Common Community Physics Package (CCPP): Requirements for supported schemes/suites and driver layer

Developed by the Global Model Test Bed

Current 9/14/2017

Points of contact: Ligia Bernardet (DTC/GMTB at ESRL/GSD; Ligia.Bernardet@noaa.gov),
Grant Firl (DTC/GMTB at NCAR/JNTP); grantf@ucar.edu)

Revisions:

11/11/2015 - Ligia Bernardet

Document created and sent to EMC and NGGPS PO for comments

01/25/2016 - Ligia Bernardet

Included definition of suite by GMTB

- Revised Figure 2 and text to represent that driver and pre/post parameterization interface does not have to be called multiple times for a single suite (as per EMC input) 01/26/2016 - Ligia Bernardet

Revised Figure 2 to be consistent with NGGPS Overarching System (OAS) Team vision for model components and mediators

02/16/2016 - Josh Hacker

Added requirements following meeting with Mark Iredell (EMC)

08/22/2016 - Laurie Carson

Reformatted table of requirements to comply with format put forth by the NGGPS Overarching System Team (described here)

Striked through table of coding standards, since coding standards are now described here

08/15/2017 Ligia Bernardet

Deleted introduction since these topics are now covered in the CCPP design Document Deleted status column since it pertained to IPD v2.

Deleted coding standards, since they are now described here.

09/14/2017 - Ligia Bernardet

Made changes after feedback received from NGGPS and EMC (Farida, Vijay, and

Moorthi)

- Added source of requirements

09/18/2017 - Ligia Bernardet, Grant Firl, Laurie Carson, Dave Gill

Additional changes and cleanup after feedback received from NGGPS and EMC (Farida, Vijay, and Moorthi)

An initial proposal on requirements for interoperability of physical parameterizations and associated driver was drafted by GMTB and sent to EMC and the NGGPS Program Office in November 2015. This proposal incorporated some materials produced previously by the Earth System Prediction Capability Physics Interoperability ESPC PI) team. Requirements evolved over the following year with input provided by EMC, the NGGPS Program Office, the NGGPS Physics Team, the EMC Strategic Implementation Plan (SIP) Physics Working Group, and the ESPC PI team, with the most substantial revision being done in February 2016 with input from EMC. These requirements are consistent with the *Project 3: Collaborative framework for developing physical parameterizations* listed in the document *EMC Strategic Implementation Plan (SIP) for Evolution of NGGPS to a National Unified Modeling System*.

Driver Requirements

Classification used in tables below

D= dycore and model application development

O = operations

P = parameterization development

U = model user

ID	Class	Type	Item
D1	D P	Requirement	
D2	D		

ment

The driver shall allow parameterizations to be agnostic of host application.

Require The driver shall provide

an easily configurable entry point for passing information to/from

physics

parameterizations.

Require The driver shall be

D3

P

ment

D4

DPU

expandable to include new variables.

Require The driver shall provide

the ability to select different

ment

parameterizations of the same category via an

Reason

Well-established convention facilitates data mapping.

Enhances portability and simplifies the

interface for community contributions.

Newly added parameterizations may

need information not already provided by host application.

Provides flexibility and ease-of-use; allows direct comparison between schemes, possibly within an existing suite.

D5

D

DPU

D6

DPU

DP

Require
ment

Require

ment

ment

external option selection.

The driver shall allow parameterizations to be used as suites or be selected individually.

The driver shall allow the order and frequency of calls to individual parameterizations to be configurable.

Require The driver shall provide the capability to share the same instance of physical constants with all model components (host application and parameterizations).

Requirement: The driver shall include documentation including references, functional descriptions of code, guidance for how to call parameterizations as suites or individually in any order, and guidance on how to connect new parameterizations or host applications.

Requirement: The driver shall be developed using modern and robust coding standards balancing portability, computational performance, usability, maintainability, and flexibility, and follow coding guidelines listed here.

Suites are useful in both an operational and research environment; the ability to choose individual schemes is important for testing and development.

Allows for sensitivity testing of different physics configurations.

Maintains consistency among model components.

Community code should be well-documented for users and developers.

Following Kalnay rules

Require The driver shall provide Offline mode allows for sensitivity and

D7

D8
DOPU

D9

D

O

P

D10

P

D11

D12

D13

D14

D15

DOP

PU

DP

P

D

P

PU

DP

ment

Require ment

Require ment

ment

the ability to drive parameterizations or suites in "offline mode".

The driver shall provide the ability to pass arbitrary "chunks" of input variables to parameterizations.

The driver shall provide the ability to deliver variables computed by, or for, use within any parameterization for diagnostic purposes to the model I/O component.

Require The driver shall provide the ability to deliver variables computed by, or for, use within any parameterization to external models.

ment

Require The driver shall not modify answers produced by the parameterizations.

Require The driver shall allow

ment

run-time specification of parameters (possibly greater than 1D).

Require The driver shall ensure

that more than one code
variable with the

P

ment

DOP

same physical meaning
but different names cannot
exist.

Require The driver code

management and support shall be

ment

designed so community
scientists can use and propose
contributions.

process-based studies; removes response from other components to focus on impact from
parameterizations.

Follows modified Kalnay rules 6, 7. Increases computational performance.

Important for testing, development,
and evaluation.

Facilitates consistency with other Earth System models (e.g. ability to share roughness length
between parameterizations and LSM). Note
that coupling to external models will be
done at the host application level.

Eliminates inadvertent errors.

Allows rapid tuning and sensitivity
experimentation.

Minimizes ambiguity.

Meets the NCEP goals for community modeling and enhances R20.

D16 D

D17 D

CCPP Requirements

ID Class Type

Item

C1 P Require ment

C2 D

The CCPP schemes shall conform to standard variables.

Require The CCPP shall allow multiple

ment

parameterizations of each category to coexist in the CCPP.

Reason

Common variables facilitate scheme portability.

The CCPP can support all NCEP needs (including research and development).

C3

C4

C5

C6

PU

U

PU

PU

DPU

Require ment

ment

Transparent criteria shall be used to guide number and choice of parameterizations included in CCPP.

Require The CCPP schemes shall have standard and documented testing procedures and metrics applied by all physics developers.

Require The CCPP schemes

ment

shall have standard and documented observation and model databases for testing.

ment

Require The CCPP schemes shall permit parameterizations to expose all tunable parameters

A Change Review Board reviews test results and ensures quality control of parameterizations and has authority over portfolio of supported parameterizations. Maintenance is kept to a manageable level while focusing on operational and research applications.

The Change Review Board defines minimum testing procedures and metrics. This may include specific codes/tools to be employed in the test harness.

Both observation and model-generated datasets need to be selected and available for testing. This ensures that the Change Review Board has material that is easy to judge. Tools to subset or process data may be part of this, as necessary.

Tunable aspects of parameterizations will be configurable by run-time settings, e.g. Fortran namelists, allowing a single software instance of a parameterization to satisfy all foreseeable models and

C7

DP

P

Requir
e
ment

The CCPP schemes

C8

P

Requir
e
ment

C9

DPU

C10 D
DPU
shall permit a capability to share same instance of physical constants with all model components (host application and parameterizations).

The CCPP schemes code management shall be designed so community scientists can use and propose contributions

Require The CCPP schemes
ment
shall have documentation including references, functional descriptions of code, information on inputs/outputs to parameterizations, and guidance on how to add new parameterizations.

Require The CCPP schemes
ment
shall employ modern and robust coding standards supporting portability, computational performance, usability, maintainability, and flexibility and follow coding guidelines

listed
in the Coding Standards.

applications.

Maintains consistency among model components.

Meets the NCEP goals for community modeling and enhances R20.

Community code should be well-documented for users and developers.

Follows modified Kalnay rules.

Host Application Cap Requirements

ID

Class Type

Item

H1

D

Require ment

H2 D

Require

ment

H3

D

Requirement

The Host Application Cap shall be written in Fortran.

Reason

The CCPP Layer assumes various C to Fortran constructs, and assumes Fortran module constructs.

The Host Application Cap (or the upstream calling application) shall manage all variables that are arguments to individual parameterization schemes. Manage includes, but is not limited to, allocation, distributed communication, initialization, I/O, correct numbers of scalars, and metadata.

For consistency, there cannot be some argument variables to physical parameterizations that the scheme chooses to manage and others that the scheme allows the Host Application Cap (or the upstream calling application) to manage. Since most physical parameterizations do not have enough information to make decisions on variable management, it is the task of the upstream systems to manage the variables in the routine argument lists.

The Host Application Cap (or the upstream calling application) shall perform all required I/O, except for a few notable cases. Individual physical parameterization schemes may do some or all of the following input tasks:

look-up table initializations, table entries. The physical parameterization schemes shall not input gridded data that has been horizontally decomposed. The physical parameterization schemes shall not output data to disk directly. The physical parameterization schemes shall not

The fundamental purpose of a physical parameterization scheme is to compute some specific physical process. The more the parameterizations stay aligned with this standard, the more portable the schemes are. Allowing complicated I/O systems to be introduced into parameterizations reduces the chance at simple portability of those schemes with a new Host Application.

H4

D

H5
D

H6
D
Require
ment

Require

ment

Requir
e

ment

require external I/O libraries.

The Host Application Cap (or the upstream calling application) shall handle all required distributed memory processing for decomposed arrays, if any is required. Upon entry into each parameterization scheme, each input field in the argument list is assumed to be the correct value to use. The parameterizations do not include logic or machinery to determine or act on information for neighboring grid locations. The parameterization schemes may broadcast look-up table information from the master task to the rest of the communicator

The Host Application Cap (or the upstream calling application) shall handle all processing that requires that the parameterizations be computed on a grid or resolution different than the upstream calling application.

The Host Application Cap (or the upstream calling application) shall handle all processing that requires that the parameterizations be run concurrently.

The physical parameterization schemes do not carry information that allows them to determine neighboring grid columns. The parameterizations are all assumed to be 1d columns of independent data, though for performance purposes, blocks of those 1d columns may be bundled into arrays.

For ease of portability, the parameterizations only know the arrays provided in the argument lists, the provided array sizes, and the computational extent for each of the arrays. The parameterizations are not aware if the incoming arrays are indeed the original size of the grid that the Host Application is running.

The physical parameterizations have no information about the sequential nature of their own processing, other than the list of arguments defined as either input or output. Because all information for a parameterization comes through the argument list, the parameterization is well suited to being insulated from external processing techniques. The upstream

H8

D

Require

ment

Require

ment

The Host Application Cap (or the upstream **calling** application) **shall have** no OpenMP **parallel** regions.

The Host Application Cap shall **use** Fortran **array** syntax that is valid for **arguments that have** an explicit interface.

calling application has the **necessary** software infrastructure tools **to** set up concurrent **parameterization processing**.

The cap for the physical parameterization schemes is **automatically** manufactured. **For** timing performance and portability, all OpenMP threading for a particular scheme is controlled by each scheme's

cap.

Taking advantage of argument **mismatch** for type, kind, and **rank** is only available with explicit interfaces. Given that **the** purpose **of** the effort is to include **additional** schemes, allowing the compilers to find argument mismatches is **a** benefit.

Approval/Signature

Mikel Jama

NWS EMC Director

MIKE FARRAR

Date

10/19/
2012

Date

10/1
7/20
17

NGGPS **Program**
Manager

FREDERICK TOEPFER