Variable Recycling in GEM:

Technical Description and Preliminary Results

11 September 2013

Presented by: Ron McTaggart-Cowan and Ayrton Zadra

Base on the work of (alphabetical):
Mark Buehner
Stephane Chamberland
Michel Desgagne
Vivian Lee
Michel Roch
Paul Vaillancourt

Overview

- What does "variable recycling" mean?
- Types of variables in GEM
- Recycling history: "Physics Bus" recycling
- What is Step-0 physics?
- Selective physics recycling
- Dynamics recycling
- Framework for testing recycling impact
- Preliminary results and recommendations for dynamics and selective physics recycling

Definitions

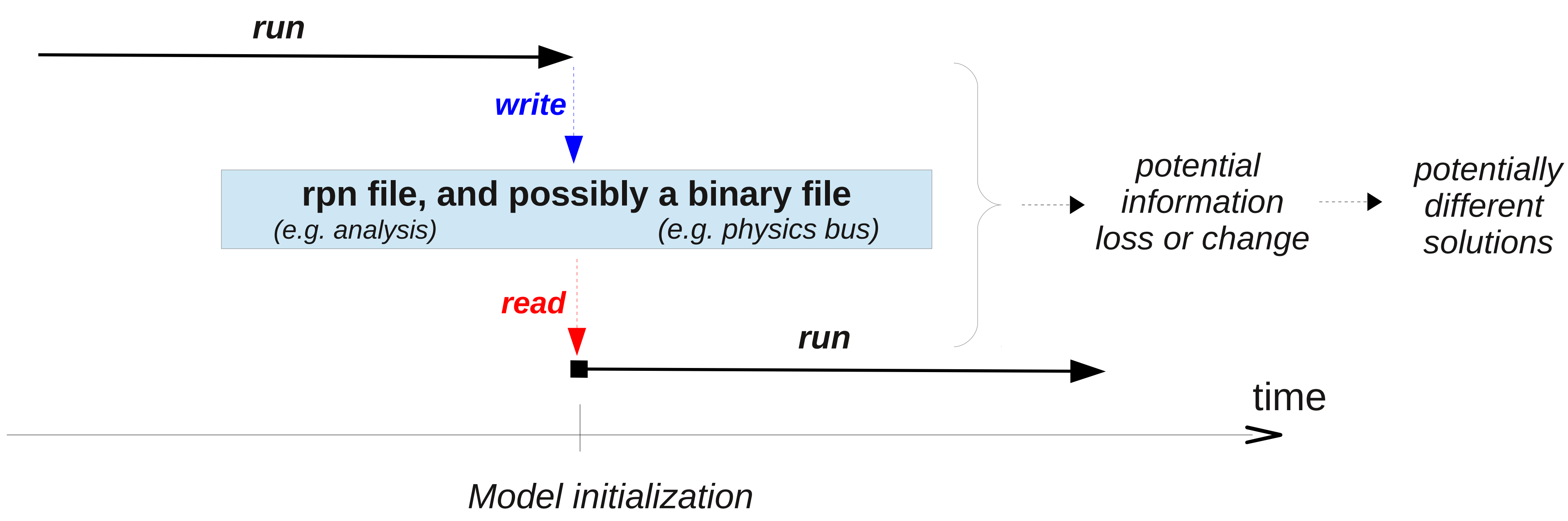
- What does “variable recycling” mean?

A certain minimum number of fields are required to initialize the model atmosphere (temperature, surface pressure, winds …) while the rest of the model state is filled with diagnostic values. Instead of using these diagnostic values, fields can be read into the model based on a previous integration. This is called “variable recycling” since information from an earlier run is recycled in order to avoid a diagnostic estimate.

- What is the difference between “recycling” and “restarting”?

The model's “restart” capability has existed for many years is not at all related to variable recycling. Restarts are used to avoid queue limits of integration time (currently 3h): a full image of the model's memory is written to disk at the end of a restart “time slice” and is read back into memory at the beginning of the next slice. These raw binary files serve very little purpose other than to restart these time slice “clones”. Restarting in GEM is **perfect**, in that an integration split over many time slices yields bitwise identical results to the same integration run in once slice.

Variable Recycling



In this schematic, the model is initialized from data generated by a previous integration. This is usually provided in the form of an RPN standard file (FST) that contains at least the mandatory variables required to start the model.

Information is lost when the model is re-initialized, since numerous diagnostic calculations are used to populate the initial model state (particularly in the physics): this data was available in the previous integration, but is lost when the FST output file is generated. Additional loss of precision may arise from FST storage depending on the number of bits allotted to each record.

Types of Variables in GEM

- In the standard (uncoupled, no chemistry...) version of GEM, there are two primary types of variables:
  - Dynamics variables belong to the model's adiabatic core
  - Physics variables belong to the RPN physics package, and are arranged in buses, including:
    - Permanent Bus: physical quantities kept from one timestep to the next (e.g. time-evolving quantities). Also called the “physics bus”.
    - Volatile Bus: physical quantities that persist only during the current physics timestep (e.g. values transferred between parameterizations)
- One class of variables is shared between the dynamics and physics: tracer variables are read and managed by the dynamics, but are of primary relevance to the physics (loaded tracers also affect the dynamics through density)
- For perfect recycling, all physics, dynamics and tracer fields must be recycled in their native forms (i.e. without derived calculation and at full precision)

Physics Bus Recycling

- Perfection in physics recycling is achieved using physics (permanent) bus recycling
- This strategy is employed in restarts, and can also be used for recycling via the GEM_busper option to the Um_* system
- An image of busper is written as a Fortran binary file to disk by the preliminary integration, then read back into memory by the second integration
- The busper reading step replaces the Step-0 physics operation, since the full physical state retained from step-to-step (busper) is already available, such that no initial diagnostics are required

Physics Bus Recycling

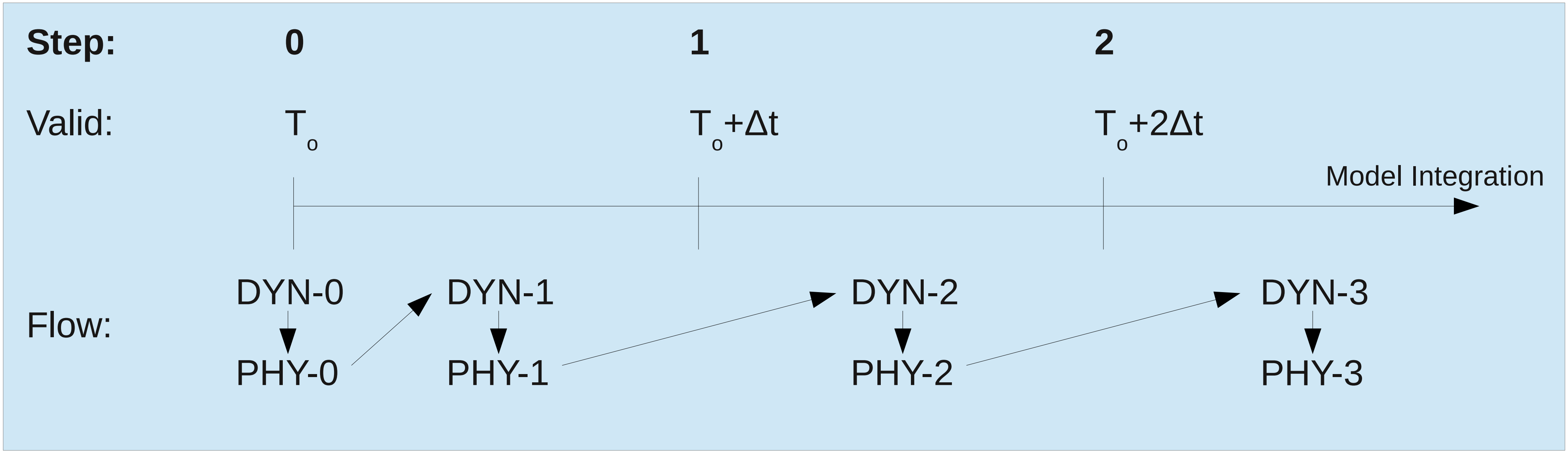
- In reality, using full bus recycling is impractical since it is non-selective: all variables are recycled, including (for example) the surface fields for which a new analysis is available
- There are also important technical limitations:
  - Cannot change physics configurations tests
  - Cannot change the bus structure for debugging or tests
  - Cannot change the model MPI topology for debugging or rerunning (imagine a future machine on which the topology cannot support the grid size used to produce busper)
  - Full busper is very large to hold onto at each analysis time (~7G for a 25 km GU model)
  - Binary format of data makes the data more fragile, particularly for archives across Endianness changes
- These limitations grow more severe the longer the data has been archived for (i.e. imagine having these restrictions on rerunning a case from 2001).

Step-0 Physics

An important element of physics bus recycling seems to be the suppression of step-0 physics... but what does that mean exactly?

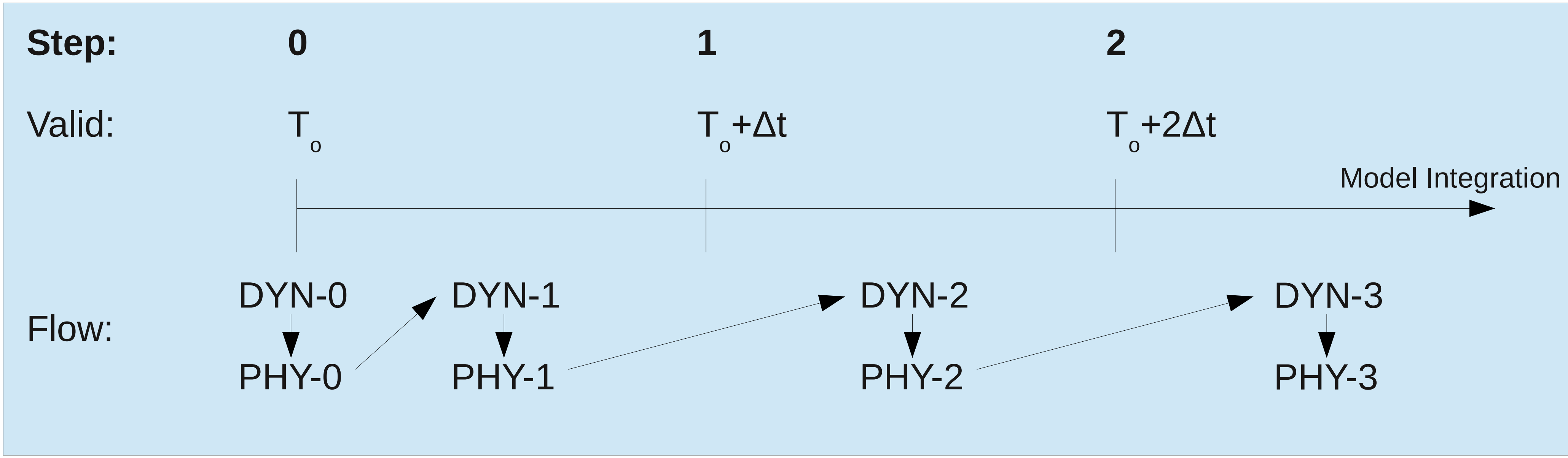
- Step-0 physics is a special step that serves only to initialize the state of the model physics
- Step-0 functionality is identifiable as “kount==0” operations throughout the RPN physics package
- It is not even really a “step” since it does not actually advance the model in time
- The exception to this behaviour occurs in the surface scheme (ISBA), where the surface properties (soil temperature, moisture …) do evolve as a time step at Step-0

Step-0 Physics



- The DYN-0 (interpolation/initialization) and PHY-0 (initialization) are performed at Step-0
- The positioning of PHY-0 is unique, since it does not advance the model state to the next timestep
- Tendencies from all schemes are computed, and the state updated during PHY-0 within the physics column, but the tendencies are never applied (exception: the non-read tracer fields are updated after Step-0 physics for legacy / aesthetic purposes)

Step-0 Physics



- The “physical world” state of the model (the exchange state between the dynamics and physics) is the same at the beginning and end of Step-0 because the validity time is unchanged (surface fields are the exception to this)
- The PHY-0 operations ensure that busper is full at the first real physics timestep, PHY-1
- Physics bus recycling avoids PHY-0 entirely by reading a complete image of the busper state back into the model

Alternative to Bus Recycling

- The drawbacks of physics bus recycling (mostly related to its rigidity) make the alternative Selective Recycling approach much more attractive
- Instead of recycling the entire busper, a small subset of the contents can be recycled using FST files that are written by the preliminary run and read by the physics in the subsequent integration
- The selective recycling technique depends on a few important elements:
  - The technical capability to have the physics read values directly into its own buses
  - The ability of the system designer to identify important physical variables that should be recycled
  - The suppression of Step-0 initializations for variables read during the recycling process

Reading Physics Data

- As of GEM v4.4.0, the RPN Physics package reads its own inputs using the Input Module (IM) – default reading is now done immediately before Step-0 physics, rather than during the entry
- The IM is general, allowing any component to read a field from an FST file, supporting:
  - 2D and 3D fields
  - Horizontal and vertical interpolation of the fields
  - Time interpolation for data stored at both regular and irregular intervals
- For recycling, IM fulfills the reading requirements, but additional work is required to recycle:
  - Step-0 initialization of the read field must be suppressed
  - Consistency checking for surface fields following reading is preliminary in GEM v4.6.0
  - Note that tracers read using the IM have 0 values in the piloting region because of the reduced physics grid extent

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

Field name in input file, or pointer name on bus (gesdict IN=, ON=, VN= in that order of preference. Second component of a vector field is read with optional entry “in2”.

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

Time frame at which IM provides an updated field. Format is freq=[unit],first,[interval],[last] where UNIT (optional) can be STEP (default), MONTH, HOUR or MINUTE. A value of 0 means only read initial data.

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

File search path for the data in order of preference, with implied treatments of field dates. Possible values: GEOP (date ignored), ANAL, CLIM (ip2 for month), INREP (like ANAL, but for a directory).

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;       interp=near;
in=TM; freq=0; search=ANAL;       interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;    timeint=linear
in=I9; freq=0; search=ANAL;       interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;       interp=near;    levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

Order of horizontal interpolation.
Possible values: nearest, linear, cubic (default).

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;       interp=near;
in=TM; freq=0; search=ANAL;       interp=linear;
in=I8; freq=0; search=ANAL,CLIM;  interp=near;    timeint=linear
in=I9; freq=0; search=ANAL;       interp=linear;  levels= 1,2;
in=VF; freq=0; search=GEOP;       interp=near;    levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

Order of time interpolation. Possible values: nearest, linear, step, any, next, none (default)

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

$rpnphy/include/physics_input_table_GENERIC

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

configexp.cfg

```
GEM_anal=/path/to/analysis
GEM_ptopo=10x12x4
GEM_phy_intable=/path/to/my/table
...
```

List of levels to read. Format is levels=first,[last] where special values include 0 (surface), 1... (kind=3 arbitrary level index), -1 (caller-provided list).

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
```

By default, GEM uses a physics input table available at:

`$rpnphy/include/physics_input_table_GENERIC`

This table is designed to reproduce the behaviour of older versions of GEM in which the dynamics was responsible for reading fields for the physics. To modify this file, copy it locally and make the desired changes, then point to the new file in your configexp.cfg.

Reading Physics Data

$GEM_phy_intable

```
in=MG; freq=0; search=GEOP;      interp=near;
in=TM; freq=0; search=ANAL;      interp=linear;
in=I8; freq=0; search=ANAL,CLIM; interp=near;   timeint=linear
in=I9; freq=0; search=ANAL;      interp=linear; levels= 1,2;
in=VF; freq=0; search=GEOP;      interp=near;   levels= 1,26;
...
in=EN; freq=0; search=ANAL;      interp=cubic;  vinterp=cubic;
```

Read of 3D variable EN

- During selective recycling, it is likely that 3D fields will need to be read by the physics: this is triggered using the vinterp key in the input table
- In this example, the final line shows that turbulent kinetic energy (EN) is recycled through the analysis file, read in at the initial time using cubic interpolation (in both the horizontal and the vertical)
- The physics is aware of which fields are mandatory, so that in the example above GEM will not abort if EN is not present in the analysis; instead, a warning will appear in the listings and the run will continue

Reading Physics Data

- Using the IM to read data into the physics solves only part of recycling problem
- The RPN physics package must also be modified to prevent the initialization of a variable that has been read: this must be done only once for each variable
- The phyinread_list_s variable is provided to help with this control
- In the example shown below, TKE is initialized at KOUNT==0 **only** if EN has not been read from a file during initialization (note that phyinread_list_s is valid for each timestep independently)
- The physics have been already updated to suppress initialization of the variables described in this document (as of GEM v4.6.0-b1)
- Surface field initialization and consistency checking, particularly for fields read later in the integration (e.g. a glacier update) have been preliminary implemented in GEM v4.6.0-b1

turbul.ftn90

```
...
#include “phyinput.cdk”
...
  IF (KOUNT.EQ.0) THEN
    INIT_TKE: if (.not.any('en'==phyinread_list_s(1:phyinread_n))) then
      TKE= ...
    endif INIT_TKE
  ENDIF
...
```

Pseudocode snippet for the top level of the PBL scheme, showing the suppression of TKE initialization during Step-0 physics.

An Option for Tracers

- Tracers belong to both the dynamics (for advection) and the physics (for source/sink)
- They can therefore be read two different ways:
  - Using the IM in the physics through the physics input table as described in the previous slides (allows update of field in auto-cascade mode, where the entry is not run)
  - Reading the field into the dynamics as a tracer via the model entry (provides valid values outside the physics domain and will thus eliminate any edge effects related to the advection of IM-read values by strong winds at the boundaries)
- Which of these tracer recycling options to employ depends on the system under consideration
- The HRDPS will be using the IM/physics reading option to recycle high-resolution clouds in an auto-cascade context
- The GDPS will be using the entry tracer nesting as part of the YEC-15 project

An Option for Tracers

gem_settings.nml

```
...
&gement
  E_tr3d_list_S = 'QC'
...
```

prep_cw_rad.ftn90

```
...
#include “options.cdk”
...
  if (any(dyninread_list_s == 'QC')) then
    f(lwc) = d(qcmoins)
  endif
...
```

- Tracer reading by the entry is triggered by the E_tr3d_list_s list in the entry namelist
- The list of entry-read tracers is passed to the physics as dyninread_list_s
- In the example above, the microphysical cloud water field is used to initialize liquid water content for the radiation scheme, replacing the diagnostic initialization at Step-0

Dynamics Recycling

- The model state for the dry dynamics is diagnosed from the fields ready by the entry (generally TT, P0, UU, VV)
- For perfect recycling, the model variables rather than these standard state variables must be recycled:
  - P0 becomes ST1 [$\ln(P_o/P_{ref})$]
  - TT becomes TT1, which is the virtual temperature including hydrometeor loading
  - UU and VV become UT1 and VT1, the horizontally staggered winds on the u and v grids (note that a model running from UT1 and VT1 must have the same rotation as the data, but can cover a smaller domain: rotation is impossible because the grids are not collocated)
- In each case, the operations required to create “standard” output from the model state – and then to read this state back into a new integration – reduces the precision of the reconstructed model state

Dynamics Recycling

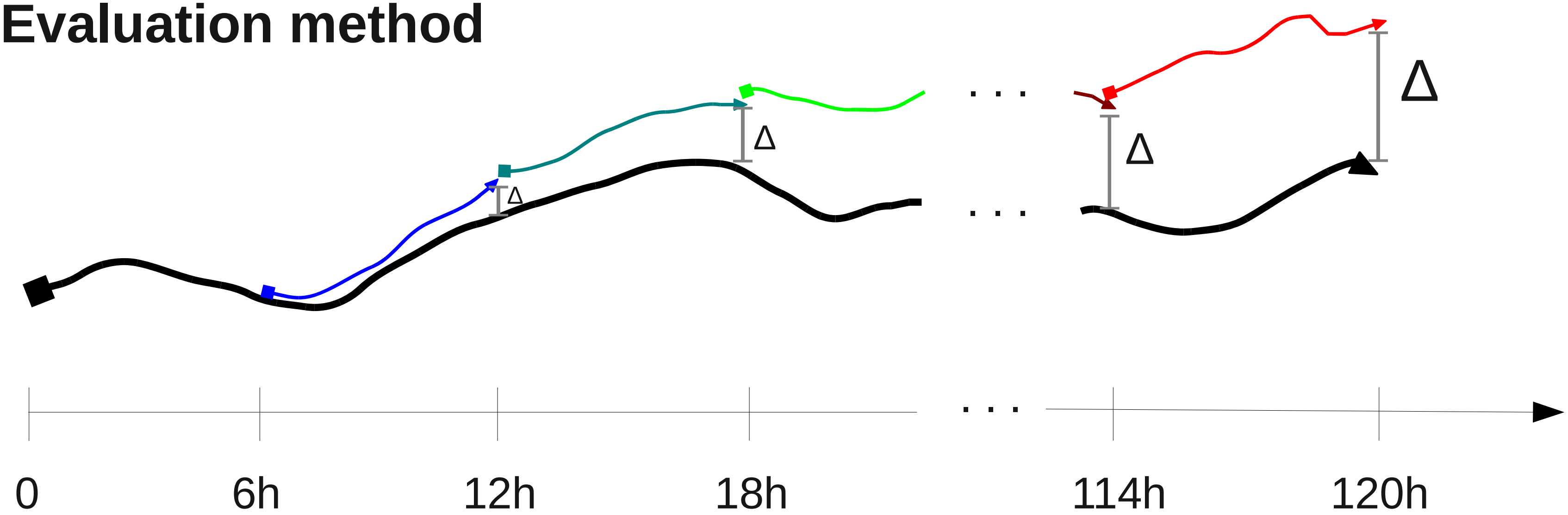
- In certain configurations (e.g. running without physics, but including horizontal diffusion) additional “physical world” variables may also be required for a perfect restart
- As of GEM v4.6.0-b1, the model is equipped to read preferentially the basic model variables rather than their “standard” counterparts
- In practice, the results later in this presentation show that the impact of recycling these fields is small, except for UT1 and VT1 staggering; however, the ability to run perfect dynamics recycling is essential for development and testing

Model Variable	Description	Search Order
ST1	Log of normalized surface pressure $\ln(P_o/P_{ref})$	ST1, P0
TT1	Virtual temperature	TT1, TT, VT
UT1	Image wind in x-direction	UT1, UU
VT1	Image wind in y-direction	VT1, VV
ZDT1	Coordiante vertical motion (“zeta-dot”)	ZDT1
WT1	Real vertical motion	WT1, ZZ
QT1	Nonhydrostatic pressure perturbation	QT1
HU	Specific humidity (tracer)	HU

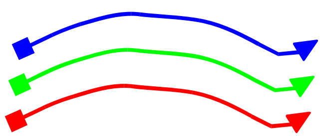
Barriers to Perfect Recycling

- With 32-bit FST records, the combination of physics bus and full dynamics recycling yields a perfect re-initialization of the model (i.e. it is bitwise impossible to tell that an integration has been interrupted)
- However, the identified problems with physics bus recycling and the requirement for additional full-precision 3D outputs in the dynamics makes perfect recycling impractical
- Once selective physics recycling is chosen, the issue of surface evolution at Step-0 and the running of the radiation scheme at Step-0 and Step-1 (regardless of KNTRAD: required for skin temperature) mean that the recycling is imperfect by construction
- Once we accept imperfect recycling, we need a way to evaluate the impact of the recycling errors that we introduce
- The design and testing of such a framework is the topic of the remainder of this session

Evaluation method



control: continuous 120h run



experiment: sequence of 6h restarts, from t = 6 to 120h

Δ

difference* between experiment and control,
at the end of each 6h restart

***** *e.g. spatial average of absolute value of the difference field, for a chosen field***

Experiments

<table>
<tr><td colspan="3">full recycling (of dynamics variables & physics bus)</td></tr>
<tr><td colspan="3">bit-flip test</td></tr>
<tr><td rowspan="3">partial recycling</td><td>dynamics variables</td><td>physics bus</td></tr>
<tr><td>partial</td><td>full</td></tr>
<tr><td>partial</td><td>partial</td></tr>
</table>

Dynamics variables: ST1, TT1, UT1, VT1, ZDT1, WT1, HU, QC

related to surface pressure — *related to temperature* — *staggered image winds* — *related to vertical motion* — *scalars (spec. humidity and condensate)*

Physics bus: hundreds of variables/fields

Results

“Acid” test

Full recycling of dynamics variables and physics bus:
- *bit-pattern validation*
- *curve not shown here*

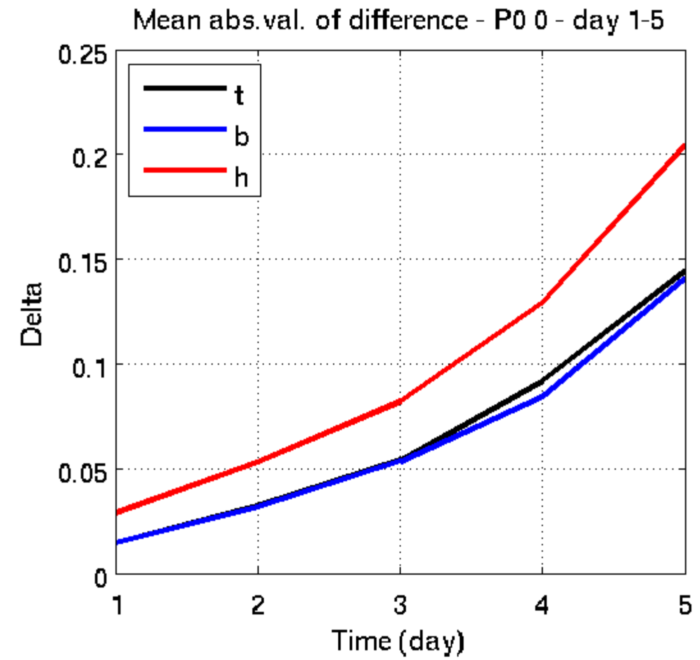
Experiments

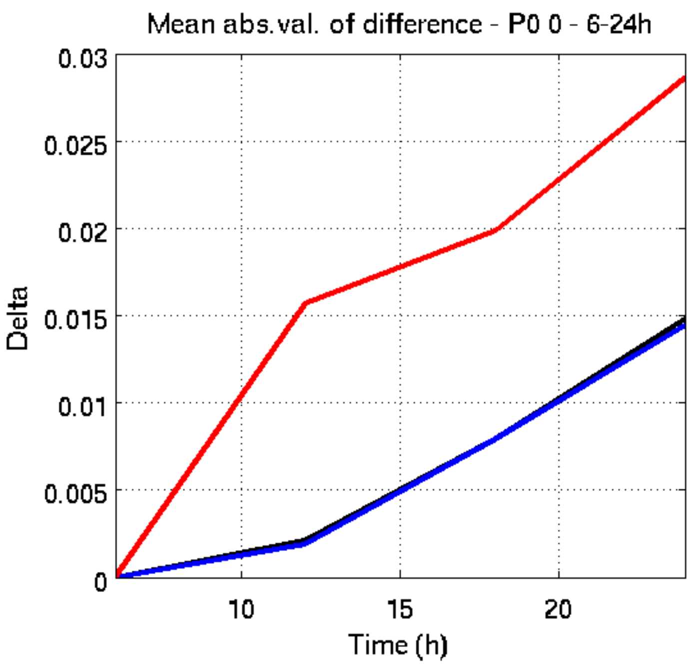
---- bit-flip test

Full recycling of physics but partial recycling of dynamics:

---- use **P0** instead of **ST1**

---- use (**P0, UU, VV**) instead of (**ST1, UT1, VT1**)





Results (cont.)

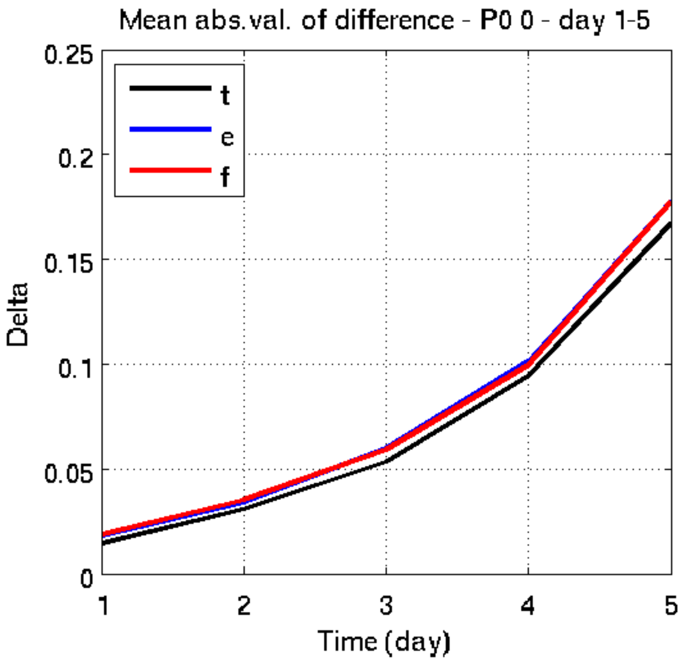
More experiments

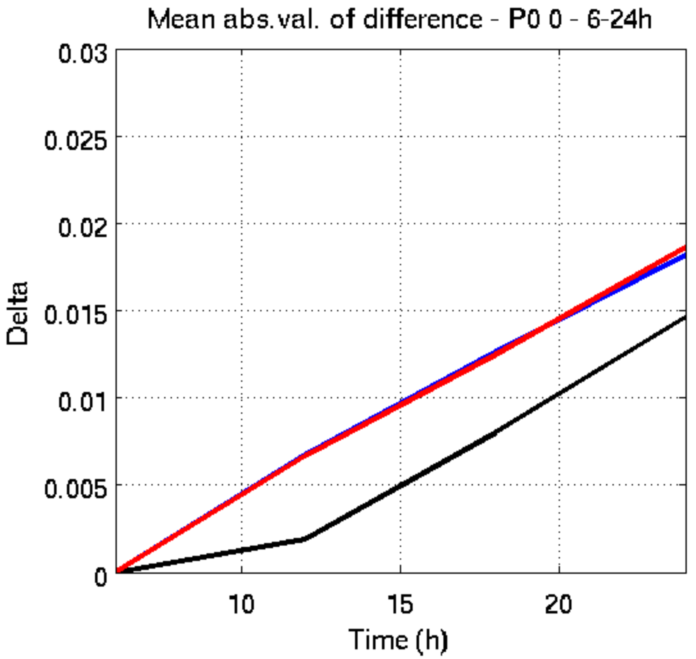
---- bit-flip test

Full recycling of physics but
partial recycling of dynamics:

---- use **P0** instead of **ST1** and
do not recycle **ZDT1, WT1**

---- use **P0** instead of **ST1**,
do not recycle **ZDT1, WT1**
and use **TT** instead of **TT1**





Conclusions about recycling of dynamics variables

- using P0 – instead of recycling of ST1:
  - little impact (errors comparable to those of a bit-flip)

- not recycling ZDT1, WT1:
  - also little impact

- using UU, VV – instead of recycling UT1, VT1:
  - relatively large impact
  - still, the errors are smaller than those caused by the non-recycling of QC and physics (to be shown)

Note: Results shown here are based on tests performed with the <u>hydrostatic</u> dynamics.

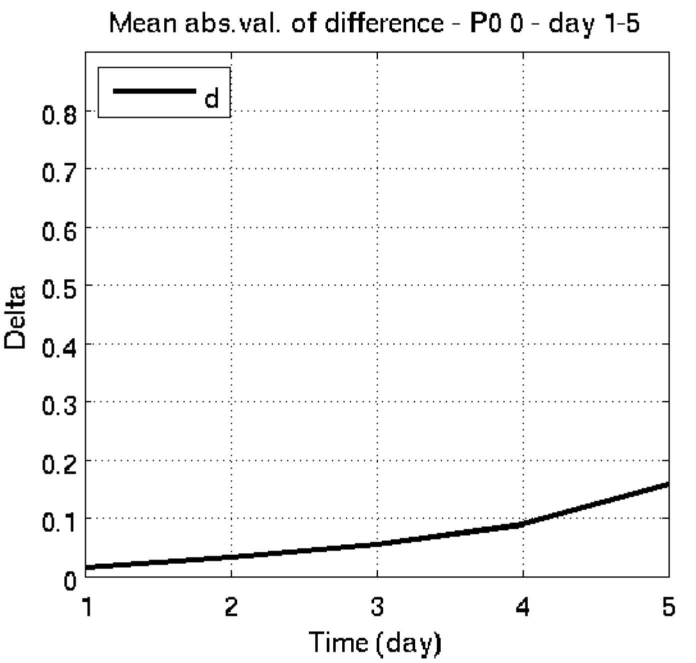
Cloud & physics variables: identified candidates for recycling

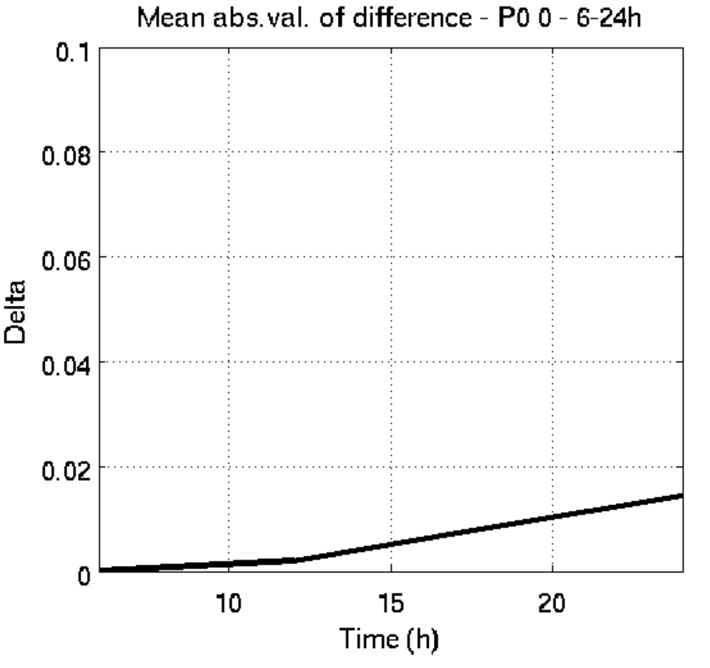
output name (ON)	bus name (VN)	description	bus, related scheme(s)	code modifed	comments
QC	-	3D field, total condensate	dynamics, tracer	--	- everybody wants to recycle this
EN	f(en)	3D field, prognostic TKE	physics, PBL	turbul.ftn90	- questionable initialization at kount=0
TURB	f(turbreg)	3D field, turbulent regime for PBL scheme	physics, PBL	eturbl.ftn90 (if clef is used)	- memory of turbulent regime (laminar vs turbulent), when hysteresis is active
FN	f(ftot)	3D field, cloud fraction	physics, radiation	prep_cw_rad.ftn90	- important input for radiation scheme
QD	f(lwc)	3D field, cloud/water content	physics, radiation	prep_cw_rad.ftn90	- important input for radiation scheme
ZN or ZM	f(zn) f(znm1)	3D field, mixing length (at time t or t-dt)	physics, PBL	turbul.ftn90	- used in a time filtering applied to the mixing length ZN
B5	f(frv)	2D field, friction velocity	physics, surface fluxes and PBL	water.ftn90	- friction velocity at time t-dt is used to compute the roughness length over water (Charnock formula)
H	f(h)	2D field, height of boundary layer	physics, surface fluxes and PBL	inichamp2.ftn90	- H at time t-dt is used in a time-relaxation formula during the turbulent-to-stable transition - in GEM4 H is not purely diagnostics, since it is used in the calculation of KT at the lowest thermo level

Results (cont.)

More experiments:
Impact of recycling of QC and physics

---- use (**P0 ,UT1, VT1, TT1, HU**),
recycle **QC** and entire **physics bus**





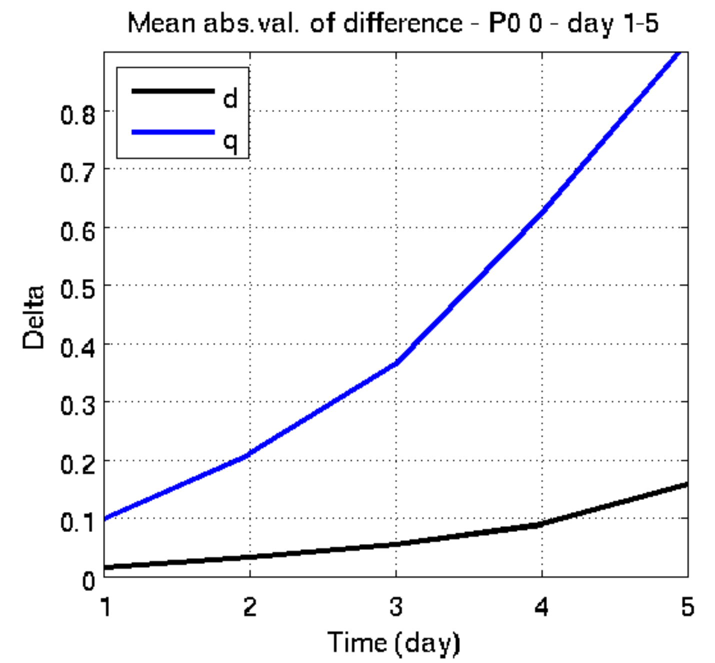
Results (cont.)

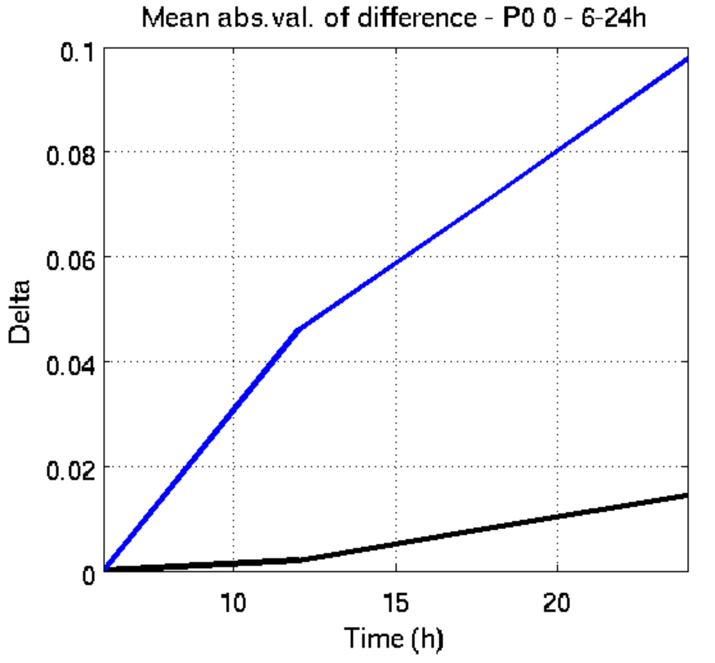
More experiments:
Impact of recycling of QC and physics

---- use (**P0 ,UT1, VT1, TT1, HU**),
recycle **QC** and entire **physics bus**

Fixed recycling of dynamics, with
<u>no recycling of clouds or physics</u>:

---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled





Results (cont.)

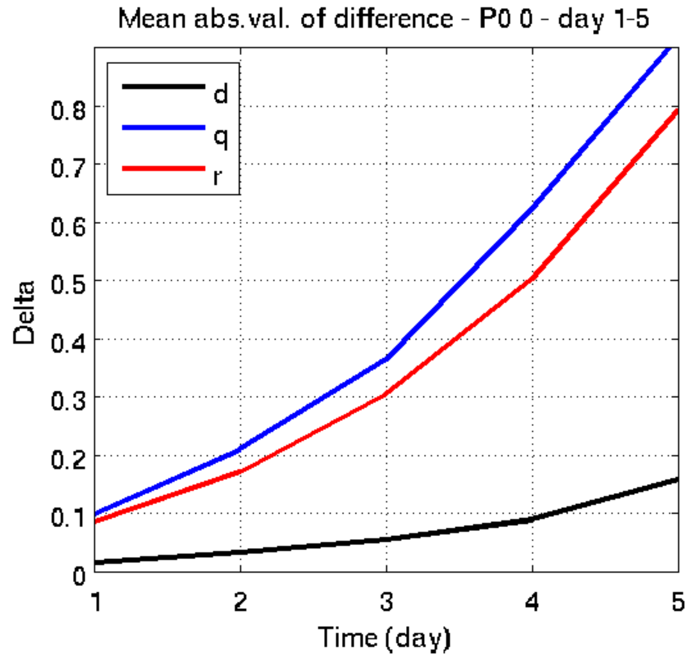
More experiments:
Impact of recycling of QC and physics

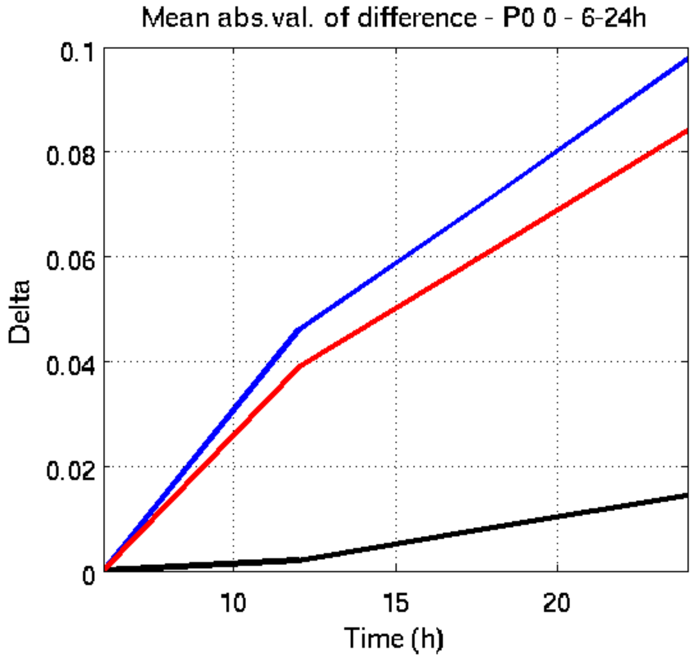
---- use (**P0 ,UT1, VT1, TT1, HU**),
recycle **QC** and entire **physics bus**

Fixed recycling of dynamics, with
<u>no recycling of clouds or physics</u>:

---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled

---- as in **----** plus **QC**





Results (cont.)

More experiments:
Impact of recycling of QC and physics

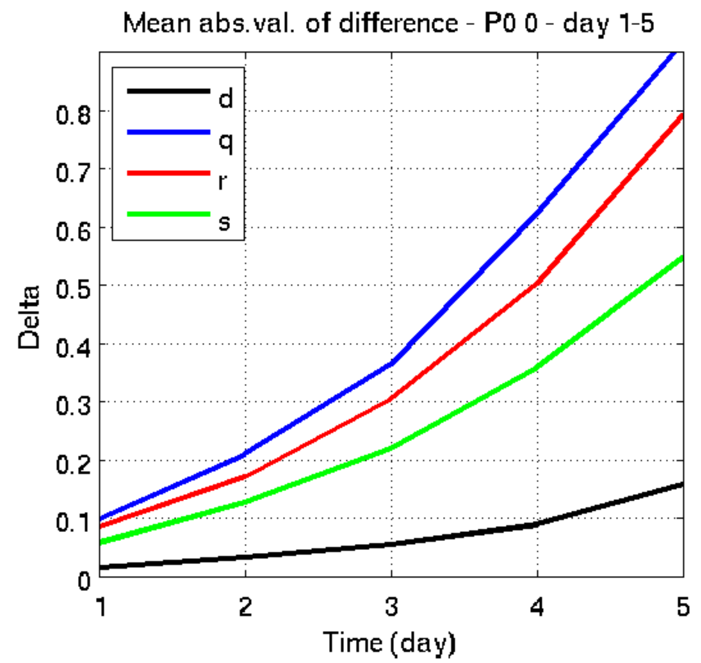
---- use (**P0 ,UT1, VT1, TT1, HU**),
recycle **QC** and entire **physics bus**

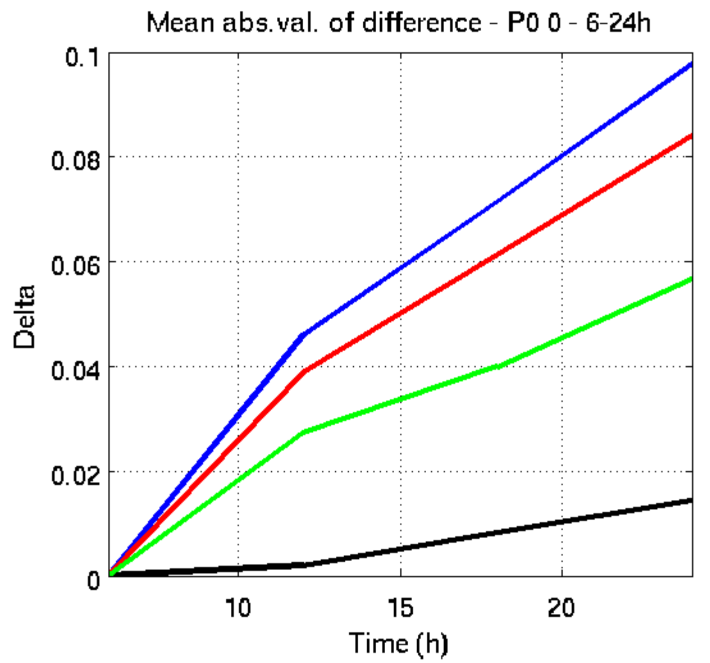
Fixed recycling of dynamics, with
<u>no recycling of clouds or physics</u>:

---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled

---- as in ---- plus **QC**

---- as in ---- plus **EN**





Results (cont.)

More experiments:
Impact of recycling of QC and physics

---- use (**P0 ,UT1, VT1, TT1, HU**), recycle **QC** and entire **physics bus**

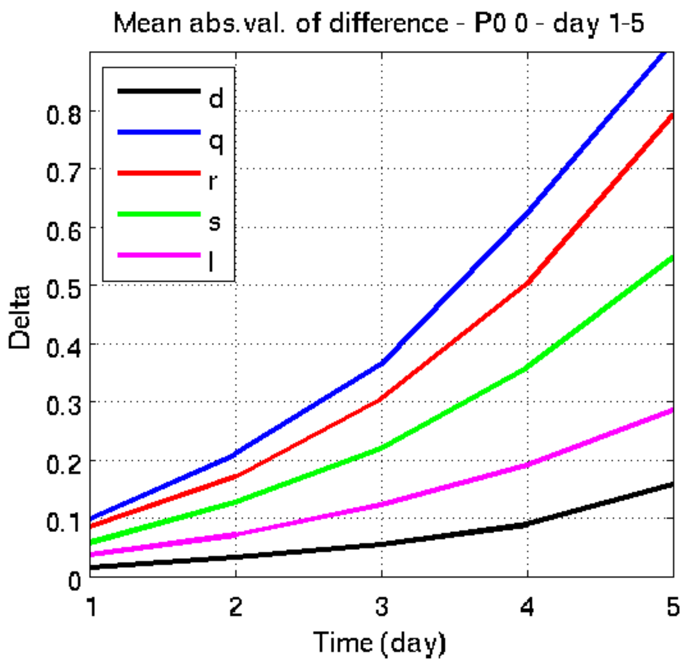
Fixed recycling of dynamics, with
<u>no recycling of clouds or physics</u>:

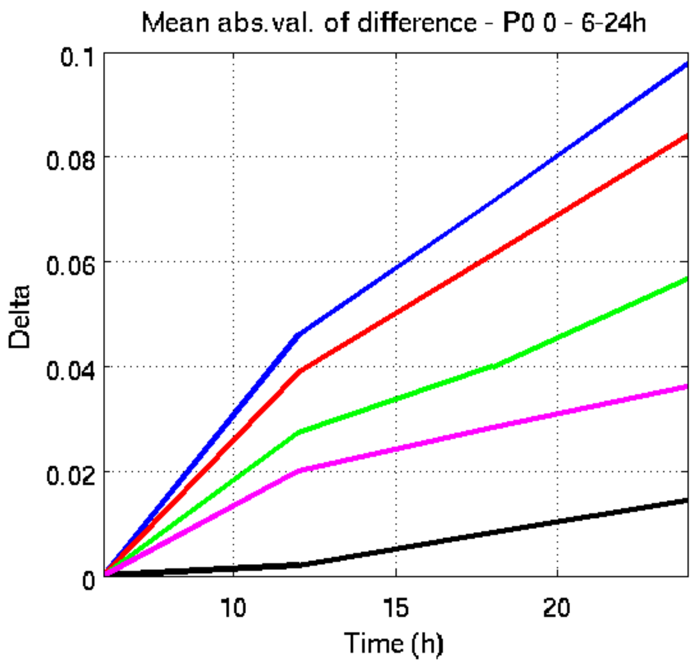
---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled

---- as in ---- plus **QC**

---- as in ---- plus **EN**

---- as in ---- plus **TURB**





Results (cont.)

More experiments:
Impact of recycling of QC and physics

---- use (**P0 ,UT1, VT1, TT1, HU**),
recycle **QC** and entire **physics bus**

Fixed recycling of dynamics, with
<u>no recycling of clouds or physics</u>:

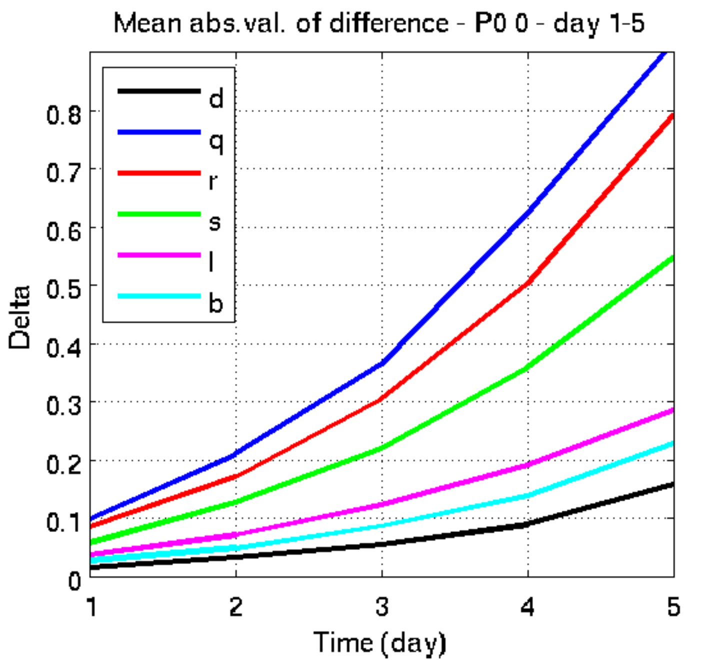
---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled

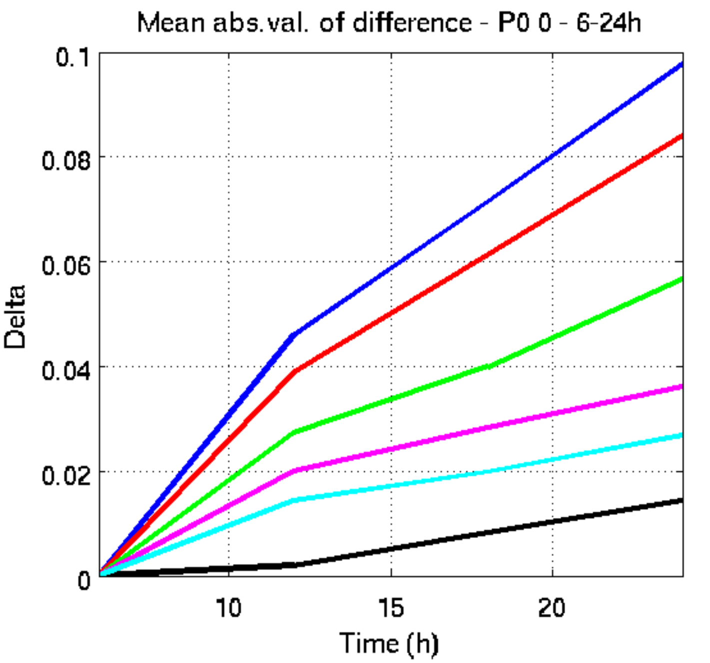
---- as in ---- plus **QC**

---- as in ---- plus **EN**

---- as in ---- plus **TURB**

---- as in ---- plus **ZN, B5, H**





Results (cont.)

More experiments:
Impact of recycling of QC and physics

---- use (**P0 ,UT1, VT1, TT1, HU**), recycle **QC** and entire **physics bus**

Fixed recycling of dynamics, with <u>no recycling of clouds or physics</u>:

---- use (**P0 ,UT1, VT1, TT1, HU**),
no QC,
no physics variable recycled

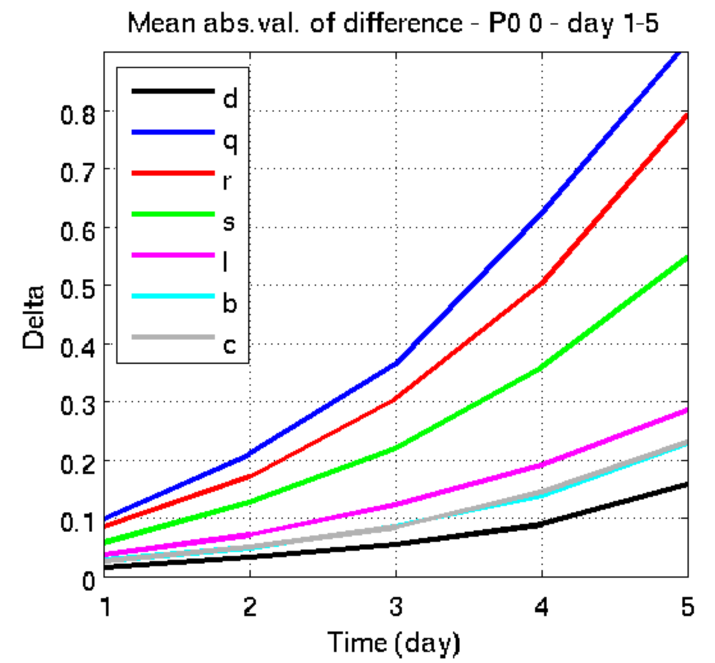
---- as in ---- plus **QC**

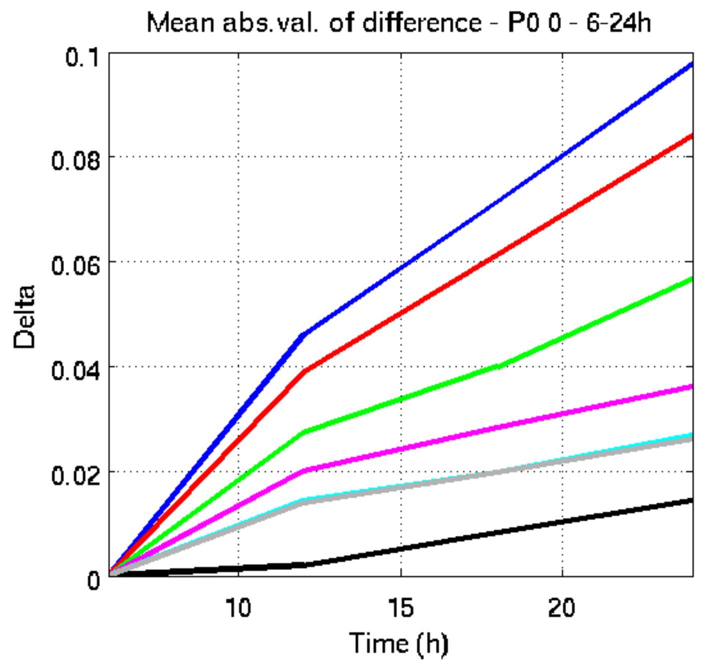
---- as in ---- plus **EN**

---- as in ---- plus **TURB**

---- as in ---- plus **ZN, B5, H**

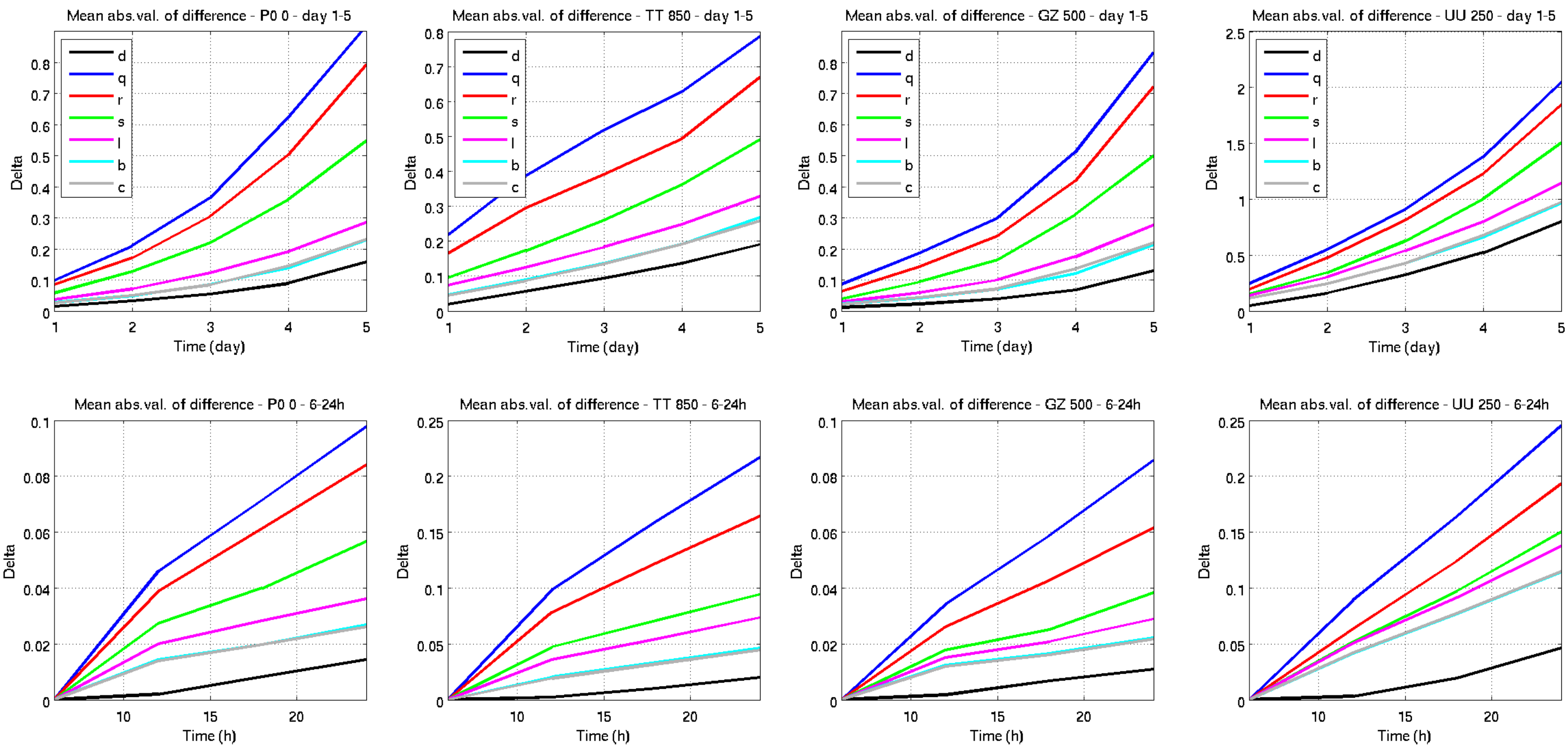
---- as in ---- plus FN, QD





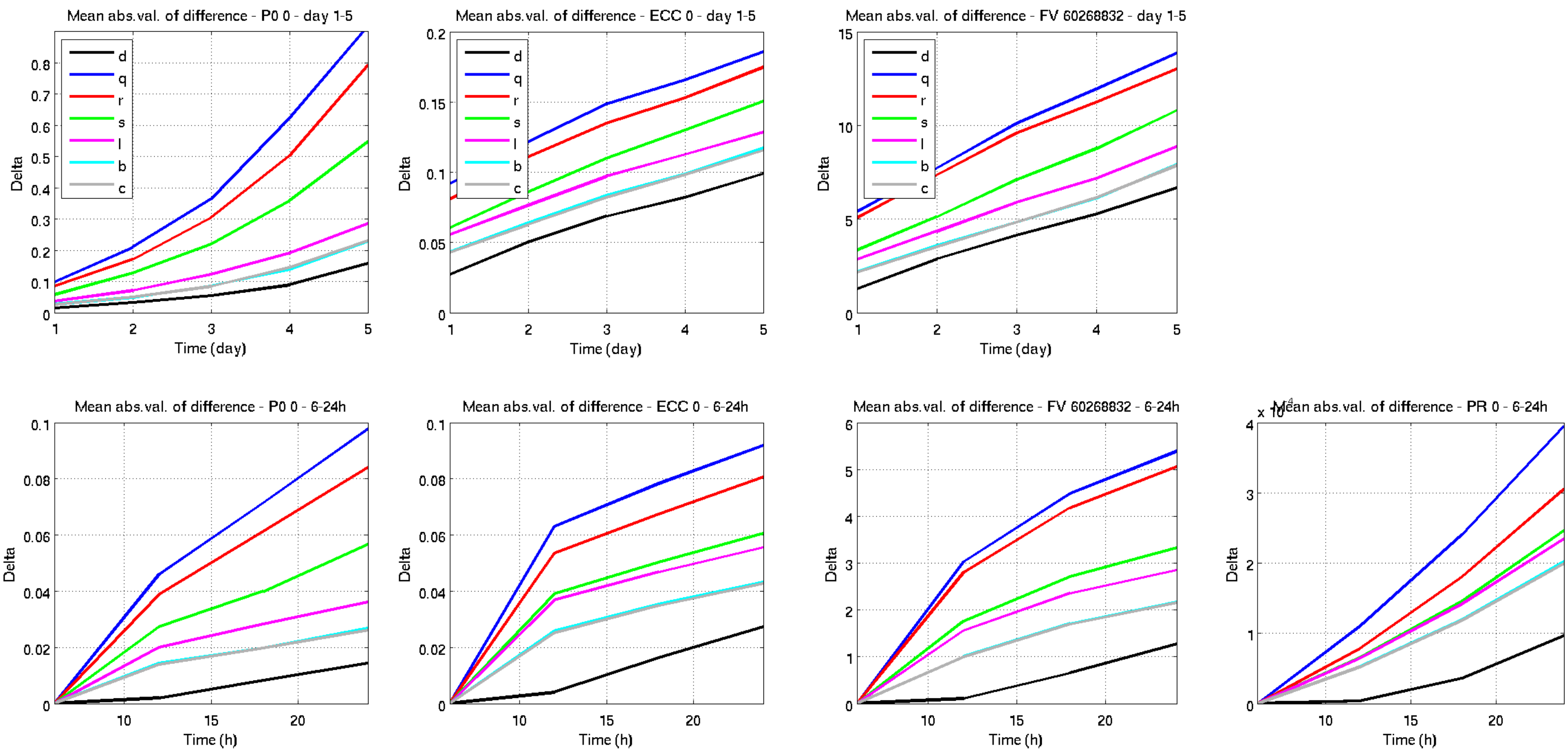
Impact on other variables

P0 | TT850 | GZ500 | UU250



Impact on other variables (cont.)

P0 | ECC | FV | PR



Conclusions on the recycling of clouds & physics variables

A large fraction (up to 90%) of the errors caused by the “cold start” can be eliminated by the recycling of:

- the cloud condensate (QC)
- a few variables (EN, TURB, ZN, H, B5) associated with the PBL scheme

Note:

- so far, all tests were performed using the physics of the GDPS (i.e. work to be done for schemes such as moistke, M.-Y., etc.)
- recycling of surface (ISBA) variables: to be evaluated
- best results if radiation scheme frequency and restart frequency are 'synchronized'

FAQs

- I am running a high resolution LAM. Will these results apply to me?

While the bulk of these results (in terms of the important recycling variables) will probably be valid over a range of scales, this investigation was done with a GDPS physics configuration, and therefore does not include (for example) advanced microphysics with numerous hydrometeor fields. Additionally, the GDPS configuration is hydrostatic, so the impact of QT1 recycling should be evaluated in a high resolution context.

- Why not invert the Step-0 physics and the IM read of the recycled values so that the read values are not overwritten?

The IM read fills the physics entry bus with a large number of analysis, climatological and geophysical fields that are required for the Step-o initializations. This means that the IM read must be run before the physics.

- Why not run a second IM read after Step-0?

Running a second IM read would ensure that all of the overwriting of the read values is undone; however, any diagnostic quantities based on the Step-0 initialization will be based on this poor estimate. The only correct way to avoid overwriting the values read by the IM is to avoid the initializations on a variable-by-variable basis within the physics code. Over time, all important recycling variables will be protected.