The Common Community Physics Package (CCPP): a shared infrastructure for model physics for operations and research

Dom Heinzeller[1,3,5], Grant Firl[2,3], Ligia Bernardet[1,3], Laurie Carson[2,3], Man Zhang[1,3,5], Jack Kain[4]

[1] NOAA/ESRL Global Systems Laboratory
[2] NCAR Research Applications Laboratory;
[3] Developmental Testbed Center
[4] NOAA National Severe Storms Laboratory
[5] University of Colorado Cooperative Institute for Research in Environmental Sciences

Developmental Testbed Center

6th ENES HPC Workshop, 05/28/2020

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 by enabling the community to participate in the development and testing.

https://ufscommunity.org/index.html

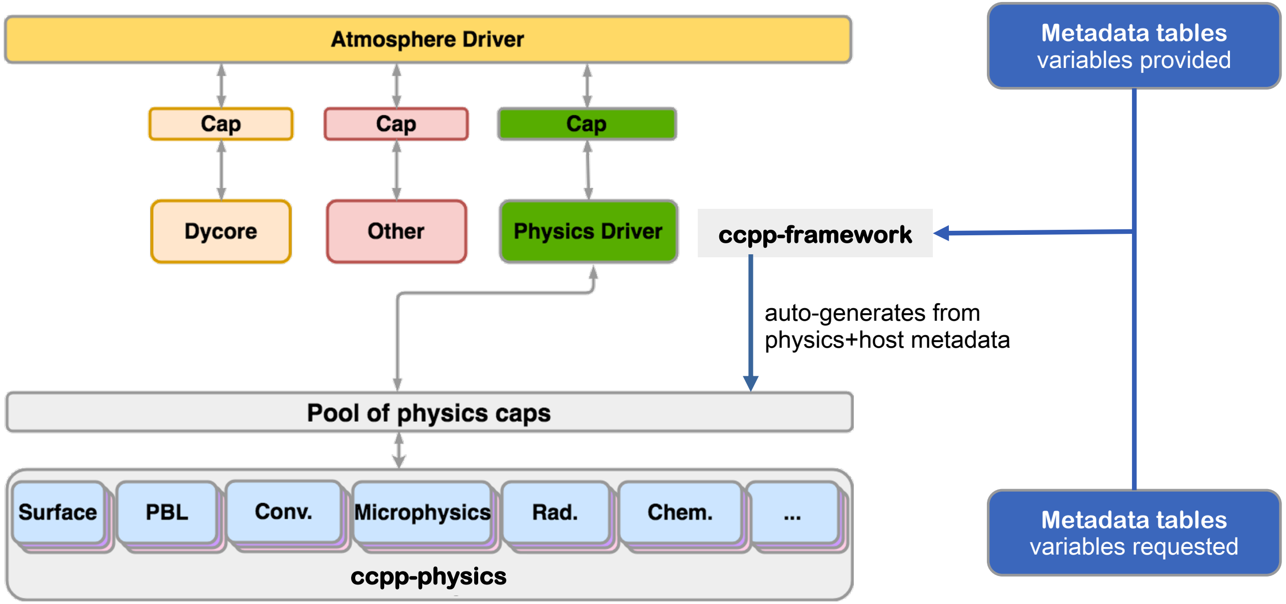
Infrastructure for development of model physics

The Common Community Physics Package (CCPP) 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, accepting external contributions (review/approval process)
- Model-agnostic to enable collaboration and accelerate innovations
- Documented interfaces (metadata) facilitate using existing schemes adding new schemes or transferring them between models
- Physics suite construct (vetted combination of schemes) is important, but the CCPP must enable easy interchange of schemes within a suite
- Scientific documentation generated from inline doxygen markup and metadata

The CCPP within the model system



- ccpp-framework/auto-generated caps replace traditional physics drivers
- glue code between physics in drivers needs to become interstitial schemes

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">
    ...
  </group>
  <group name="physics">
    ...
    <!-- Surface iteration loop -->
    <subcycle loop="2">
      ...
      <scheme>lsm_noah</scheme>
      ...
    </subcycle>
    ...
  </group>
</suite>
```

Writing a CCPP-compliant parameterization is easy

```fortran
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

CCPP Scientific Documentation: GFDL Cloud Microphysics Module

CCPP Scientific Documentation

v4.0

Search

- GFS RRTMG Longwave 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 Simple Ocean Scheme Modul
- GFS Noah LSM Model
- GFS Three-layer Thermodynamic
- GFS Hybrid Eddy-Diffusivity Mass
- GFS Scale-aware TKE-based Mois
- CIRES Unified Gravity Wave Physi
- 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 Scale-Aware Mass-Flux Shal
- GFS Convective Cloud Diagnostic
- GFDL Cloud Microphysics Module
- GFS Precipitation Type Diagnosti
- GFS Stochastics Physics Module
- Morrison-Gettelman MP Driver M
- CSAW adjustment Module
- Chikira-Sugiyama Cumulus Sche
- Grell-Freitas Convection Scheme
- GSD MYNN-EDMF PBL Scheme M
- GSD RUC LSM Model
- Aerosol-Aware Thompson MP M
- GFS Physics Function Module
- GFS Physics Parameter Module
- GFS Physics Constants Module
- GFS RRTMG Constants Module
- Mersenne Twister Module
- Files

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) [106]. 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) [31]). 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

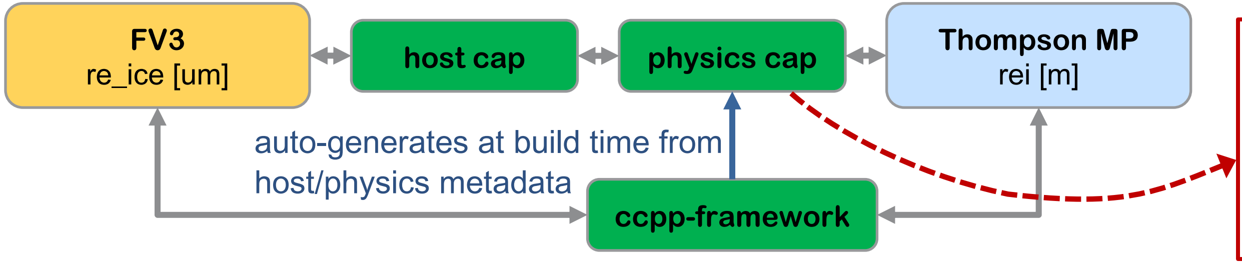
local_name	standard_name	long_name	units	type	dimensions	kind	intent	optional
levs	vertical_dimension	number of vertical levels	count	integer	()		in	False
im	horizontal_loop_extent	horizontal loop extent	count	integer	()		in	False
con_g	gravitational_acceleration	gravitational acceleration	m s-2	real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

Generated by doxygen 1.8.11

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 schemes and variables, and auto-generates Fortran caps for each of the suites

- CCPP supports automatic unit conversions to expedite 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

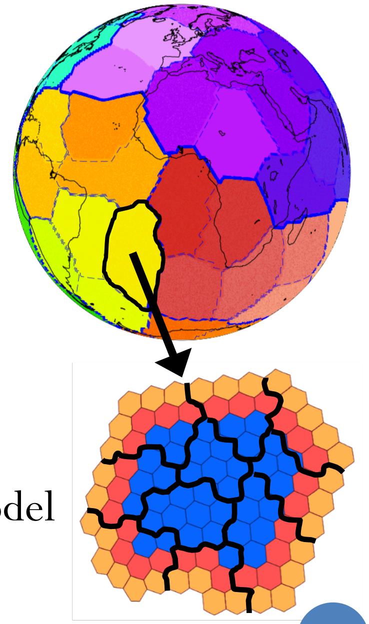
- physics are column-based, no communication during time integration in physics
- physics initialization/finalization can not be called by multiple threads

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

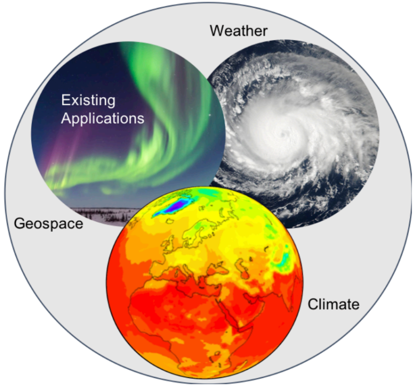


NOAA–NCAR Memorandum of Agreement (2019)

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 only
- Compare metadata to actual Fortran code
- Improved build system and code generator

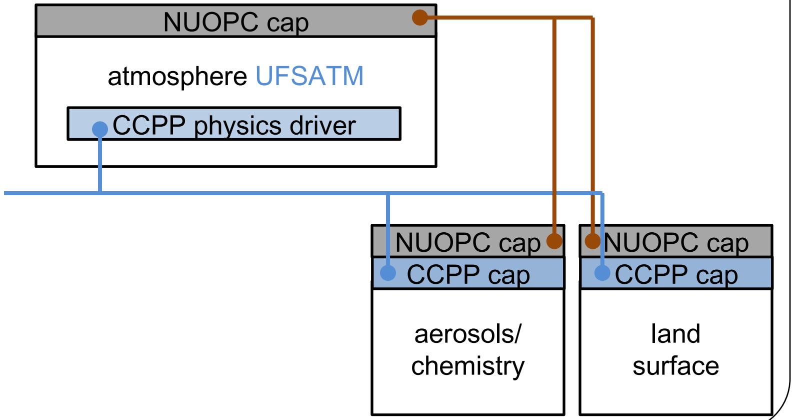


*SIMA: System for Integrated Modeling-Atmosphere

A bounty of low- (and higher-)hanging fruit

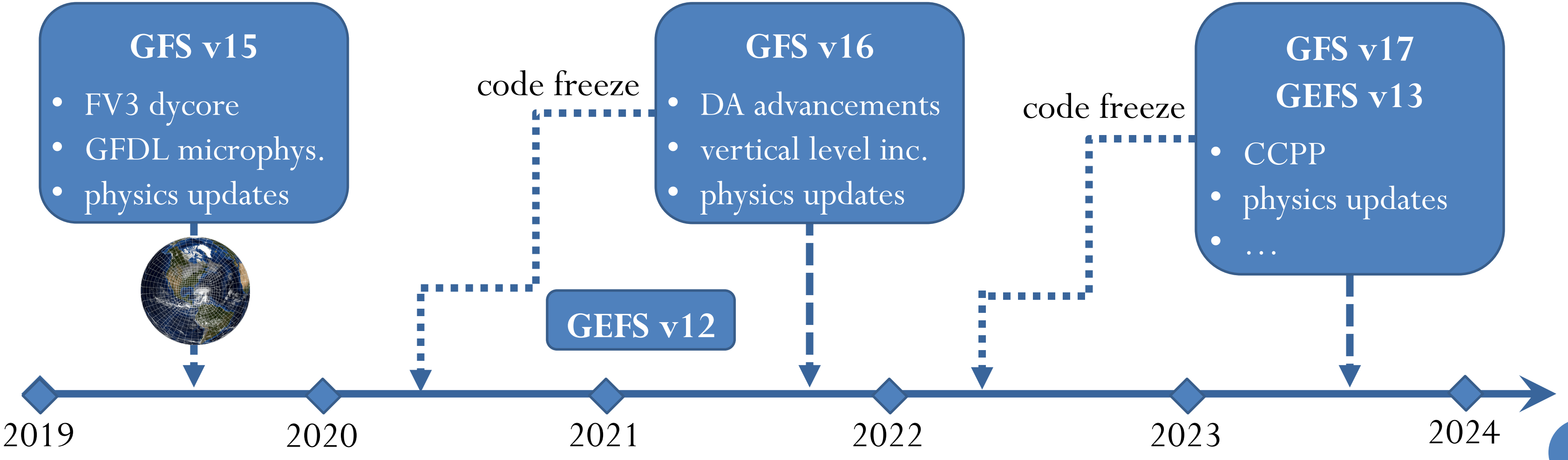
The existing CCPP framework capabilities and the NCAR contributions provide opportunities for development:

- Automatic array transformations (i,j,k to i,k to k,i to …)
- Calculation of derived variables (pot. temp. from temp. & geopotential)
- Detect logical flaws in suites (read variable that has not been written to)
- Automated saving of physics scheme state for restarts
- Extended diagnostic output capabilities from schemes
- Creation of CCPP or NUOPC cap for physics, run either inline or as a separate component (required for UFS)
- Generation of optimized caps to dispatch physics on CPUs, GPUs, … (required for next-generation HPCs)
- …



CCPP is part of the authoritative UFS code repository

- Merged into the authoritative UFS Weather Model code repository in July 2019
- CCPP physics are bit-for-bit identical with existing physics with reproducibility compiler flags
- Scheduled for operational implementation in the GFS/GEFS 2024
- Part of the UFS Medium-Range Weather App public release in March 2020



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 Mar	GFS v15 operational Developmental schemes/suites (incl GFS v16 developmental)	SCM UFS WM UFS MRW

CCPP v4: https://dtcenter.org/ccpp

- Docs: Scientific Doc, Users Guide, Technical Documentation, FAQ
- Helpdesk: gmtb-help@ucar.edu
- UFS Users' Support Forums: https://forums.ufscommunity.org

SCM – CCPP Single Column Model
UFS WM – UFS Weather Model
UFS MRW – UFS Medium-Range Weather App

CCPP v4 supported suites

	Operational	Experimental		
	GFS_v15p2	**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	SAS	SAS	**Chikira-Sugiyama**	**Grell-Freitas**
Shallow Convection	SAS	SAS	SAS	**MYNN and GF**
Radiation	RRTMG	RRTMG	RRTMG	RRTMG
Gravity Wave Drag	uGWP	uGWP	uGWP	uGWP
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 under-development. * with SCM only

Parameterizations in ccpp-physics master

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, **RRTMGP**
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
- NOAA GSL
- NOAA PSL
- OU
- NOAA EMC

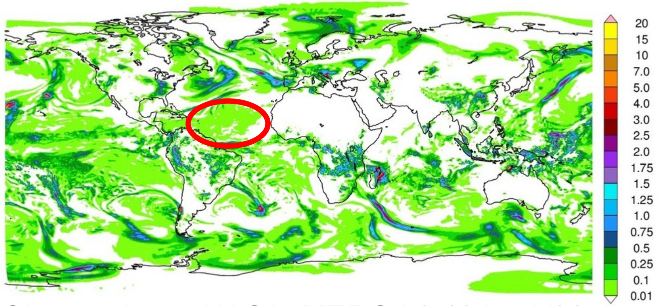
Collaboration with NRL NEPTUNE

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

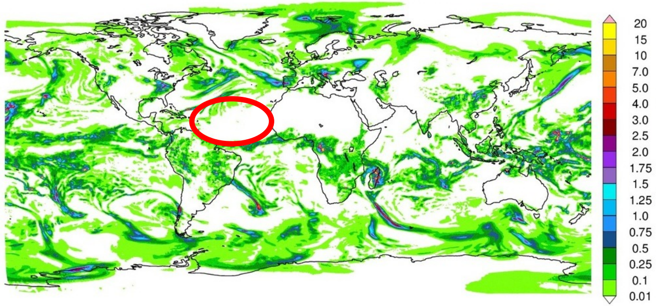
Total precipitation (explicit + parameterized) for 60-h forecast (mm/h)

Suite 4 improves drizzle bias

NEPTUNE 32 km CCPP GFSv14

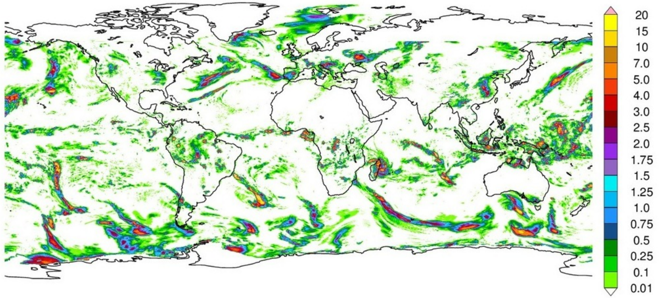


NEPTUNE 32 km CCPP Suite 4



Observations: NASA IMERG L3 10 km V06

[mm/6h]



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

One framework to rule them all (and accelerate R2O)