SEWG: 2018-02-27

A Community Physics Framework for CAM

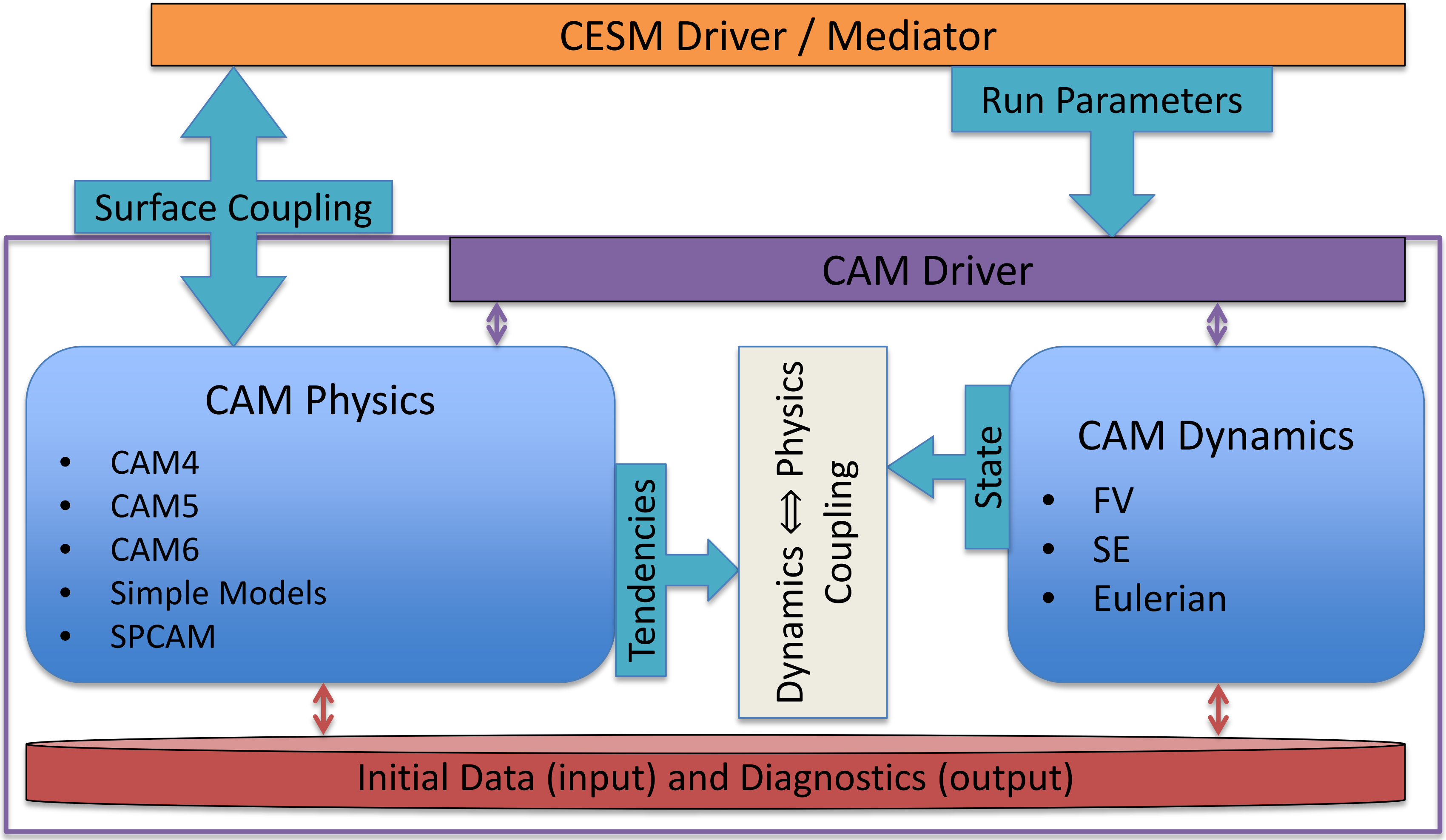
Michael Duda, Dave Gill (MMM)
Steve Goldhaber, Mariana Vertenstein, Francis Vitt (CGD)

Outline

- Brief overview of CAM6 infrastructure changes
- Some challenges ahead for CAM and atmospheric modeling
- Introduction to the Community Physics Framework

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

CAM6 Architecture Overview



Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

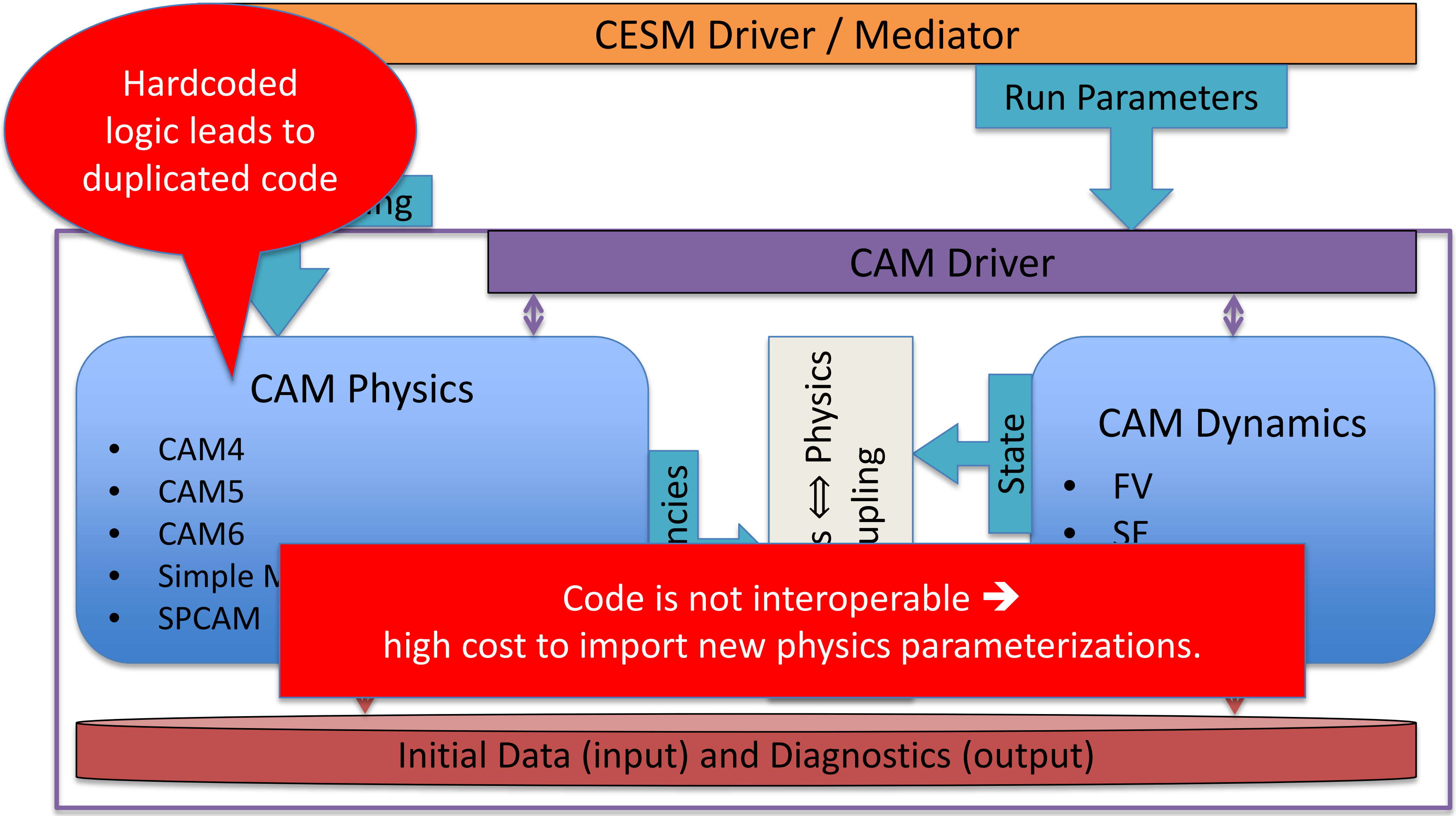
CAM6 Architecture and Infrastructure Improvements

- Better interfaces and tools for dycore development
  - Reimplementation of diagnostics library
  - Cleaner separation – no dycore-specific code in diagnostics infrastructure
  - Simple physics options and analytic DCMIP initial conditions for dycore development
- Limited capability to define new, simple physics suites
- Ability to run physics on different grid (CAM-SE only)
- Ability to run part of physics package using sub-grid-scale columns

- So what's the problem?

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

CAM6



Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Requirements for CAM Physics -- after CAM6

- Support for new physics suites (packages) while maintaining ability to run older suites
  - Interoperable development of new unified physics suites
  - Ability to continue to run mainline CAM physics suites
- Interoperability between NCAR atmosphere models (WRF, MPAS, CAM)
  - For example, run WRF physics inside CAM without any changes to parameterizations or suite definition
- Ability to run chemistry and/or physics on different grid from dynamics

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

What is wrong with what we have?

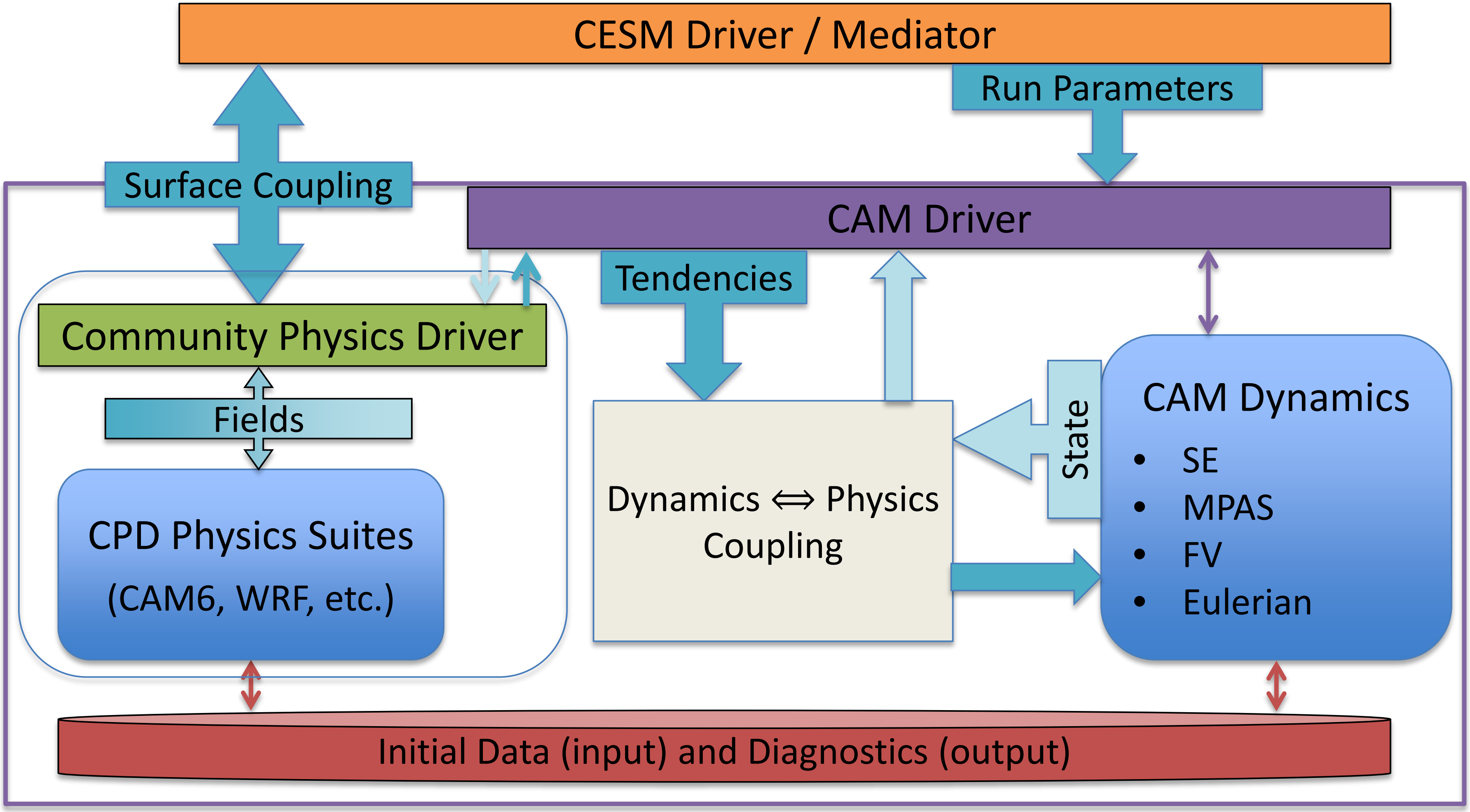
- CAM physics parameterizations depend on several CAM-specific data structures (physics_state, physics_tend, surface fields in, surface fields out, PBUF). Other models have very different state data structures.
  - This inhibits portability between models.
- physpkg (tphysbc, tphysac) logic has combined implementation of CAM3, CAM4, CAM5 & CAM6 including several options for CAM5 and CAM6.1
  - Increases difficulty in experimenting with new physics parameterizations and suites.

What is the Community Physics Driver / Framework?

- Multi-model effort to build flexible physics-package driver with a common, model-independent interface
- Replaces hardcoded, complex logic with a data-driven schedule of parameterization calls
- Handles data flow to and from host model as well as between parameterization calls
- Recently funded for implementation by CGD (CAM), MMM (WRF & MPAS), ACOM (Chemistry package)
- Goal is to also be compatible with NOAA (NGGPS, CCPP)

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

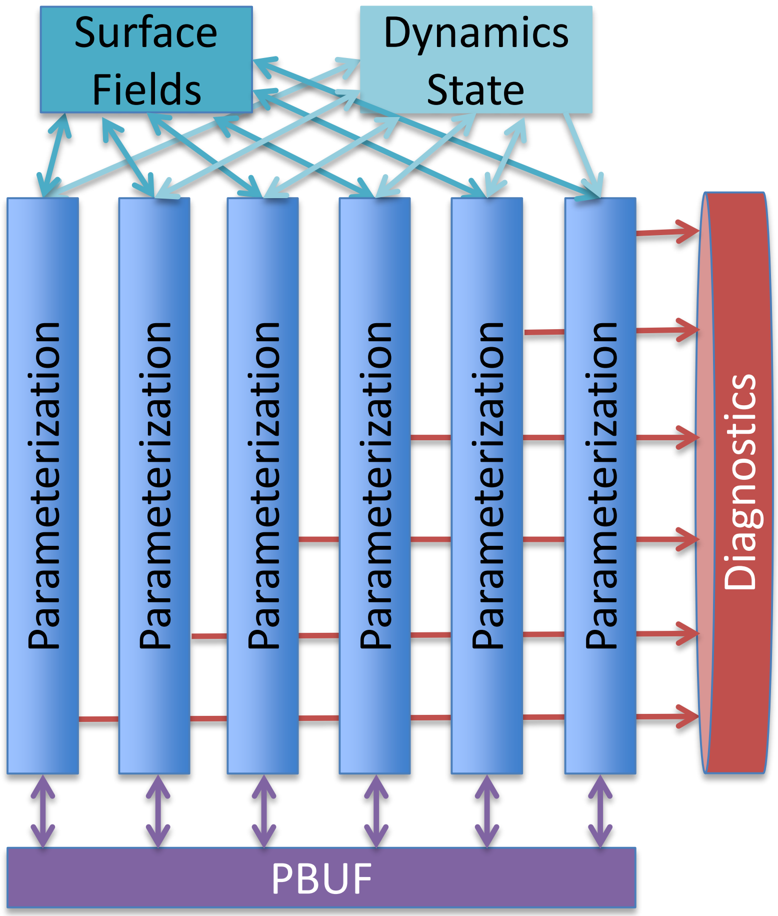
CAM6 with Community Physics Driver

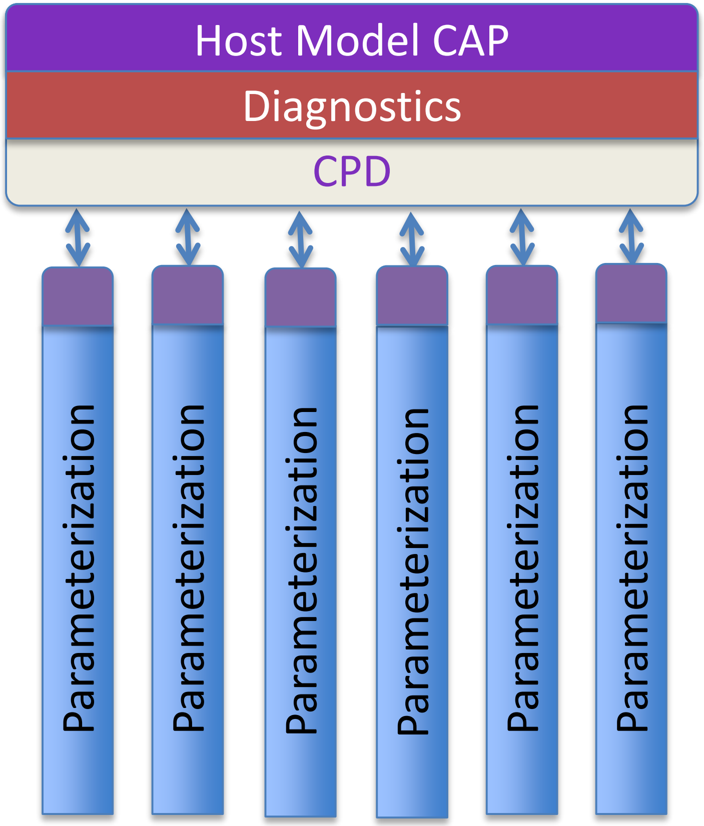


Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

CAM6 ⟹ CPF

CAM6 Physics vs. CPF





Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Parameterization

CAM6 Physics Parameterization vs. CPD

CAM6 Physics Parameterization

Gather data from state, previous tendencies, & physics buffer ⇩

Update state, tendencies, physics buffer & diagnostic output ⇧

Parameterization portable layer
(all I/O through Fortran arrays)
Examples: microphysics, cloud physics, radiation

CPD

Parameterization Cap
(Fortran code generated from Parameterization metadata)

⇩ ⇧

Parameterization portable layer
(all I/O through Fortran arrays)
Examples: microphysics, cloud physics, radiation

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Summary I

- Physics parameterizations and suites can be shared among models without modification.
- The Community Physics Framework creates a uniform data interface for parameterization inputs and outputs.
- Shared infrastructure lowers coding, testing, and maintenance costs.
- Well-documented interfaces makes it easier for the community to contribute usable parameterizations.
- Framework helps to manage model complexity.

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Parameterization CAP

```
! \section arg_table_held_suarez_1994
!| var   | standard name                | description              | units   | rank | type    |
!|-------|------------------------------|--------------------------|---------|------|---------|
!| ncol  | horizontal_loop_extent       | horizontal loop extent | index   | 0    | integer |
!| pcols | horizontal_dimension         | horizontal dimension     | index   | 0    | integer |
!| pver  | vertical_dimension           | vertical layer dim       | index   | 0    | integer |
!| delt  | time                         | physics time step        | s       | 0    | real    |
!| pmid  | air_pressure                 | midpoint pressure        | Pa      | 2    | real    |
!| u     | eastward_wind                | zonal wind speed         | m s-1   | 2    | real    |
!| v     | northward_wind               | meridional wind speed    | m s-1   | 2    | real    |
!| t     | air_temperature              | temperature              | K       | 2    | real    |
!| du    | tendency_of_eastward_wind    | zonal wind tendency      | m s-2   | 2    | real    |
!| dv    | tendency_of_northward_wind | meridional wind tend.    | m s-2   | 2    | real    |
!| ds    | tendency_of_air_temperature_due_to_radiative_heating
!                                       | heating tendency         | K s-1   | 2    | real    |
```

NCAR

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Parameterization CAP

```
! \section arg_table_held_suarez_1994
!| var   | standard name                | description             |   kind     | intent | opt.|
!|-------|------------------------------|-------------------------|------------|--------|-----|
!| ncol  | horizontal_loop_extent       | horizontal loop extent |            | in     | F   |
!| pcols | horizontal_dimension         | horizontal dimension   |            | in     | F   |
!| pver  | vertical_dimension           | vertical layer dim     |            | in     | F   |
!| delt  | time                         | physics time step      |            | in     | F   |
!| pmid  | air_pressure                 | midpoint pressure      | kind_phys | in     | F   |
!| u     | eastward_wind                | zonal wind speed       | kind_phys | in     | F   |
!| v     | northward_wind               | meridional wind speed  | kind_phys | in     | F   |
!| t     | air_temperature              | temperature            | kind_phys | in     | F   |
!| du    | tendency_of_eastward_wind    | zonal wind tendency    | kind_phys | out    | F   |
!| dv    | tendency_of_northward_wind | meridional wind tend.  | kind_phys | out    | F   |
!| ds    | tendency_of_air_temperature_due_to_radiative_heating
!                                     | heating tendency       | kind_phys | out    | F   |
```

Physics Suite

```
<suite name=”Held_Suarez">
 <init>held_suarez_init</init>
 <ipd part=“tphysbc">
  <subcycle loop="1">
   <scheme>check_energy_fix</scheme>
   <scheme>physics_update</scheme>
   <scheme>held_suarez_tend</scheme>
     ...
   <scheme>physics_update</scheme>
  </subcycle>
 </ipd>
</suite>
```


Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

What will change

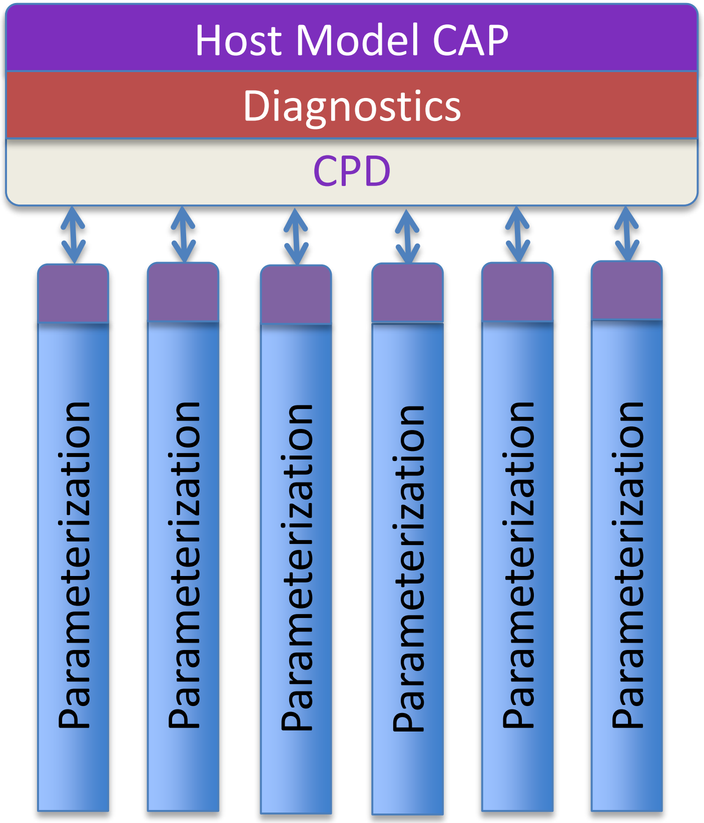
- Parameterizations will not have a custom interface (which is different for each host model). This will be replaced with a metadata header (written as a Fortran comment block) which describes the interface needs of the parameterization.
- Physics suites will not be written as hand-coded Fortran logic. This will be replaced with a dataflow description of the suite's process ordering.

What will *not* change

- The portable core of each parameterization will not have to change

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Using the CPF for parameterization development



- Parameterization developers need an easy way to run and test their code on a laptop.

- Replace Host Model CAP with standalone test driver
- Test driver provides methods for providing input data
- Test driver can capture parameterization output:
  - to a file for offline study
  - to compare to expected output

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Summary II

- CPF will improve data provenance by requiring standard metadata for every field
- The interface code and driver loop will be generated, lowering redundant code and opportunities for error
- CPF can provide development / testing framework
- Code generation can be flexible to accommodate the needs of different models.
- Framework helps to manage model complexity

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

CPF development plan

- CPF funded 2018-01-30
- Plan is to demonstrate a single physics suite focused on future, unified physics research ideas
- Software team: Cheryl Craig, Michael Duda, Dave Gill, Steve Goldhaber, Mariana Vertenstein, Francis Vitt
- Requirements definition nearly complete
- Design to begin in March
- CPF demonstrated in CAM, MPAS, and WRF by end of year

NCAR

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Questions

Thank You

- Questions?

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu

Brief CPD history to date

- Early 2016: GMTB begins work on Common Community Physics Package (CCPP).
- May 2017: Jim Hurrell asks small software team for proposal on unified infrastructure for atmospheric modeling at NCAR. Whitepaper delivered at end of July.
- November 2017: CPD section of unified infrastructure submitted as proposal for reinvestment funds.
- Q1 2018: CCPP v1 released demonstrating GFS physics running in FV3 and in GMTB single column model.

NCAR

Steve Goldhaber
Tel: 303.497.1770 | Email: goldy@ucar.edu