Subcolumns in CAM

Cheryl Craig and Steve Goldhaber
with
Andrew Gettelman,Julio Bacmeister, and Brian Eaton

National Center for Atmospheric Research

CESM Breckenridge Workshop
June 20, 2013

Outline

The CAM subcolumn infrastructure provides the ability to create fields with several elements within a single grid column.

- Rationale for subcolumns
- Introduction to subcolumn usage in CAM physics
- Introduction to infrastructure changes made to support subcolumns
- Current subcolumn development efforts

Subcolumns in CAM

- Rationale for subcolumns
- Introduction to subcolumn usage in CAM physics
- Introduction to infrastructure changes made to support subcolumns
- Current subcolumn development efforts

Rationale for subcolumns

- Allow parameterizations finer granularity than the traditional grid box
- Use a statistical approach to sample within a single column or physically sub-divide the column
- Don't need to increase resolution on entire model to study one parameterization in detail

(Satellite view of earth)

Standardizing subcolumns

- Subcolumns already exist in CAM – radiation, SPCAM branch, etc.
- Each implementation is specialized – “shoehorned” into old existing static structures
- Not able to share subcolumns between parameterizations due to unique implementations

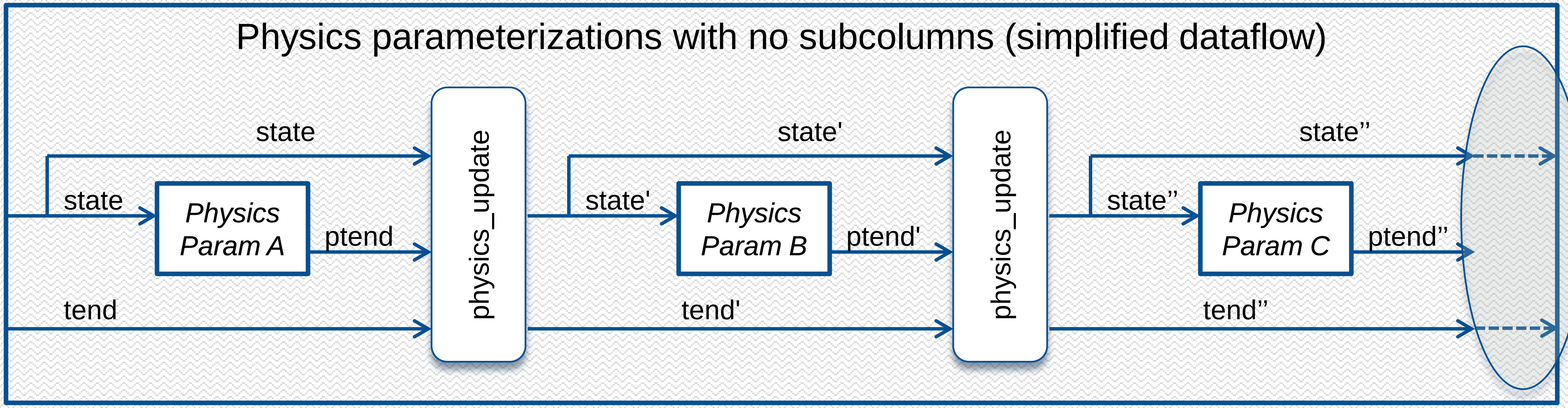
Subcolumns in CAM

- Rationale for subcolumns
- Introduction to subcolumn usage in CAM physics
- Introduction to infrastructure changes made to support subcolumns
- Current subcolumn development efforts

The CAM Physics Package

- A collection of parameterized physics or chemistry processes (usually called parameterizations).
- A parameterization is handed a ‘chunk’ of grid columns to work on in one call.
- Each parameterization takes the model state and returns changes to the model state (‘tendencies’).
- subcol_gen creates a state with subcolumns which a subcolumn-aware parameterization will use to create a tendency with subcolumns.

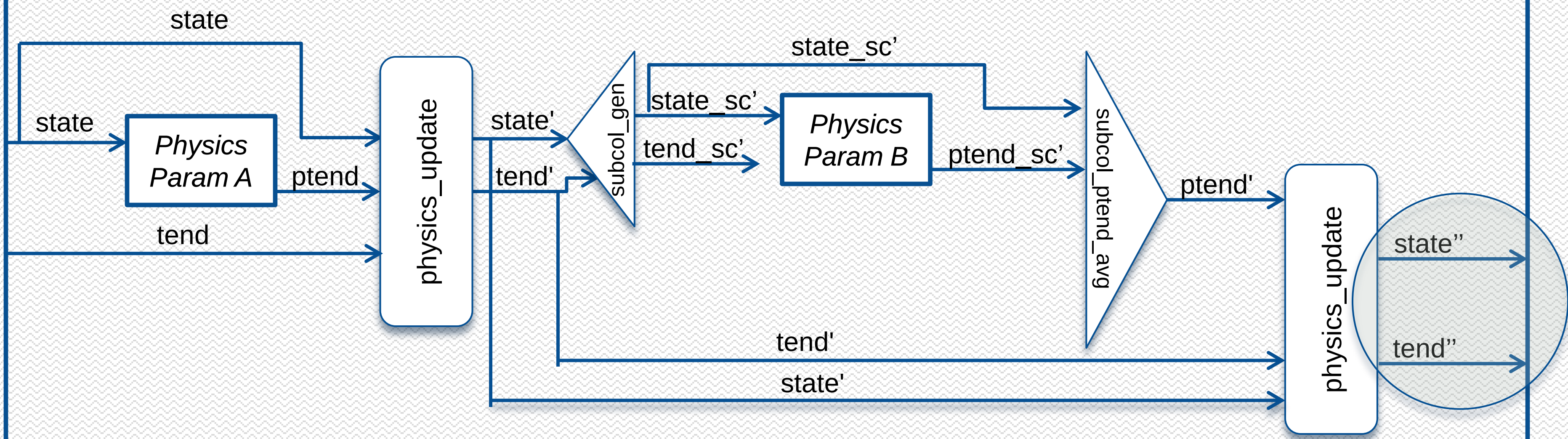
The CAM Physics Package



- state: The CAM physics state
- tend: The changes to variables used by the dynamical core (e.g., $\dot{u}, \dot{v}, \dot{t}$)
- ptend: The changes to state variables calculated by a parameterization

The CAM Physics Package

Adding subcolumn support (still simplified dataflow)



subcol_gen creates a state and tend with subcolumns

Physics parameterization B uses subcolumns

Parameterization tendency is averaged with subcol_ptend_avg

Physics Parameterization w/ Subcolumns

(subcolumn code not on CAM trunk)

- Retrieve needed fields from the physics buffer: pbuf_get_field (using the col_type optional input)
- Copy fields which are not defined on subcolumns: subcol_field_copy
- Operate on fields (math; involves loop over state%ncol)
- Update parameterization tendency fields
- Average subcolumn fields: subcol_field_avg
- Output grid fields to history: outfld
- Output subcolumn fields to history: subcol_outfld

Subcolumns in CAM

- Rationale for subcolumns
- Introduction to subcolumn usage in CAM physics
- Introduction to infrastructure changes made to support subcolumns
- Current subcolumn development efforts

Features of changes

- state%ncol is still the number of columns to loop over inside parameterizations
  - underlying physics parameterizations do not require code modifications - work the same whether grid or subcolumns
- Variable number of subcolumns per grid column
- Grid and/or subcolumn fields only allocated as requested
- state (for grid) and state_sc (for subcolumns) may both exist at same time – synchronization occurs between parameterizations
- Several subcolumn generators may exist in CAM, but only one will be used per run

Dynamic state/tend/ptend changes

- Variables unchanged within new subcolumn framework
  - pcols - maximum number of grid columns
  - state%ncol - number of columns to loop over inside parameterizations
    - may be larger than pcols is using subcolumns
- New implementation to support subcolumns
  - psubcols - maximum number of subcolumns = 1 for grid
  - state%psetcols - maximum number of total columns
    - whether using grid or subcolumns = pcols*psubcols
    - replaces pcols in a lot of places
  - **state/tend/ptend are now dynamically allocated**

New Fields to support subcolumns

Subcolumnized data

Internal Storage Layout - compressed

Grid Columns	1		2	3			4	5	6
	⬇		⬇	⬇			⬇	⬇	
Subcolumns	X	X	X	X	X	X	X	X	

Conceptual Layout

Subcolumns \ Grid columns	1	2	3	4	5	6
			X			
	X		X			
	X	X	X	X	X	

X = Data locations

pcols = 6 ngrdcol = 5
psubcols = 4 nsubcol(6) = (2,1,3,1,1,0)
psetcols = 24 ncol = 8

indcol(24) = (1,1,2,3,3,3,4,5,0,...)

ppgrid_parameter	state_variables

Grid data

Grid columns					
1	2	3	4	5	6
X	X	X	X	X	

pcols = 6 ngrdcol = 5
psubcols = 1 nsubcol(6) = (1,1,1,1,1,0)
psetcols = 6 ncol = 5

indcol(6) = (1,2,3,4,5,0)

Physics buffer (pbuf) changes

pbuf structure contains:

(buffer_field_type):: bfg%data – holds grid data

(buffer_field_type):: bfg_sc%data – holds subcolumn data – NEW

- bfg%data and bfg_sc%data are only allocated as requested
- Each physics buffer field can have grid-only, subcolumn-only or both grid and subcolumn data

New control parameters

col_type: int - 0=grid, 1=subcolumns

Used to identify WHICH field (used in pbuf_get_field for example)

grid_type: int(bit_field_kind) - each bit is turned on/off to indicate which field(s) are required

1	**1**	**0**	**0**	**0**	**…**	**Grid and subcolumn**
1	**0**	**0**	**0**	**0**	**…**	**Grid only**
0	**1**	**0**	**0**	**0**	**…**	**Subcolumn only**

Used to identify ALL fields which are currently turned on (used in pbuf_add_field for example)

Subcolumns in CAM

- Rationale for subcolumns
- Introduction to subcolumn usage in CAM physics
- Introduction to infrastructure changes made to support subcolumns
- Current subcolumn development efforts

Status

- Currently being implemented in CAM
- Infrastructure changes committed to CAM development trunk:
  - Part of CAM trunk since cam5_3_03
  - Note that this infrastructure is not in CESM 1.2 (except for the dynamic allocation of state/tend/ptend which was committed in cam5_2_09).
- Subcolumn support is being implemented in CAM microphysics (Gettelman, Craig)

Subcolumn Schemes

- Subcolumn generation/averaging is still under development (2 groups are currently prototyping subcolumn schemes).
  - SILHS and CLUBB: Improve the representation of sub-grid variability using an ensemble approach to microphysics driven by sampling the PDF output from CLUBB (Thayer-Calder, Larson, Bacmeister, Gettelman).
  - Cloud Model: Study extended cloud dynamics by creating subgrid-scale cloud objects which persist across time steps (Bacmeister, Goldhaber).

Acknowledgements

- We gratefully acknowledge useful discussions with William Collins, Steve Ghan, Phil Rasch, Vincent Larson, Cecile Hannay, Kate Thayer-Calder, Charles Bardeen, Chris Fischer, Joe McInerney, Sean Santos, John Truesdale, Jim Edwards, Mariana Vertenstein, and Francis Vitt.
- This work has been supported by the National Science Foundation and by the US DOE Advanced Scientific Computing Research and Biological and Environmental Research Programs.

Summary

- CAM now has a subcolumn infrastructure which introduces a standard method for working with a variable number of subcolumns.
- We are happy to support efforts to use subcolumns in CAM.

IF YOU PLAN ON IMPLEMENTING SUBCOLUMNS:
CONTACT US – THIS IS A WORK IN PROGRESS

Cheryl Craig: cacraig@ucar.edu
Steve Goldhaber: goldy@ucar.edu
Andrew Gettelman: andrew@ucar.edu