The Common Community Physics Package: overview and available physics suites

Dom Heinzeller

University of Colorado Cooperative Institute for Research in Environmental Sciences (CIRES)
at NOAA/ESRL Global Systems Division & Developmental Testbed Center

AMS 2020 short course SCM-CCPP, Jan 12, 2020 – Boston, MA

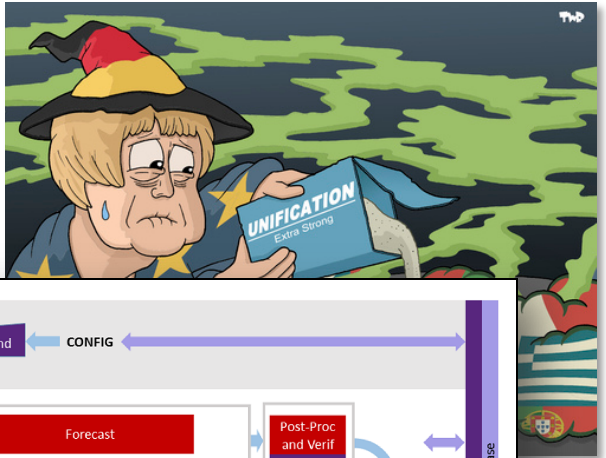
Model unification - omnipresent and misunderstood

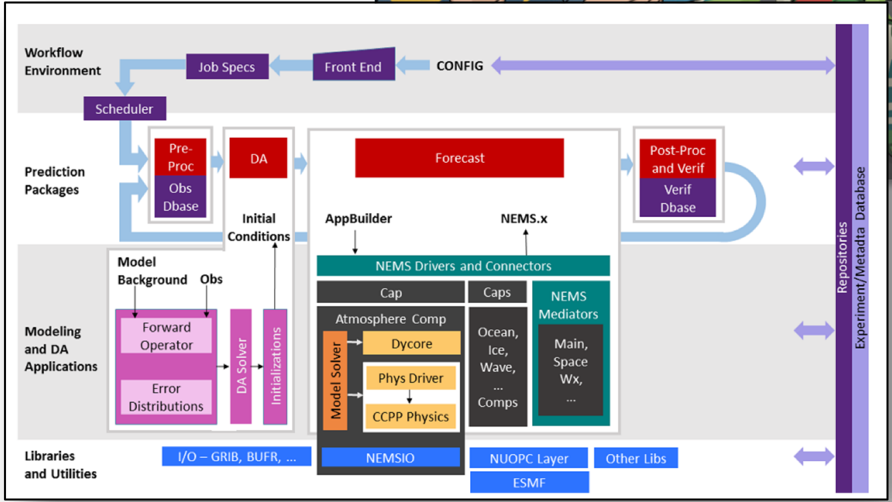
Unification has become a buzz word, but it is often misunderstood.

In atmospheric sciences, unification does **not** imply a single model and set of physics that everyone must use.

Unification refers to creating a system of earth model components that can be exchanged and combined easily …

- to foster collaborations and facilitate research and development
- to accelerate transitions from research to operations and vice versa
- to avoid duplication of efforts and repetition of mistakes





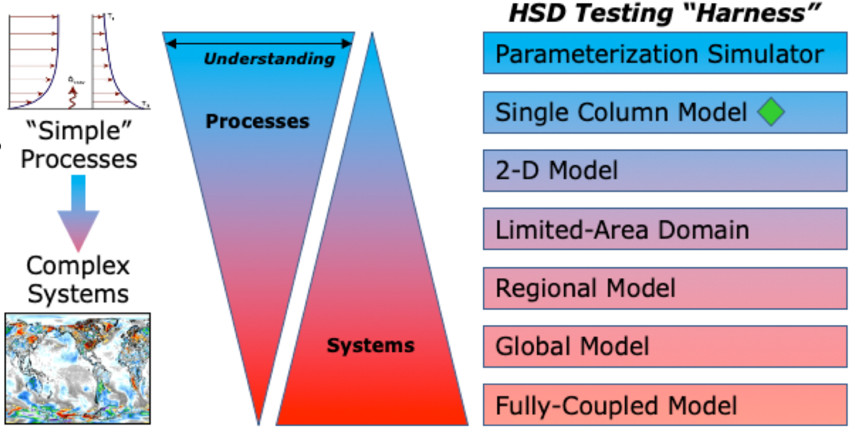
NOAA's Unified Forecasting System

The Unified Forecast System (UFS)

- is a community-based, coupled comprehensive earth system modeling system
- is designed to provide numerical guidance for applications in the forecast suite of NOAA's National Centers for Environmental Prediction (NCEP)
- spans local to global domains and predictive time scales from hours to years
- provides the foundation for closing the gap between ECMWF and NCEP

One cornerstone of the UFS is to **facilitate the improvement of physical parameterizations** and their transition from research to operations.

https://ufscommunity.org/index.html



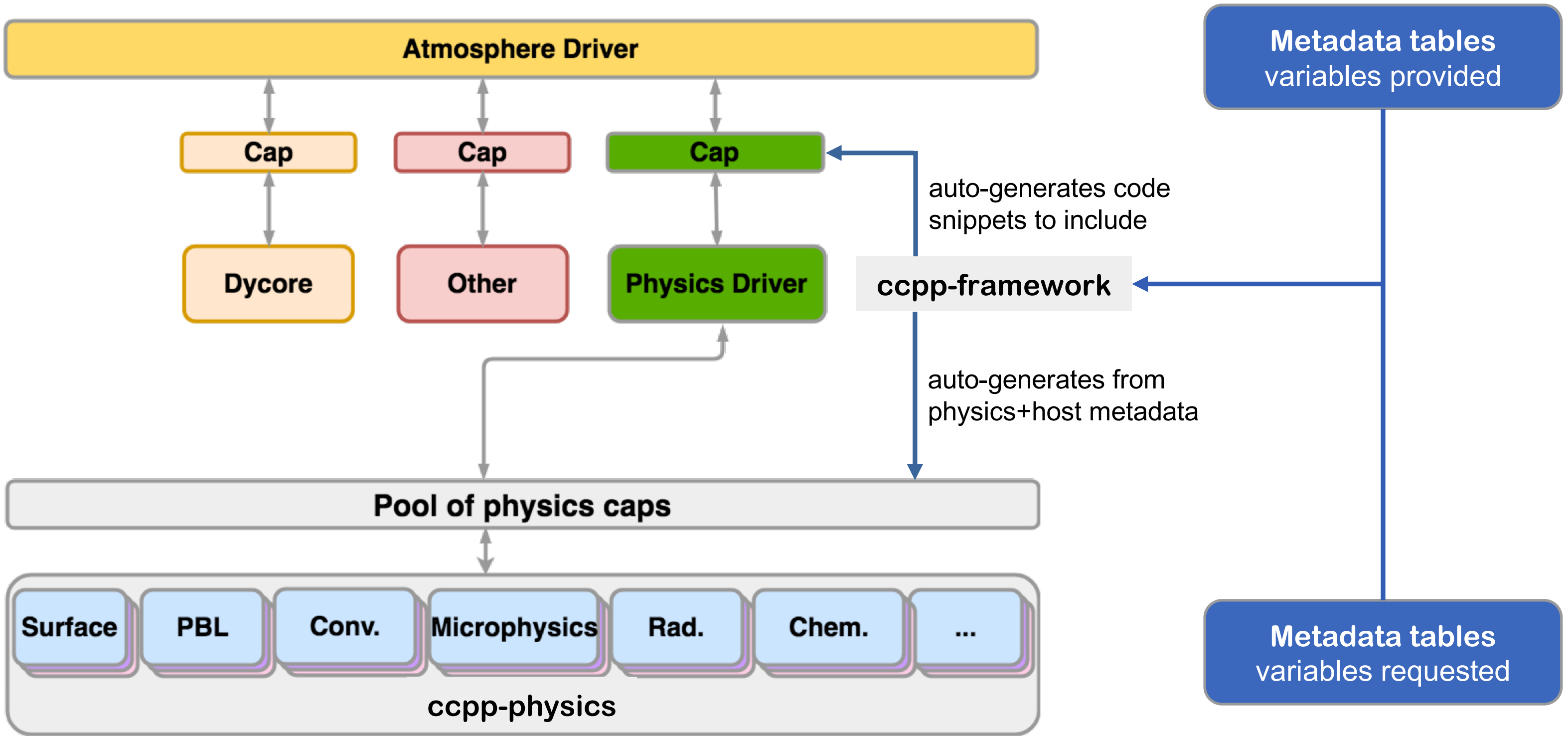
Infrastructure for development of model physics

The Common Community Physics Package consists of an infrastructure component, **ccpp-framework,** and a collection of compliant physics suites, **ccpp-physics**.

Driving principles:

- Readily available and well supported: open source, on GitHub
- Model-agnostic to enable collaboration and accelerate innovations
- Documented interfaces (metadata) facilitate developing and testing physics
- Inline documentation using doxygen for generating scientific documentation
- Physics suite construct is important, but the CCPP must enable easy interchange of schemes within a suite (need for interstitial/glue code)

The CCPP within the model system



Key features of the CCPP

- **Compile-time configuration**: suite definition file (XML)
- **Grouping**: schemes can be called in groups with other computations in between (e.g. dycore, coupling)
- **Subcycling/iterations**: schemes can be called at higher frequency than others/dynamics
- **Ordering**: user-defined order of execution of schemes (may require changing interstitial code)

```
<suite name="GFS_v15p2">
  ...

</suite>
```

Key features of the CCPP

- **Compile-time configuration**: suite definition file (XML)
- **Grouping**: schemes can be called in groups with other computations in between (e.g. dycore, coupling)
- **Subcycling/iterations**: schemes can be called at higher frequency than others/dynamics
- **Ordering**: user-defined order of execution of schemes (may require changing interstitial code)

```
<suite name="GFS_v15p2">
...
  <group name="radiation">
   <scheme>GFS_rrtmg_pre</scheme>
   <scheme>rrtmg_sw_pre</scheme>
   <scheme>rrtmg_sw</scheme>
   <scheme>rrtmg_sw_post</scheme>
   <scheme>rrtmg_lw_pre</scheme>
   <scheme>rrtmg_lw</scheme>
   <scheme>rrtmg_lw_post</scheme>
   <scheme>GFS_rrtmg_post</scheme>
  </group>
...
</suite>
```

suite interstitial   scheme interstitial   scheme

Writing a CCPP-compliant parameterization is easy

```
module myscheme
  implicit none
  contains
  subroutine myscheme_init ()
  end subroutine myscheme_init
!> \section arg_table_myscheme_run Argument Table
!! \htmlinclude myscheme_run.html
!!
  subroutine myscheme_run(ni, psfc, errmsg, errflg)
    integer,            intent(in)    :: ni
    real,               intent(inout) :: psfc(:)
    character(len=*),   intent(out)   :: errmsg
    integer,            intent(out)   :: errflg
    ...
  end subroutine myscheme_run
  subroutine myscheme_finalize()
  end subroutine myscheme_finalize
end module myscheme
```

myscheme.F90

```
[ccpp-arg-table]
  name = myscheme_run
  type = scheme
[ni]
  standard_name = horizontal_dimension
  long_name = horizontal dimension
  units = count
  dimensions = ()
  type = integer
  intent = in
  optional = F
[psfc]
  standard_name = surface_air_pressure
  long_name = air pressure at surface
  units = Pa
  dimensions = (horizontal_dimension)
  type = real
  intent = inout
  optional = F
...
```

myscheme.meta

Metadata is used for scientific documentation

Common Community Physics Package (CCPP) Scientific Documentation: GFDL Cloud Microphysics Module

Common Community Physics Package (CCPP) Scientific Documentation Version 3.0

Search

- Common Community Physics Package
  - Introduction
  - Parameterizations and Suites Overvi
  - GFS_v15 Suite
  - GFS_v15plus Suite
  - csawmg Suite
  - GSD_v0 Suite
  - Namelist Options Description
  - Bibliography
  - Modules
    - GFDL In-Core Fast Saturation Adj
    - GFS Physics Time Update
    - GFS RRTMG Longwave Module
    - GFS RRTMG Shortwave Module
    - RRTMG Aerosols Module
    - RRTMG Astronomy Module
    - RRTMG Clouds Module
    - RRTMG Gases Module
    - RRTMG Surface Module
    - RRTMG dcyc2t3 Module
    - GFS Surface Layer Scheme Modul
    - GFS Near-Surface Sea Temperatu
    - GFS Noah LSM Model
    - GFS Three-layer Thermodynamic
    - GFS Hybrid Eddy-Diffusivity Mass
    - GFS Scale-aware TKE-based Mois
    - GFS Orographic Gravity Wave Dra
    - GFS Rayleigh Damping Module
    - GFS Ozone Photochemistry (201
    - GFS Water Vapor Photochemical I
    - GFS Scale-Aware Mass-Flux Deep
    - GFS Convective Gravity Wave Dra
    - GFS Scale-Aware Mass-Flux Shal
    - GFS Convective Cloud Diagnostic
    - GFDL Cloud Microphysics Module
    - GFS Precipitation Type Diagnosti
    - Morrison-Gettelman MP scheme
    - CSAW adjustment Module
    - Chikira-Sugiyama Cumulus Schei

Modules

GFDL Cloud Microphysics Module

This is cloud microphysics package for GFDL global cloud resolving model. The algorithms are originally derived from Lin et al. (1983) [99]. most of the key elements have been simplified/improved. This code at this stage bears little to no similarity to the original Lin MP in zetac. therefore, it is best to be called GFDL microphysics (GFDL MP) . More...

Detailed Description

Author

Shian-Jiann Lin, Linjiong Zhou

The module contains the GFDL cloud microphysics (Chen and Lin (2013) [28]). The module is paired with **GFDL In-Core Fast Saturation Adjustment Module**, which performs the "fast" processes.

The subroutine executes the full GFDL cloud microphysics.

Argument Table

local_name	standard_name	long_name	units	rank	type	kind	intent	optional
levs	vertical_dimension	number of vertical levels	count	0	integer		in	F
im	horizontal_loop_extent	horizontal loop extent	count	0	integer		in	F
con_g	gravitational_acceleration	gravitational acceleration	m s-2	0	real	kind_phys	in	F
con_fvirt	ratio_of_vapor_to_dry_air_gas_constants_minus_one	rv/rd - 1 (rv = ideal gas constant for water vapor)	none	0	real	kind_phys	in	F
con_rd	gas_constant_dry_air	ideal gas constant for dry air	J kg-1 K-1	0	real	kind_phys	in	F
frland	land_area_fraction_for_microphysics	land area fraction used in microphysics schemes	frac	1	real	kind_phys	in	F
garea	cell_area	area of grid cell	m2	1	real	kind_phys	in	F
gq0	water_vapor_specific_humidity_updated_by_physics	water vapor specific humidity updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntcw	cloud_condensed_water_mixing_ratio_updated_by_physics	cloud condensed water mixing ratio updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntrw	rain_water_mixing_ratio_updated_by_physics	moist mixing ratio of rain updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntiw	ice_water_mixing_ratio_updated_by_physics	moist mixing ratio of cloud ice updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntsw	snow_water_mixing_ratio_updated_by_physics	moist mixing ratio of snow updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntgl	graupel_mixing_ratio_updated_by_physics	moist mixing ratio of graupel updated by physics	kg kg-1	2	real	kind_phys	inout	F
gq0_ntclamt	cloud_fraction_updated_by_physics	cloud fraction updated by physics	frac	2	real	kind_phys	inout	F
gt0	air_temperature_updated_by_physics	air temperature updated by physics	K	2	real	kind_phys	inout	F
gu0	x_wind_updated_by_physics	zonal wind updated by physics	m s-1	2	real	kind_phys	inout	F
gv0	y_wind_updated_by_physics	meridional wind updated by physics	m s-1	2	real	kind_phys	inout	F

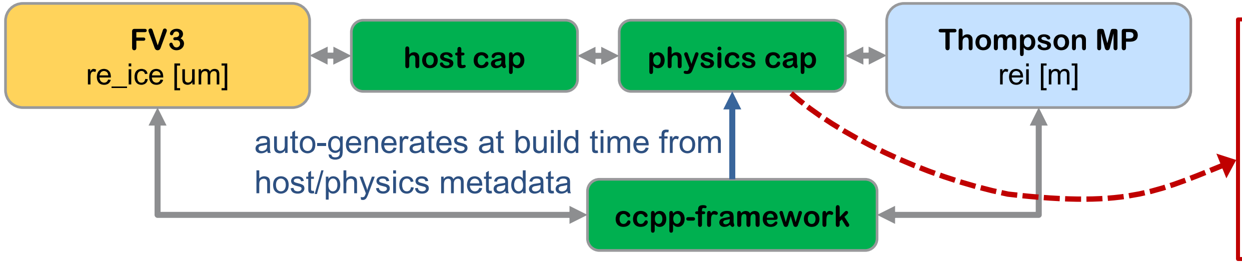
Generated by doxygen 1.8.11

https://dtcenter.org/gmtb/users/ccpp

CCPP provides options for performance and flexibility

- CCPP uses a multi-suite static build to maintain the required performance for operations
  - Compile options for the UFS (and DTC's Single Column Model SCM): `SUITES="abc,xyz,…"`
  - Filters unused suites and variables, and auto-generates Fortran caps for each of the suites

- CCPP supports automatic unit conversions to expediate development and transition




```
allocate(tmp1, source=re_ice)
tmp1=1.0E-6*re_ice
call thompson_mp(rei=tmp1,...)
re_ice=1.0E+6*tmp1
deallocate(tmp1)
```

Parallelization in CCPP: limited MPI, full threading

Overarching paradigms

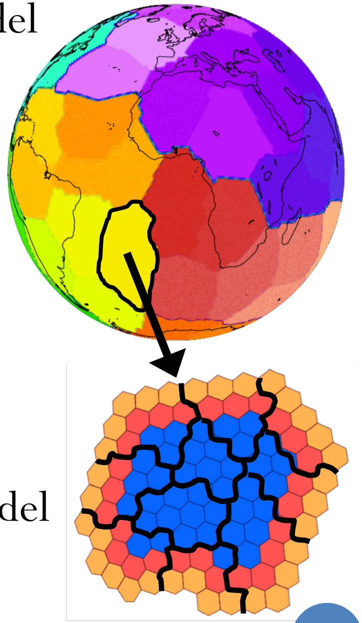
- physics are column-based, no communication during time integration in physics
- physics initialization/finalization are independent of threading strategy of the model

MPI

- MPI communication only allowed in the physics initialization/finalization
- use MPI communicator provided by host model, not `MPI_COMM_WORLD`

OpenMP

- time integration (but not init./final.) can be called by multiple threads
- threading inside physics is allowed, use # OpenMP threads provided by host model



Timeline of CCPP Activities

Ideas
2016

Requirements
2017

Design
2017-ongoing

Develop
2017-ongoing

Public releases
v1-v2-v3: 2018-19

NCAR/NRL collaboration
2018-ongoing

Adopted by EMC
2019

Training for NOAA
2019

UFS Public Release
2020

UFS Tutorial
2020

Operations
2022-2023

Parameterizations in master CCPP code (Jan. 2020)

Microphysics	Zhao-Carr, GFDL (incl. sat adj in dycore), MG2-3, Thompson, F-A
PBL	K-EDMF, TKE-EDMF, moist TKE-EDMF, YSU, saYSU, MYJ
Surface Layer	GFS, MYNN, MYJ
Deep Convection	saSAS, Chikira-Sugiyama, GF, Tiedtke
Shallow Convection	EDMF, GF, Tiedtke
PBL and Shal Convection	SHOC, MYNN
Radiation	RRTMG
Gravity Wave Drag	GFS orographic, GFS convective, uGWD, RAP/HRRR drag suite
Land Surface	Noah, Noah-MP, RUC
Ocean	Simple GFS ocean
Sea Ice	Simple GFS sea ice
Ozone	2006 NRL, 2015 NRL
H_2O	NRL

Implementation
DTC
GSD
OU
EMC

CCPP v4 Supported Suites: Operations & Research

	Operational	Experimental		
	GFS_v15.2	**GFS_v16beta**	**csawmg**	**GSD_v1**
Microphysics	GFDL	GFDL	**M-G3**	**Thompson**
Boundary Layer	K-EDMF	**TKE EDMF**	K-EDMF	**saMYNN**
Surface Layer	GFS	GFS	GFS	GFS
Deep convection	saSAS	saSAS	**Chikira-Sugiyama**	**Grell-Freitas**
Shallow Convection	saSAS	saSAS	saSAS	**saMYNN and GF**
Radiation	RRTMG	RRTMG	RRTMG	RRTMG
Gravity Wave Drag	uGWD	uGWD	uGWD	uGWD
Land Surface	Noah	Noah	Noah	**RUC**
Ozone	NRL 2015	NRL 2015	NRL 2015	NRL 2015
H_2O	NRL	NRL	NRL	NRL

Additional parameterizations and suites are underdevelopment

CCPP Public Releases

V	Date	Physics	Host
v1	2018 Apr	GFS v14 operational	SCM
v2	2018 Aug	GFS v14 operational updated GFDL microphysics	SCM UFS WM for developers
v3	2019 Jul	GFS v15 operational Developmental schemes/suites	SCM UFS WM for developers
v4	2020	GFS v15 operational Developmental schemes/suites (incl GFS v16 developmental)	SCM UFS WM

CCPP v3: https://dtcenter.org/ccpp

- Docs: Scientific Doc, Users Guide, Technical Documentation, FAQ
- Helpdesk: gmtb-help@ucar.edu

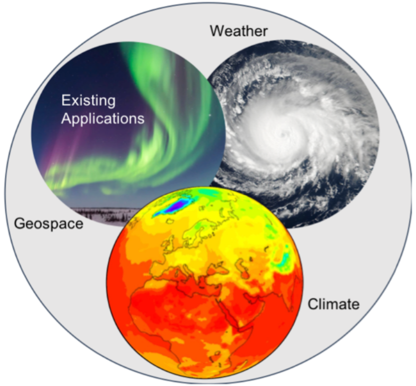
UFS WM – UFS Weather Model
SCM – CCPP Single Column Model

NOAA–NCAR Memorandum of Agreement (2019)

In 2019, NOAA and NCAR agreed to jointly develop the CCPP framework as a single system to communicate between models and physics.

NCAR contributions to the CCPP framework (within SIMA*):

- Augmented metadata standard to provide information on
  - Coordinate variables and vertical direction
  - Dimensions and index ordering of arrays
  - State variables, tendencies, persistent variables
  - Tracers and what to do with them (e.g. advection)
- Automatic variable allocation for variables used by physics suite only
- Compare metadata to actual Fortran code
- Improved build system and code generator

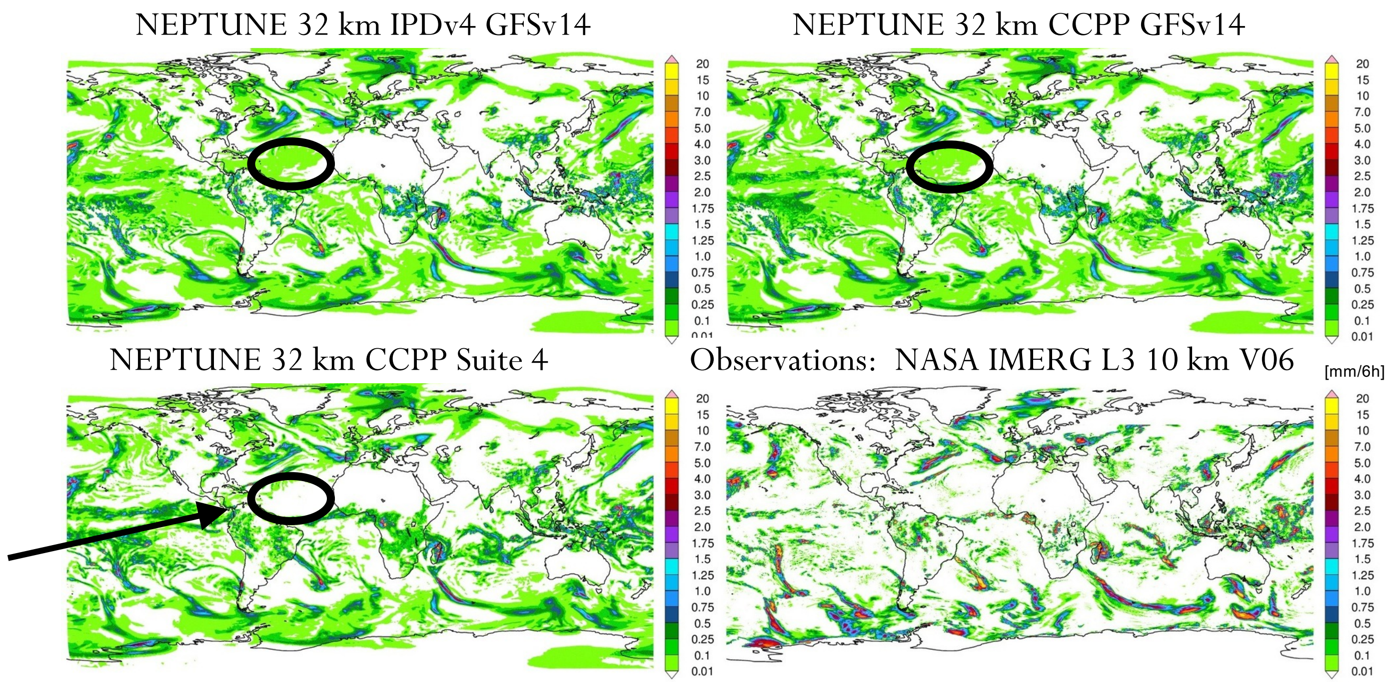


*SIMA: System for Integrated Modeling-Atmosphere

Collaboration with NRL NEPTUNE

- CCPP implemented in NEPTUNE by NRL
- Experiments in NEPTUNE have been conducted with various suites

Suite 4 improves drizzle bias



Total precipitation (explicit + parameterized) for 60-h forecast (mm/h)

Courtesy of Matus Martini (Devine), Alex Reinecke and Jim Doyle (NRL)

The CCPP is used by several U.S. flagship models