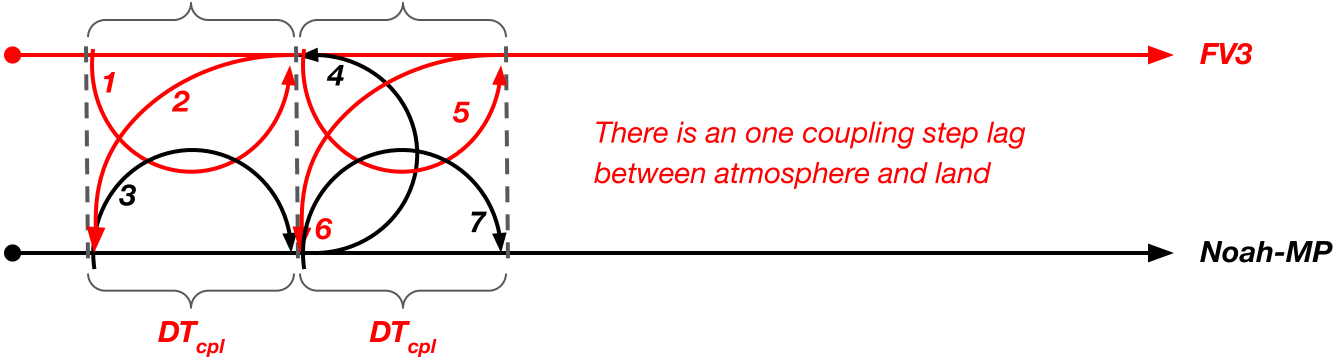
merges fluxes for fractional grid

Run Sequence: Standalone vs. Coupled

- Standalone atmosphere with CCPP/Physics Noah-MP



- Coupled atmosphere-land configuration



The difference in run sequence of the components might lead to slight difference in the results.

The only way to mimic FV3/CCPP way, is splitting the CCPP suite run phases, and allow calling component model just after radiation step.

Use Case I: Checking Out Code and Building Model

- The following steps aims to show basic steps to build fully-coupled configuration

```
# clone UFS Weather Model (feature branch)
git clone --recursive -b feature/noahmp https://github.com/uturuncoglu/ufs-weather-model.git

# build model (data atmosphere configurations)
cd ufs-weather-model/tests/
./compile.sh "cheyenne.intel" "-DAPP=LND" noahmp YES YES

# build model (fully-coupled)
cd ufs-weather-model/tests/
./compile.sh "cheyenne.intel" "-DAPP=ATML
-DCCPP_SUITES=FV3_GFS_v16,FV3_GFS_v15_thompson_mynn,FV3_GFS_v17_p8,FV3_GFS_v17_p8_rrtmgp,FV3_GFS_v15_t
hompson_mynn_lam3km,FV3_WoFS_v0,FV3_GFS_v17_p8_mynn -D32BIT=ON" noahmp YES YES

# NOTE: In this case ATML is used as application (fully-coupled) and cheyenne.intel is used as a
#       platform. The user might specify other platforms such as ufs_orion.intel
# NOTE: This will create executable under same directory - called as fv3_noahmp.exe
# NOTE: Available applications are
#       (1) LND   : CMEPS, CDEPS and NOAHMP
#       (2) ATML  : FMS, FV3, STOCH_PHYS, CMEPS, NOAHMP
#       (3) S2SWAL: CMEPS, FMS, FV3, MOM6, CICE6, STOCH_PHYS, WW3, GOCART and NOAHMP
#       The option (3) is not tested yet.
```

Use Case II: Running Data Component Configurations

- Running RT with data component configuration (C96)

```
# clone UFS Weather Model
git clone --recursive -b feature/noahmp https://github.com/uturuncoglu/ufs-weather-model.git

# run ERA5 RT
cd ufs-weather-model/tests/
./rt.sh -k -n datm_cdeps_lnd_era5

# NOTE: -k argument keep run directory and -n argument runs only specified test
# NOTE: datm_cdeps_lnd_era5_rst can be run to test restart reproducibility
# NOTE: Since this RT is under development, the input directories are not stored in the common
location
#       User need to set DISKNM as /glade/work/turuncu/RT in tests/rt.sh for NCAR Cheyenne platform
```

- Running same RT but using higher resolution land (C384, requires land IC for C384 !!!)

```
# edit ufs-weather-model/tests/tests/datm_cdeps_lnd_era5 and change LNDRES parameter from C96 to C384

# and run ERA5 RT again with same command
cd ufs-weather-model/tests/
./rt.sh -k -n datm_cdeps_lnd_era5
```

Use Case III: Running Fully Coupled Configuration

- Running RT with fully coupled configuration (both component are in C96)

```
# clone UFS Weather Model
git clone --recursive -b feature/noahmp https://github.com/uturuncoglu/ufs-weather-model.git

# run fully-coupled RT
cd ufs-weather-model/tests/
./rt.sh -k -n control_p8_atmlnd
```

- Running same RT but using higher resolution land (C384)

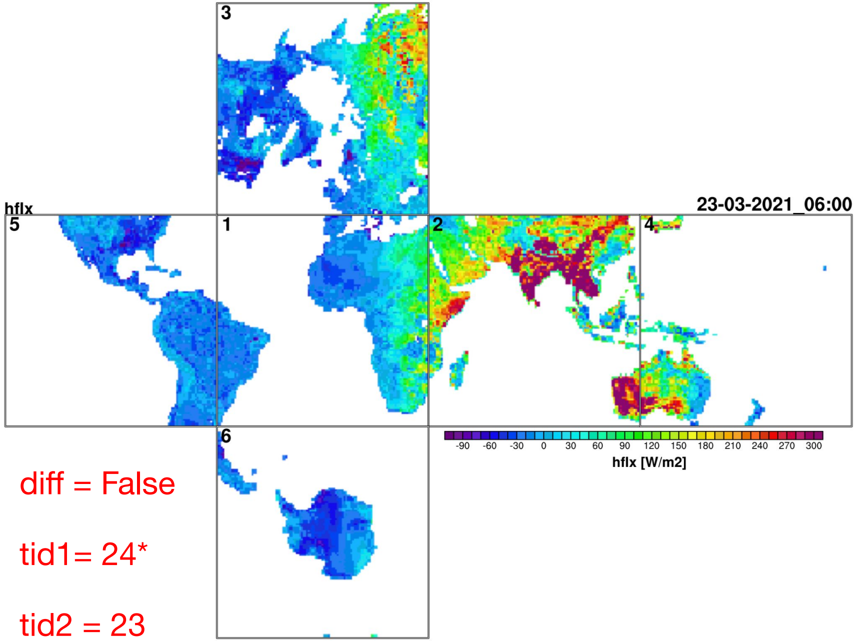
```
# edit ufs-weather-model/tests/tests/control_p8_atmlnd and change LNDRES parameter from C96 to C384

# and run fully-coupled RT again with same command
cd ufs-weather-model/tests/
./rt.sh -k -n control_p8_atmlnd

# NOTE: In this case, ufs.cpld.lnd.out.* files will be in C384 resolution.
# NOTE: As it expected, baseline check will fail since the files belong to C96 resolution.
```

Use Case IV: Checking Results

- The simple plotting scripts can be found in here: (1) *plot_cubed.ncl* for spatial plot and (2) *plot_cubed_ts.ncl* for time-series
- Steps:
  - Run test case (i.e. *control_p8_atmlnd_sbs*)
  - Copy scripts to the run directory that has both FV3 (*sfcf** files) and Noah-MP output (*ufs.cpld.lnd.out.** files)
  - Run *plot_cubed.ncl NCL* script. In this case, *vname1* and *vname2* can be changed to create different plots. If diff argument is set to True, then the script generates difference plot. Note that, FV3 needs to start time index 1 since it has initial data.
  - Run *plot_cubed_ts.ncl NCL* script to create time series. It uses again *vname1* and *vname2* to define variable names and predefined grid points in each tile (xc, yc).

Use Case IV: Checking Results …

- Simple output of sensible heat flux from *plot_cubed.ncl* after 1 day simulation.

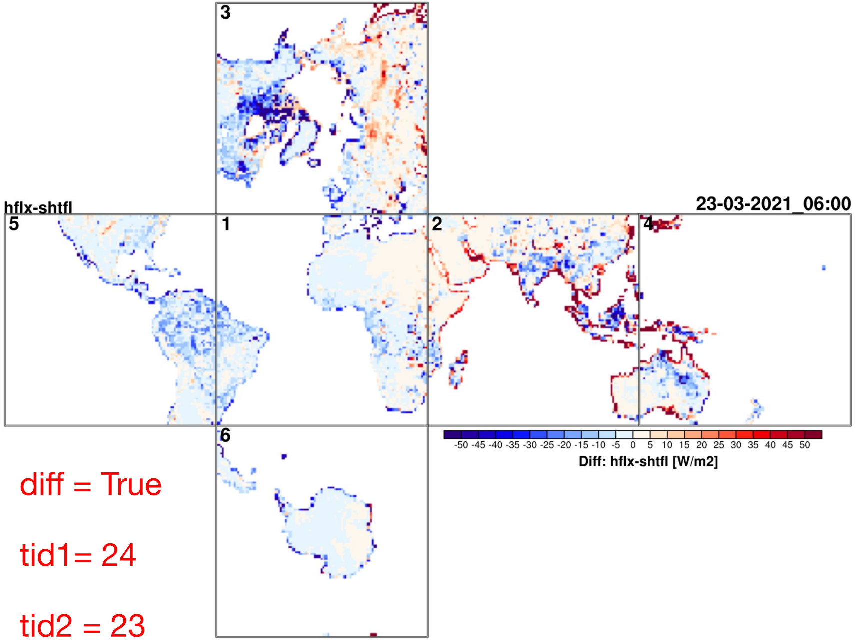


diff = False

tid1= 24*

tid2 = 23

dataset = 1 - Noah-MP



diff = True

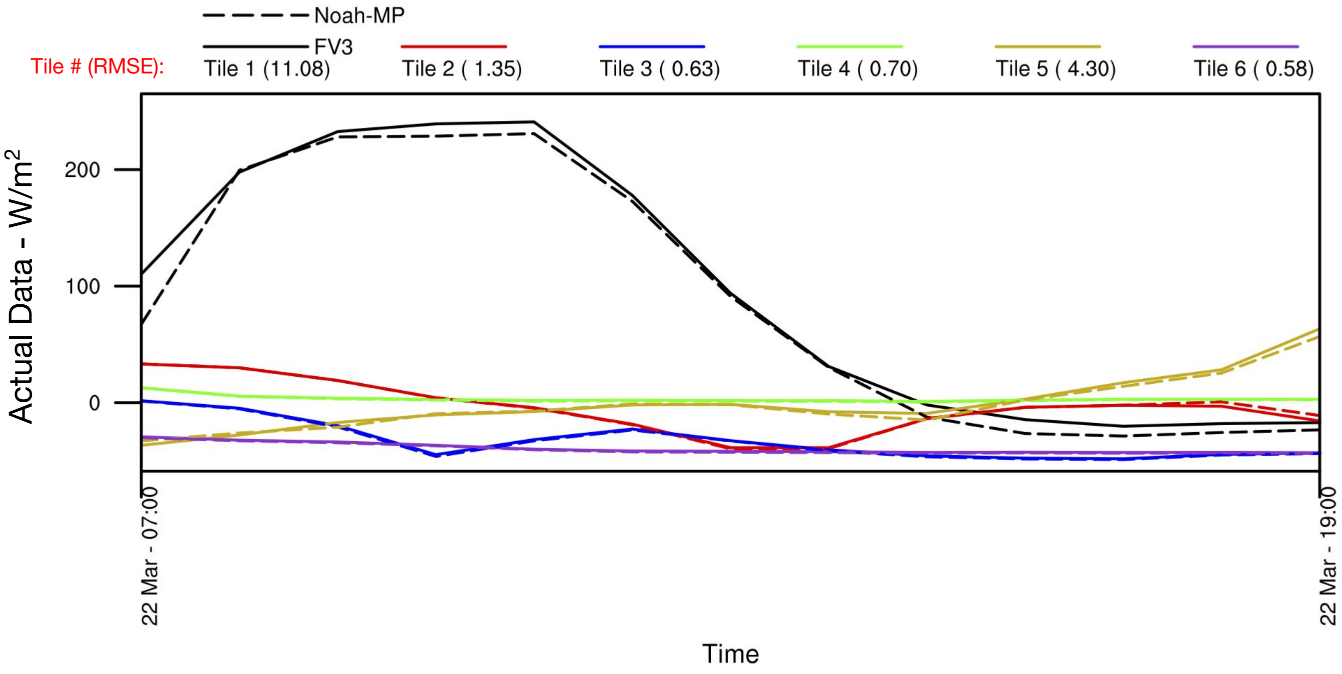
tid1= 24

tid2 = 23

* sfcf000 is written after first atm dt

Use Case IV: Checking Results …

- Simple output of sensible heat flux from *plot_cubed_ts.ncl*

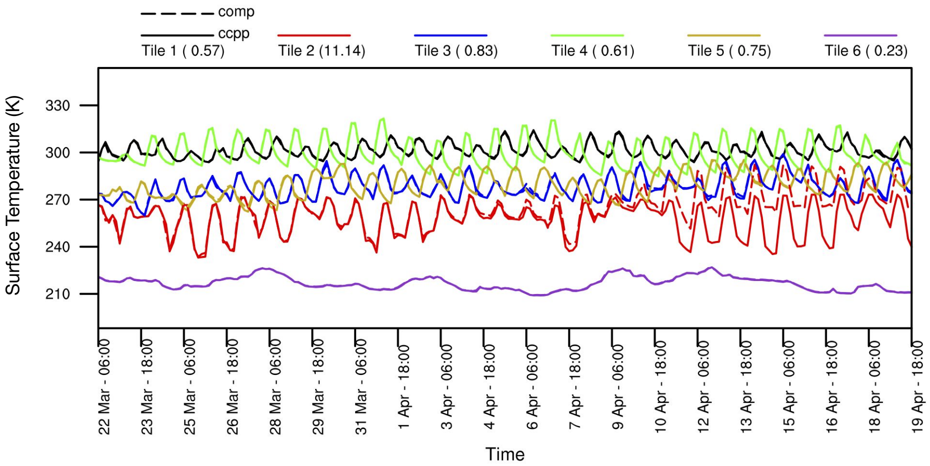


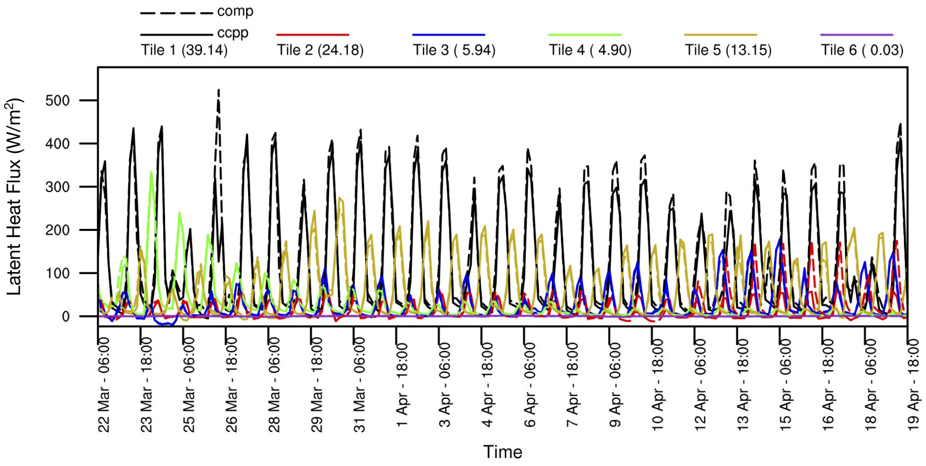
Questions?

Ufuk Turuncoglu: turuncu@ucar.edu
Michael Barlage: michael.barlage@noaa.gov

Validation of The Component Model

- A set of 30 days run was performed with different configurations of the model
- Side-by-side comparison (*control_p8_atmlnd_sbs* - sfcf* vs. land output)

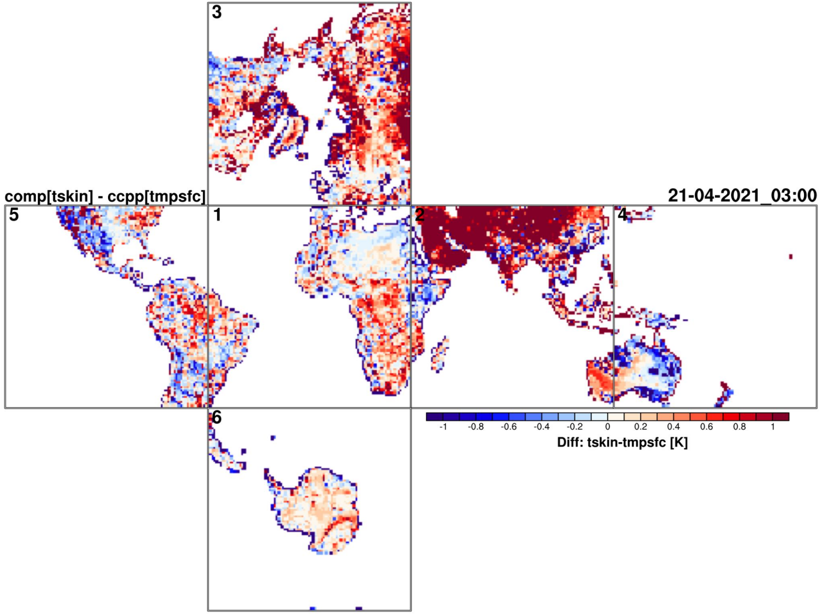


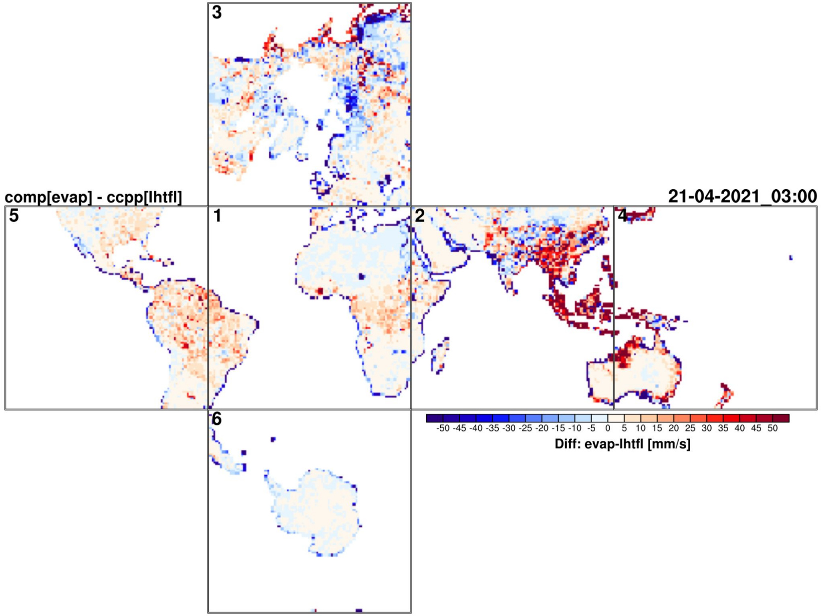


- The component version of the Noah-MP is able to reproduces the results of FV3/CCPP version. The solution start to diverge after couple of weeks (see tile 2 results)

Validation of The Component Model …

- Difference after 30-days for *control_p8_atmlnd_sbs*





- Since the solution diverges, the difference between component model and CCPP version is relatively large at the end of the simulation (after 30 days). This is not same for all regions and tiles and requires further investigation.

C96 vs C384

- The fully coupled configuration - *control_p8_atmlnd* is also run with two resolution.
    - C96 ATM / C96 LND vs. C96 ATM / C384 LND
- Following figure compares FV3ATM output of two runs after 30 days of simulation