Interoperable Physics Driver

Goals:
- Separate atmosphere driver from atmosphere physics
- Easily add physics schemes/suites
- Reduce setup time for testing various physics combinations
- Simple, fixed interfaces
- Buy in from other modeling groups
- Experts modify code in which they are experts

Ligia Bernardet
   Tim Brown
   Laurie Carson
   Jimmy Duddia
   Grant Firl
   Dave Gill
   Josh Hacker
Our Required GeoPhys Pieces: Water, Atmosphere, Land, Sun
Software “Engineered”, not Accreted
CCPP Common Community Physics Packages
IPD Interoperable Physics Driver
AD Atmosphere Driver

Engineering for Syllables and Homonyms

Portable vs Non-potable
- Modern Fortran and C, XML lib, OS, two compilers
- Error checking for AD and CCPP consistency
- Versatile: "Yes, you can use it for that"

Parallelize vs Paralyze
- IPD is pass-through, interface only
- CCPP does not substantively change existing code

Interoperable vs Inoperable
- Design supports compliant schemes in CCPP
- Design supports general approach with AD
- Only change to physics schemes is formatting (EZ PEASEY?)

Engineered vs Injureered
- Accretion tends to be BAD in software
- Designed to be extensible and flexible, no need for user mods
- (Said in a stilted and formal voice): University researchers should not view adding schemes to the CCPP as some sort of karmatic balance for their past transgressions
Interoperable Physics Driver

aka: I P D

Purpose:

*Share different physical parameterization schemes between different models*
There are LOTS of differences between NCAR models:
- purpose
- time and spatial scales
- what is considered important
- what could be ignored
But we all agree on the need for physical parameterization schemes:
For our purposes, let's consider "physical parameterization schemes" as the code to compute something that is:

- a smaller scale than we are able to resolve
- pretty hard theoretically to figure out
- really expensive to do exactly
Generically, the "Atmosphere Driver" is the existing code in the model that is responsible for the single time-step calls to the dynamics and the physical parameterization schemes.

The "Dynamical Core" is the software responsible for vertical and horizontal differencing, including such items as advection, diffusion, pressure gradient force, Coriolis.

This code is responsible for including the (typically) RHS forcing terms to be used to compute tendencies for the fundamental variables: wind, pressure, and temperature.

Calls collections of INDEPENDENT columns, physics dependency is assumed to be 1 dimensional - in the vertical only.

The "physics driver" could be the existing model software that explicitly calls the standard types of schemes: radiation, boundary layer mixing, resolved moist processes, cumulus.

The new "physics driver":
- smaller (10-100x)
- easier to manipulate - because you don’t

---

**CCPP**

*Common Community Physics Packages*

Collection of compliant schemes

**SUITE**: set of physics schemes known to work well together

Definition files:
- suite
- each scheme in the CCPP

---

**CCPP**

*Common Community Physics Packages*

The "caps" for each of the physics schemes provide uniform access to the CCPP.

- Subroutine name from scheme definition
- Single path in and out of routine
- All gridded data allocated by AD
- Meteorological data + date/time, MPI, flags
- Does not preclude existing functionality
- --> Eventually auto-manufactured <--
IPD: What's new?

- Simple text-based definition files
- No hard-coded calls to schemes
- "Physics caps"
  - Can wrap ANYTHING: release tests, diagnostics, NaNs, positive definite
  - Cap structured for auto generation
  - Type, Kind, Dimension - overloads
- Combination of physics driver and suite definition files provide general use
- When "caps" auto generated, then no source changes required when a physics scheme changes, even the subroutine call
Status

- Multi-agency project, source of NCAR github
- Releasing version with stub physics in a single column model
- Introducing several suites
  - RAP/HRRR - next two months in single column model
  - Appropriate suites in MPAS and WRF models
- Focus is intra-processor communications, no need for "component" communications to physics
- GOAL - researchers spend hours plumbing code into chosen model, not the typical weeks