CAM6 Dynamical Core Interface

Steve Goldhaber

National Center for Atmospheric Research
CGD: Atmospheric Modeling & Predictability

2017-07-17

Evaluating FV^3 in CESM-CAM

1. Implement FV^3 as CAM dynamical core
2. Perform basic functionality tests
3. Perform 'simple' physics tests
4. Compare initialized tests with FV3GFS
5. Perform AMIP tests (tuning required)
6. Perform fully coupled tests (more tuning required)
7. Investigate FV^3 in CESM performance and scaling
8. Investigate high-top (whole atmosphere) characteristics

Implement FV^3 as CAM dynamical core

- Read dynamics namelist
- Initialize grid(s)
- Register any system variables (optional)
- Initialize dycore internal structures, decomposition, and state
- Run dynamics timestep
- Couple dynamics state to physics (different data structures and decomposition)
- Couple physics tendencies/state to dynamics after physics run

Perform basic functionality tests

Is dycore 'plugged in' correctly?

- Run FV^3 in adiabatic mode (no physics)
- Verify correct model initialization, run sequence, amd, diagnostic output
- Check for stability

Perform ‘simple’ physics tests

More ’is it plugged in correctly’ tests including coupling to parameterization suites and surface.

- Held-Suarez (1994)
- Kessler physics with baroclinic wave initial conditions
- Possibly other DCMIP experiments
- Aquaplanet tests

Perform AMIP tests

Looking for Earth

- Active land and runoff, prescribed sea ice and SST
- Run with CAM4, CAM5, and CAM6 physics suites
- Typically run from 1979 - 2004
- Comparison with other CAM dycores (FV, SE)

Perform fully coupled tests

Really take FV^3 out for a spin

- Active components (CAM, CLM, CICE, MOSART, POP2, CISM2) plus biogeochemistry
- Historical runs (e.g., 20th century beginning at 1850)
- Present day runs (beginning at 2000)
- Run for 100 - 300 years under different forcing scenarios to check for climate stability

High-Top Model

Raising the Lid

- Currently, standard CAM is a troposphere model ($\sim$2hPa, $\sim$12km)
- WACCM simulations are 'high-top' up to about 140 km ($5.1x10^{-6}\,\mathrm{hPa}$)
- WACCM-X simulations go up to about 500 km and will eventually need to couple to ionosphere models (possibly after acceptance as CAM7 dycore)
- Lots of tracers (chemical species) for stratospheric chemistry
- Derive gravity wave frontogenesis for GW parameterization

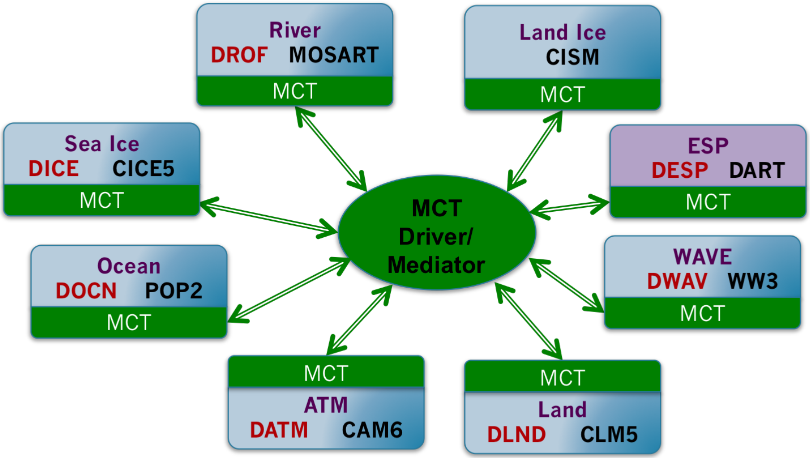
A Bit About CESM-CAM Physics

- CESM-CAM supports 9 physics packages plus 19 preset chemistry packages
- Most physics packages can be run with multiple options
- Not all combinations of chemistry and physics packages are available
- All physics and chemistry options work through the same interface so once the dycore works with one package, it should work with all*.
- *Some physics options require more information from dycore
- *The more complex packages (e.g., CAM5, CAM6) require some dycore-dependent tuning.

A Bit About CESM

- CESM2 has a coupled hub-and-spoke architecture
- CESM2 supports 7 component types
- CESM2 components can be active, data-only or not present

CESM2 Architecture



Questions?

- ??

Read dynamics namelist

- `subroutine dyn_readnl(nlfilename)`

- All namelist entries must be defined in CAM.
- Namelist entries are defined in XML files.
- A `build_namelist` phase creates appropriate namelists.
- CAM will typically read special namelist variables and then set internal dycore options (bypassing dycore namelist read(s)).

Initialize Grid(s)

- `subroutine dyn_grid_init()`

- In CAM, the dynamical core is responsible for setting up its grids early in the initialization process.
- The grid definition includes defining the parallel decomposition and grid attributes (e.g., cell area) using CAM infrastructure interfaces which allow data input and diagnostic output to function correctly.
- The dycore is also responsible for providing interfaces (listed below) which allows CAM physics to define its grid and decomposition as well as construct necessary communication interfaces.

Dynamical Core Decomposition Interfaces

- **get_block_bounds_d**: Return MPI task first and last indices used in global block ordering.
- **get_block_gcol_cnt_d**: Return number of dynamics columns in indicated block.
- **get_block_gcol_d**: Return list of dynamics column indices in given block.
- **get_block_lvl_cnt_d**: Return number of levels in indicated column. If column includes surface fields, then it is defined to also include level 0.
- **get_block_levels_d**: Return level indices in indicated column. If column includes surface fields, then it is defined to also include level 0.

Dynamical Core Decomposition Interfaces (cont.)

- **get_block_owner_d**: Return id of MPI task that "owns" the indicated block.
- **get_gcol_block_cnt_d**: Return number of blocks contain data for the vertical column with the given global column index.
- **get_gcol_block_d**: Return global block index and local column index for global column index.

Dynamical Core Decomposition Interfaces (cont.)

- **get_horiz_grid_dim_d**: Returns declared horizontal dimensions of computational grid. Note that global column ordering is assumed to be compatible with the first dimension major ordering of the 2D array. For unstructured grids, the first dimension is the number of grid columns while the second is one.
- **get_horiz_grid_d**: Return latitude and longitude (in radians), column surface area (in radians squared) and surface integration weights for global column indices that will be used by CAM physics (i.e., may be different than used internally by dynamical core).
- **physgrid_copy_attributes_d**: See CAM Grid Infrastructure

Register System Variables

- `subroutine dyn_register()`

- CAM has a data structure that allows sharing data between physics parameterizations and between physics and dynamics
- This structure is called the physics buffer or pbuf
- The pbuf is used by some dycores to diagnose entities which are not part of the normal dynamics state (e.g., gravity waves)

Initialize Dycore Internal Structures

- `subroutine dyn_init(dyn_in, dyn_out)`
- `subroutine stepon_init(dyn_in, dyn_out)`

- `dyn_in` is a dycore-specific type. It's memory is managed by CAM and is used to supply inputs to the dycore.
- `dyn_out` is a dycore-specific type. It's memory is managed by CAM and is used to supply outputs from the dycore.
- Initial dycore data is read during dyn_init. CAM provides pointers to the initial data and topography files.
- Diagnostic (history) fields are also registered during dyn_init.
- dyn_init is not called when restarting the model. Initialzation which should occur regardless of model start type should be in stepon_init.

Run Dynamics Timestep

- `subroutine stepon_run1(dtime_out, phys_state, phys_tend, pbuf2d, dyn_in, dyn_out)`
- `subroutine stepon_run2(phys_state, phys_tend, dyn_in, dyn_out)`
- `subroutine stepon_run3(dtime, cam_out, phys_state, dyn_in, dyn_out )`

- CAM calls the dynamics run via three interfaces, stepon_run1, stepon_run2, and stepon_run3. The dycore can flexibly schedule its run step as well as coupling to CAM physics via these interfaces.
- stepon_run1 is called before CAM physics is called.
- stepon_run2 is called after CAM physics (but perhaps before other entities, such as ionosphere transport).
- stepon_run3 is called after all other processes but before the end of the CAM time step.

Couple Dynamics State to Physics

- The dynamical core must update the CAM physics state during one of its three interface calls.
- If the physics decomposition has a local map (i.e., dynamics columns and physics columns reside on the same MPI tasks), the transfer is through indirect addressing.
- If the physics decompisition is non-local, communication is handled through interfaces provided by CAM physics.
- The dycore provides, column averaged state quantities for surface pressure, surface geopotential, wind, vertical pressure velocity (Pa/s), temperature, inverse exner function with respect to surface pressure, interface pressure (Pa), log of the interface pressure, and, tracers.
- The dynamics data typically comes from the `dyn_out` structure.

Couple Physics Tendencies/State to Dynamics

- The dynamical core will update its internal state using data from CAM physics during one of its three interface calls.
- If the physics decomposition has a local map (i.e., dynamics columns and physics columns reside on the same MPI tasks), the transfer is through indirect addressing.
- If the physics decompisition is non-local, communication is handled through interfaces provided by CAM physics.
- CAM physics provides tendencies for wind, temperature, total energy flux at boundary, total water flux at boundary, and heat flux at boundary. An updated physics state is also available.
- The dynamics data typically goes to the `dyn_in` structure.

Other Requirements

- Dynamical core must be able to run on a subset of the total CAM MPI task set (communicator).
- CAM must be able to set a maximum thread count which is respected by the dynamical core.

CAM Grid Infrastructure

- `function horiz_coord_create(name, dimname, dimsize, long_name, units, lbound, ubound, values, map, bnds)`: Define a distributed coordinate (e.g., longitude). These coordinates are used in distributed grid definitions.
- `subroutine cam_grid_register(name, id, lat_coord, lon_coord, map, unstruct, block_indexed, zonal_grid, src_in, dest_in)`: Define a distributed grid. These grids are used to effect parallel data input and diagnostic output.
- `subroutine cam_grid_attribute_register(gridname, name, long_name, val)`: Define a grid attribute. These attributes are output as special fields and/or metadata in diagnostic output files.

CAM Grid Infrastructure (cont.)

- `subroutine cam_grid_attribute_copy(src_grid, dest_grid, attribute_name)`: Ensure that a shared attribute is output exactly once in a diagnostic file.
- `subroutine physgrid_copy_attributes_d( gridname, grid_attribute_names)`: Specify attributes which should be output along with phyics diagnostics.
- Note that the interface to `cam_grid_attribute_register` varies depending on whether the attribute is scalar (e.g., a global property such as the number of elements along a cube edge) or vector (e.g., the area of each grid cell).

CAM timestep for CAM6 dycores

Note this is time based, not run-step based.

core	step1	step2	step3	step4	step5	step6
eul	tphysbc	coupler	tphysac	dynrun	output	
fv	dynrun	tphysbc	coupler	tphysac	dyn diags	output
se	tphysbc	coupler	tphysac	dynrun	output	

Notes:

- **output** refers to history and/or restart output
- **dynrun** refers to the dycore run step
- **dyn diags** refers to special FV dynamics diagnostics calls

CAM6 dycore stepon_run<n> functionality

core	stepon_run1	stepon_run2	stepon_run3
eul	dyn IC d_p_coupling	p_d_coupling	dynrun
fv	dyn IC dynrun d_p_coupling	p_d_coupling	dyn diags
se	d_p_coupling dyn IC	p_d_coupling	dynrun

Notes:

- **dynrun** refers to the dycore run step
- **dyn diags** refers to special FV dynamics diagnostics calls
- **dyn IC** refers to output of dynamics state as an initial conditions file.

Physics Options for Development and Testing
Simple through Fully Coupled

Available option	**CIME compset indicator(s)**
Adiabatic (no physics)	FADIAB, FDABIP04
Held-Suarez	FHS94
Kessler physics	FKESSLER
Aquaplanet	QPC4, QPC5, QPC6
AMIP	FAMIP, FAMIPC5, FAMIPC5L60, etc.
Fully Coupled	B1850, BHIST, BC5L45BGC, etc.