Community Physics Framework (and CAM Infrastructure) Update

SEWG

Steve Goldhaber,
NCAR / CGD
2019-06-19
Outline

- Brief overview of post CAM6 infrastructure redesign
- Update on the Community Physics Framework
New CAM Architecture Overview

**CESM Driver / Mediator**

Surface Coupling

Run Parameters

**CAM Driver**

**CAM (CPF) Physics**
- CAM6
- Chemistry (MICM)
- Simple Models
- Other CPF-compliant suites

**Dynamics**
- FV3
- Eulerian
- FV

**Initial Data (input) and Diagnostics (output)**

**State**

**Tendencies**

**Dynamics ↔ Physics Coupling**

**CPF Host CAP**
New CAM Architecture Design Goals

- **Design Goals:**
  - Reuse existing infrastructure where it makes sense
  - Create new infrastructure to meet future needs and/or to reduce technical debt
  - Consistent user interface (from case configuration to namelist input) from physics testbed to fully coupled simulations
  - Full physics interoperability between NCAR atmosphere models (WRF, MPAS, CAM)
  - Compatible with CMEPS

- **Initial Implementation Targets:**
  - Physics test bed (test physics schemes or suites with no dycore)
  - Simpler models (e.g., Held-Suarez, Kessler)
  - CAM6 physics with updated parameterizations
New CAM Infrastructure

- **Community Physics Framework**
  - All physics and chemistry will be run via CPF
  - CPF will use the CCPP Framework shared with NOAA and NRL

- **New physics grid and decomposition method**
  - Current serial decomposition method scales neither in time nor memory
  - New parallel decomposition algorithm will support currently-used decomposition strategies
  - Physics ‘grid’ will include ESMF mesh to allow coupling to ionosphere and ion transport models
New CAM Infrastructure (cont.)

• **Data Structures**
  - CAM state, tendency, and surface coupling structures will be redesigned to allow greater flexibility in threading models

• **Physics-Dynamics Coupling**
  - Code will be streamlined and made more modular as part of physics decomposition and data structure redesign

• **New CAM Physics sequence**
  - Realignment of physics time sequence for better compatibility with data assimilation
  - CAM history, restart and coupling data will all be ‘best’ CAM state
  - Experimental – needs to be validated before final acceptance
New CAM Infrastructure (cont.)

- **Build system**
  - Configuration and namelist build will be re-implemented using CIME tools
  - CAM namelist definition will be distributed to allow increased modularity and portability while maintaining compatibility with CIME standards

- **Testing**
  - All CAM regression tests are being ported to use CIME test system (thanks Chris!)
Update on the Community Physics Framework

• **Formal governance agreement between NCAR and NOAA**
  – CCPP Framework repository is jointly managed
  – NOAA has accepted additional requirements from NCAR

• **New, more flexible metadata standard adopted**
  – Metadata now in a separate file (same name as Fortran file with .meta suffix)
  – Uses config format
  – Facilitates flexible host model interface (variables in host-model calls listed in special metadata table)

• **NCAR implementations underway**
  – MICM (chemistry), WRF physics, and CAM
Overview of an Atmosphere Model with CPF

Atmosphere Driver

- Dycore Interface
  - Dycore
- Other Interface
  - Other
- Physics Interface
  - Physics Driver (autogenerated)

- Physics Suite 1 Cap (autogenerated)
- Physics Suite 2 Cap (autogenerated)

CCPP Physics

- PBL
- Convection
- Microphysics
- Radiation
- Chemistry
- ...

Steve Goldhaber: goldy@ucar.edu
New CPF Metadata (host)

```
[ccpp-arg-table]
  name = hello_world_sub
  type = host
[  col_start ]
  standard_name = horizontal_loop_begin
  type = integer | units = count
  dimensions = ()
[  col_end ]
  standard_name = horizontal_loop_end
  type = integer
  units = count | dimensions = ()
[  errmsg ]
  standard_name = ccpp_error_message
  long_name = Error message for error handling in CCPP
  units = 1 | dimensions = ()
  type = character | kind = len=512
[  errflg ]
  standard_name = ccpp_error_flag
  long_name = Error flag for error handling in CCPP
  units = flag
  dimensions = ()
  type = integer
```

These variables are present in the host model calls to physics. Other types are:
- **module**: Other host model data
- **scheme**: A physics scheme (subroutine)
- **ddt**: A derived data type definition
New CPF Metadata (host)

```
[ccpp-arg-table]
 name = hello_world_sub
 type = host
 col_start ]
 standard_name = horizontal_loop_begin
 type = integer | units = count
 dimensions = ()
 [ col_end ]
 standard_name = horizontal_loop_end
 type = integer
 units = count | dimensions = ()
 [ errmsg ]
 standard_name = ccpp_error_message
 long_name = Error message for error handling in CCPP
 units = 1 | dimensions = ()
 type = character | kind = len=512
 [ errflg ]
 standard_name = ccpp_error_flag
 long_name = Error flag for error handling in CCPP
 units = flag
 dimensions = ()
 type = integer
```

Local variable name begins each config section.
New CPF Metadata (host)

[ccpp-arg-table]
name = hello_world_sub
type = host

[ col_start ]
standard_name = horizontal_loop_begin
  type = integer  |  units = count
  dimensions = ()

[ col_end ]
standard_name = horizontal_loop_end
  type = integer
  units = count  |  dimensions = ()

[ errmsg ]
standard_name = ccpp_error_message
  long_name = Error message for error handling in CCPP
  units = 1  |  dimensions = ()
  type = character  |  kind = len=512

[ errflg ]
standard_name = ccpp_error_flag
  long_name = Error flag for error handling in CCPP
  units = flag
  dimensions = ()
  type = integer
[ccpp-arg-table]
  name = hello_scheme_run
  type = scheme

[ ncol ]
  standard_name = horizontal_loop_extent
  type = integer | units = count
  dimensions = ()
  intent = in

[ lev ]
  standard_name = vertical_layer_dimension
  type = integer
  units = count
  dimensions = ()
  intent = in

[ temp ]
  standard_name = potential_temperature
  units = K
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  type = real
  kind = kind_phys
  intent = out

[ . . . ]

Automatic adaptation to host loop variables
New CPF Metadata (scheme)

[ccpp-arg-table]
name = hello_scheme_run
type = scheme

[ ncol ]
standard_name = horizontal_loop_extent
type = integer | units = count
dimensions = ()
intent = in

[ lev ]
standard_name = vertical_layer_dimension
type = integer
units = count
dimensions = ()
intent = in

[ temp ]
standard_name = potential_temperature
units = K
dimensions = (horizontal_loop_extent, vertical_layer_dimension)
type = real
kind = kind_phys
intent = out

[ ... ]

Explicit dimensions allows more automatic code checking and generation
Host Model Interface

```
suite = 'hello_world_suite'
! Initialize physics suite at beginning of run
! Not threaded, I/O and MPI communication allowed
call HelloWorld_ccpp_physics_initialize(suite, errmsg, errflg)

! Timestep initialization (not threaded, I/O and MPI allowed)
call HelloWorld_ccpp_physics_timestep_initial(suite, errmsg, errflg)

! Physics run (can be in threaded region, no I/O, no MPI)
suite_part = 'physics'
call HelloWorld_ccpp_physics_run(suite, suite_part, &
   col_start, col_end, & ! Loop vars
   errmsg, errflg) ! Error check

! Timestep finalization (not threaded, I/O and MPI allowed)
call HelloWorld_ccpp_physics_timestep_final(suite, errmsg, errflg)

! Finalize physics suite at end of run
! Not threaded, I/O and MPI communication allowed
call HelloWorld_ccpp_physics_finalize(suite, errmsg, errflg)
```
Community Physics Framework: Status and Upcoming

- **CCPP Framework + GFS Physics**
  - Release 3 is in final testing for incorporation into EMC master

- **New CCPP Framework development**
  - New framework being developed by NCAR for joint use
  - Will meet advanced NCAR requirements
  - Parses new metadata format
  - Checks metadata against Fortran code and flags inconsistencies
  - Analyzes array shapes and either makes necessary adjustments or flags incompatibilities
  - Automatic memory management for suite-local variables
  - Generates flexible host model interface and code to run any number of physics suites
Thanks!

Questions?

New CAM Implementation Team:
Cheryl Craig (CPF physics porting)
Brian Eaton (New CAM physics sequence)
Chris Fisher (Testing)
Steve Goldhaber (Infrastructure)
Jesse Nusbaumer (CPF physics porting, infrastructure)
Sample Physics Suite

`<suite name="hello_world_suite" version="1.0">`
`<group name="physics">`
`  <scheme>hello_scheme</scheme>`
`  <scheme>temp_adjust</scheme>`
`</group>`
`</suite>`