Coupling of EULAG dynamical core with COSMO - the software development aspects

Damian Wojcik, Marcin Kurowski,
Michał Ziemiański, Bogdan Rosa,

Institute for Meteorology and Water Management,
Warsaw, Poland

COSMO General Meeting, Rome, September 2011

Outline

1. Introduction

2. EULAG in Fortran 90

3. C&E coupling

   - differences between COSMO and EULAG
   - plugging new dynamical core in
   - data flow
   - parallelization
   - model configuration
   - grid adaptation
   - initialization
   - external forcings

4. Future work

Introduction

CDC Priority Project

Task 1.6: Coupling of EULAG dynamical core with COSMO via an interface.

To better compare the behavior of the new dynamical core in more realistic model applications with full physics parameterisations, the dynamical core of EULAG will be coupled with the COSMO-model. As an intermediate step and to keep the amount of work in a reasonable range this will be done via ***an interface.***

This means that the EULAG dyn. core keeps his own variables, data structure, etc. It is not the aim to have a very efficient code version at this stage but to have a ***useable*** *model version.*

EULAG in Fortran 90

F77 to Fortran 90 migration

- Dynamic memory allocation (ALLOCATE/DEALLOCATE), no COMMON blocks, no DATA and BLOCKDATA statements

- Modularization (data and source code separation, logical code decomposition into specialized modules)

- Fortran 90 language syntax (free format syntax, upper/lower cases, names and comments in English, COSMO-like code indentations, new operators (>, <, etc.), KIND argument for real and integer types, no GOTO instructions,...)

EULAG in Fortran 90

F77 to Fortran 90 migration (cont.)

- Successfully completed 3D idealized test cases w/wo orography

- Still working on:
  - "USE modulename, ONLY:"
  - IMPLICIT NONE
  - consistent error handling
  - removing of C preprocessor usage
  - in-code documentation (comments, recommended COSMO headers, meaningful variable names)
  - automatic arrays

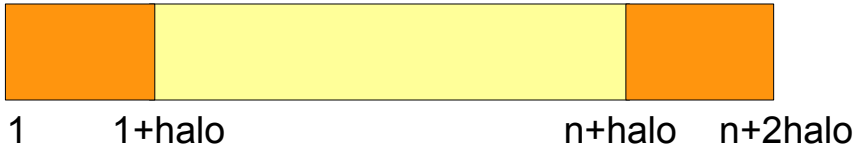
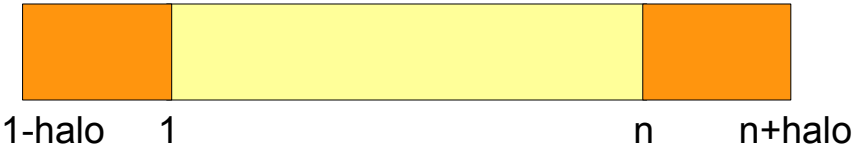
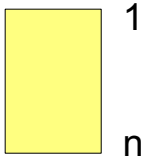
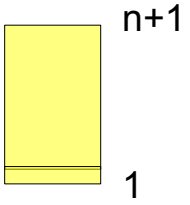
We started to use SVN subversion control system (thanks Oliver!)

EULAG F90 modules

data_absorb_filt	- absorbers (thickness, time scale, etc.)
data_euconstants	- constants (thermo, math and other)
data_eufields	- model variables
data_moist	- moist model variables
data_msg	- processor geometry for MPI (halo, proc number, etc.)
data_param	- grid size, boundary conditions, basic config
data_services	- I/O, history tape
data_sgs	- subgrid scale turbulence model
eulag_utilities	- interpolation C-A grid, transformation geo-contr. vels
organize_eulag	- functions for setting up EULAG dynamical core
src_advec	- MPDATA advection scheme
src_cosmoinit	- MPI initialisation (geomset)
src_kernel	- kernel functions (model initialization, solver)
src_parallel	- one processor service
src_services	- supplementary functions (time, sort, shift, etc.)
preprocessing.def	- preprocessor directives
msg_k	
tempr, tempw	- definitions for temp array in I/O routine
vrtstr.fnc	- vertical stretching function

EULAG F90 vs COSMO

Important differences between COSMO and EULAG

COSMO	EULAG
Matrix indices	
array(1,...,n+2*halo)	array(1-halo,...,n+halo)
	
Vertical indexing	
	
Reference state	
T0, p0	the ≠ f(T0,p0), based on initial temperature
Model variables	
contravariant u,v,w, T, p (C-grid)	geographical u,v,w, Th (A-grid)

C&E coupling – modification of the COSMO code

Files modified (8):

data_runcontrol.f90	- added **leulag** switch for EULAG namelist
lmorg.f90	- setting up EULAG dyn. core: reading namelist, dyn. memory allocation, model initialization
organize_data.f90	- call for output profiles
organize_dynamics.f90	- reading **leulag**
src_allocation.f90	- allocation of 2 new fields carrying info for EULAG
src_artifdata.f90	- call for the() initialization, f_xyz_sharp_straka, *_bd fields
src_runge_kutta.f90	- import of EULAG fields, temperature init, **EULAG dynamical core** instead of R-K core
src_setup_vartab.f90	- added EULAG variables for grib output (PP_EULAG, TH_EULAG, etc.)

C&E coupling – plugging new dynamical core

New elements in main program (lmorg.f90):

!- Section 1: Setup of the model and Namelist input for all components

```
  ! Input of the eulag namelist
  IF (leulag) THEN
    CALL organize_eulag ('input', izerror, yzerrmsg)
    IF (izerror /= 0_iintegers) THEN
      CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                     'organize_eulag: input')
    ENDIF
  ENDIF
```

! Section 2: Allocation of space and computation of constant fields

```
   CALL organize_eulag('init', izerror, yzerrmsg)
    IF (izerror /= 0_iintegers) THEN
      CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                     'organize_eulag: init')
    ENDIF
    CALL organize_eulag('allocate', izerror, yzerrmsg)
    IF (izerror /= 0_iintegers) THEN
      CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                     'organize_eulag: allocate')
    ENDIF

  ENDIF
```

C&E coupling – plugging new dynamical core

!- Section 4: Initializations and allocation of extra space

```
IF (leulag) THEN
  CALL organize_eulag('late_init', izerror, yzerrmsg)
  IF (izerror /= 0_iintegers) THEN
    CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                   'organize_eulag: late_init')
  ENDIF
  CALL organize_eulag('init_fields', izerror, yzerrmsg)
  IF (izerror /= 0_iintegers) THEN
    CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                   'organize_eulag: init_fields')
  ENDIF
ENDIF
```

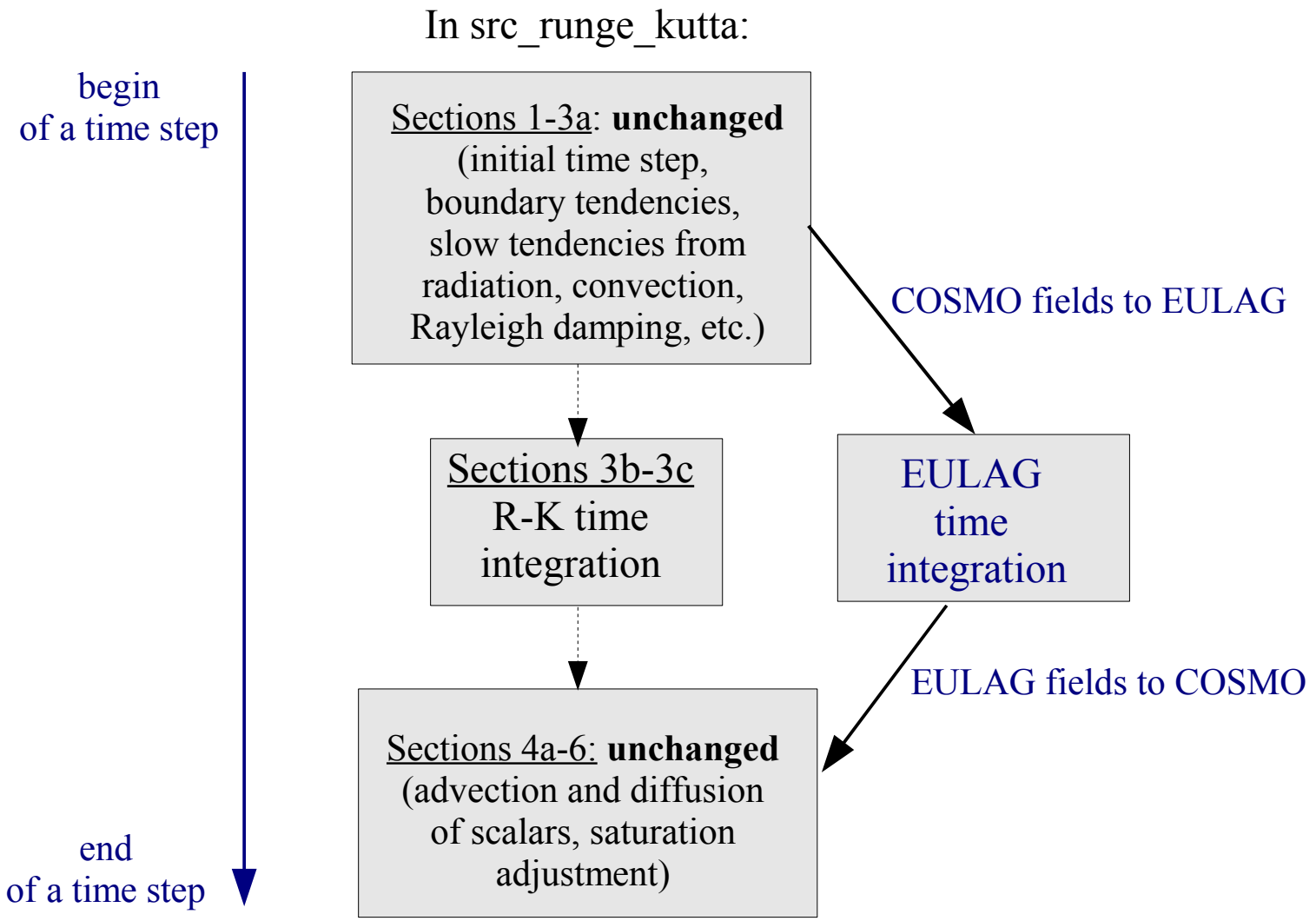
!- Section 9: Final arrays deallocation

```
IF (leulag) THEN
  CALL organize_eulag('clean', izerror, yzerrmsg)
  IF (izerror /= 0_iintegers) THEN
    CALL model_abort (my_world_id, 100+izerror, yzerrmsg,              &
                                   'organize_eulag: clean')
  ENDIF
ENDIF
```

EOF

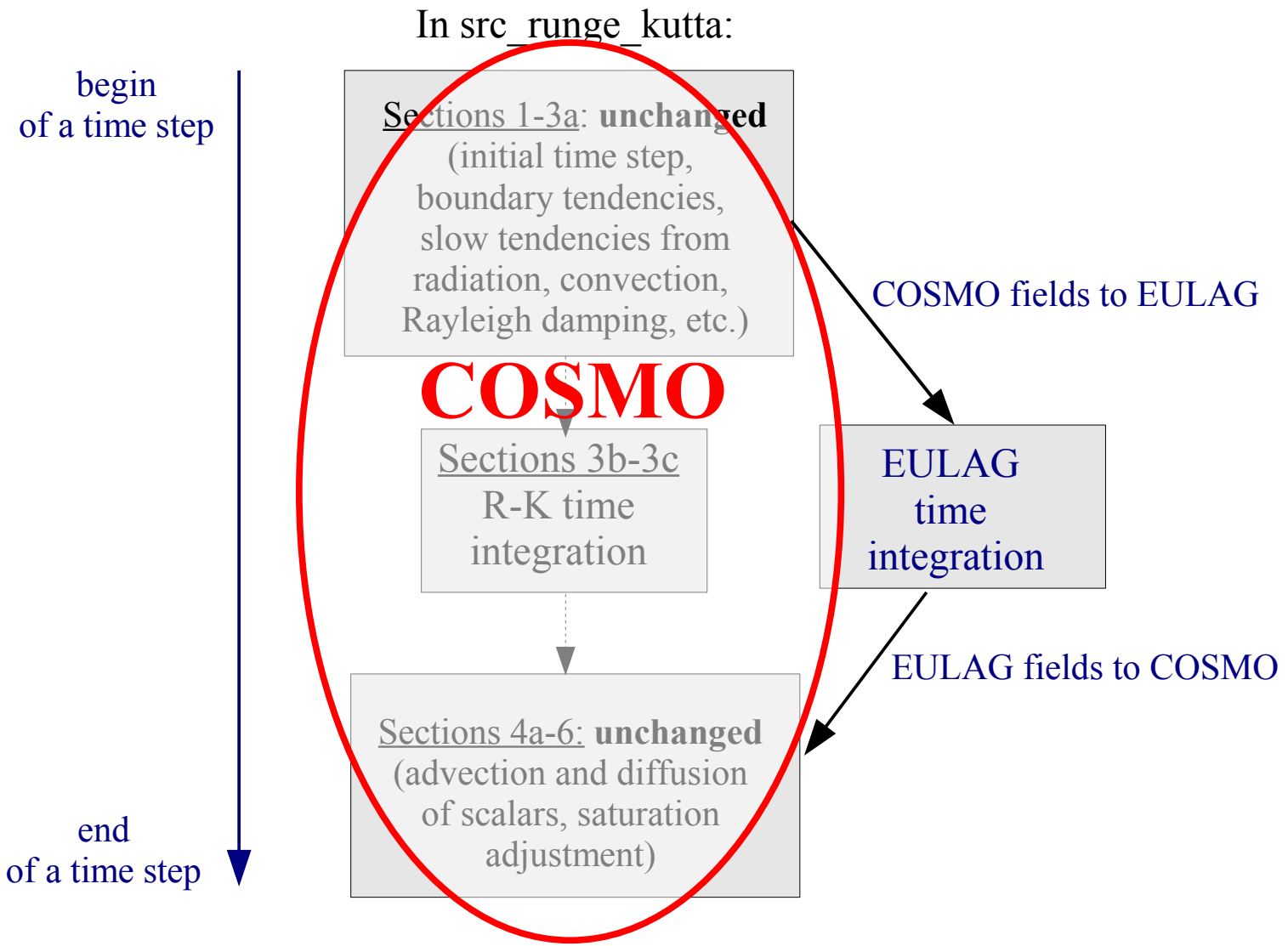
C&E coupling – plugging new dynamical core in

EULAG dynamical core is temporarily plugged instead of default R-K dynamical core in src_runge_kutta module. It uses a set of variables defined in data_eufields and data_absorb_filt modules.



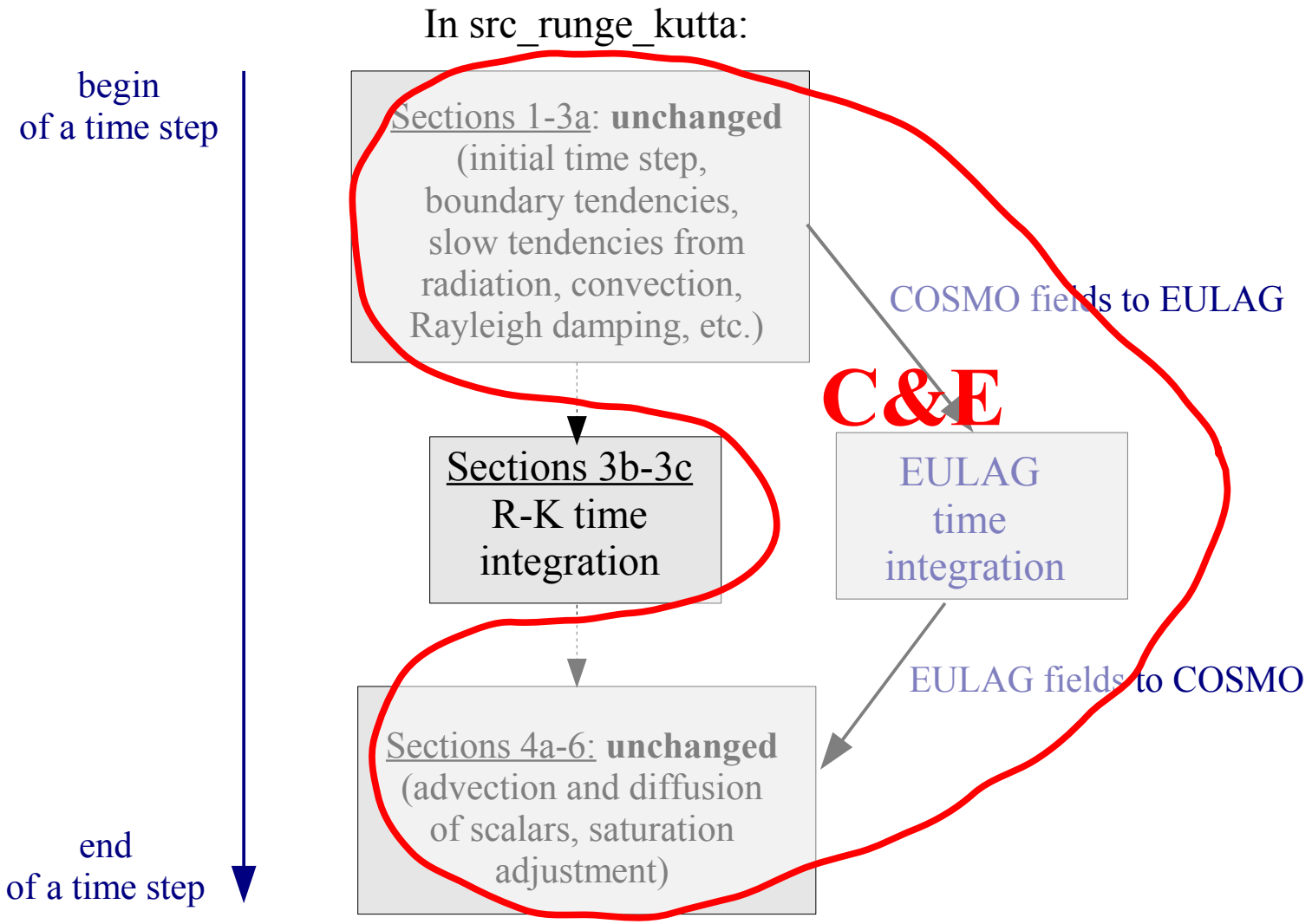
C&E coupling – plugging new dynamical core in

EULAG dynamical core is temporarily plugged instead of default R-K dynamical core in src_runge_kutta module. It uses a set of variables defined in data_eufields and data_absorb_filt modules.



C&E coupling – plugging new dynamical core in

EULAG dynamical core is temporarily plugged instead of default R-K dynamical core in src_runge_kutta module. It uses a set of variables defined in data_eufields and data_absorb_filt modules.

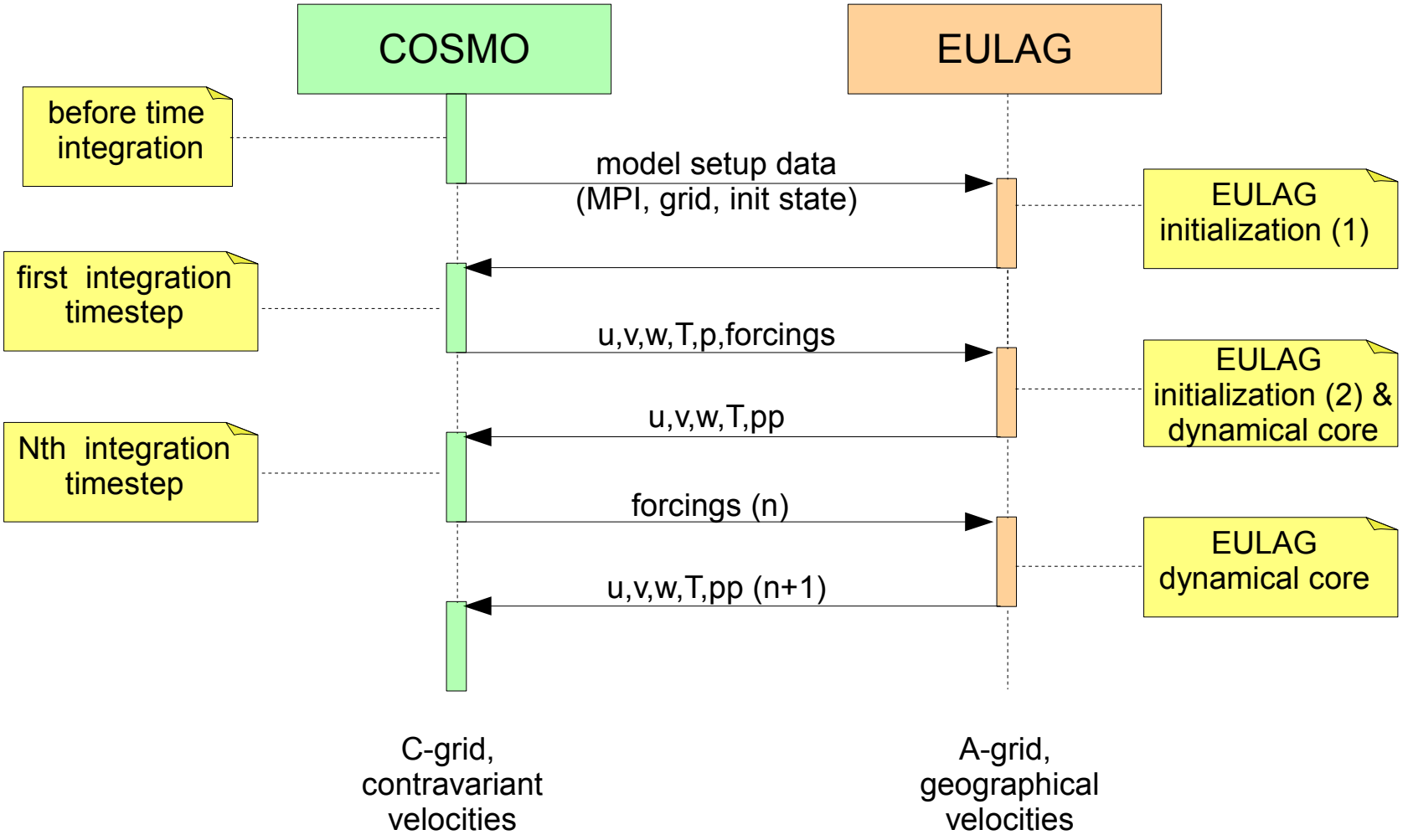


C&E coupling – data flow

Exchanging data between COSMO and EULAG

When data flow is required?

1. at the initialization stage (to initialize EULAG dyn. core)
2. at each time step

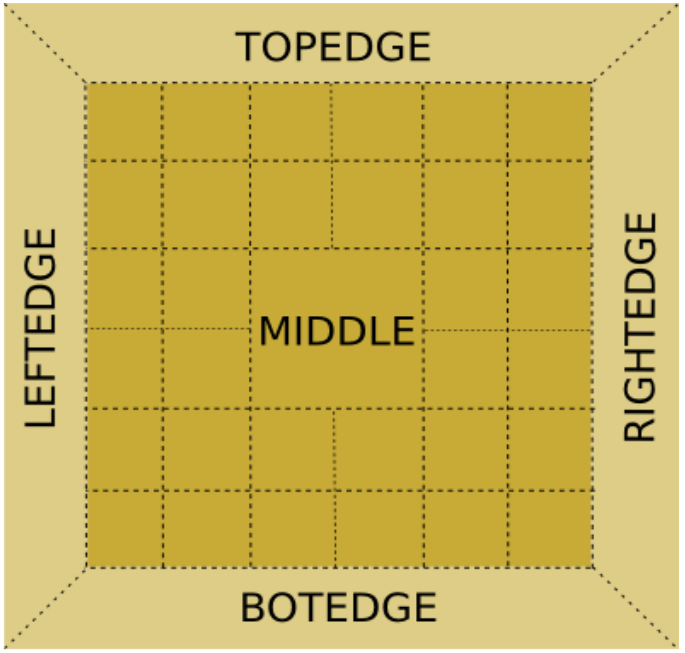


C&E coupling – parallelization

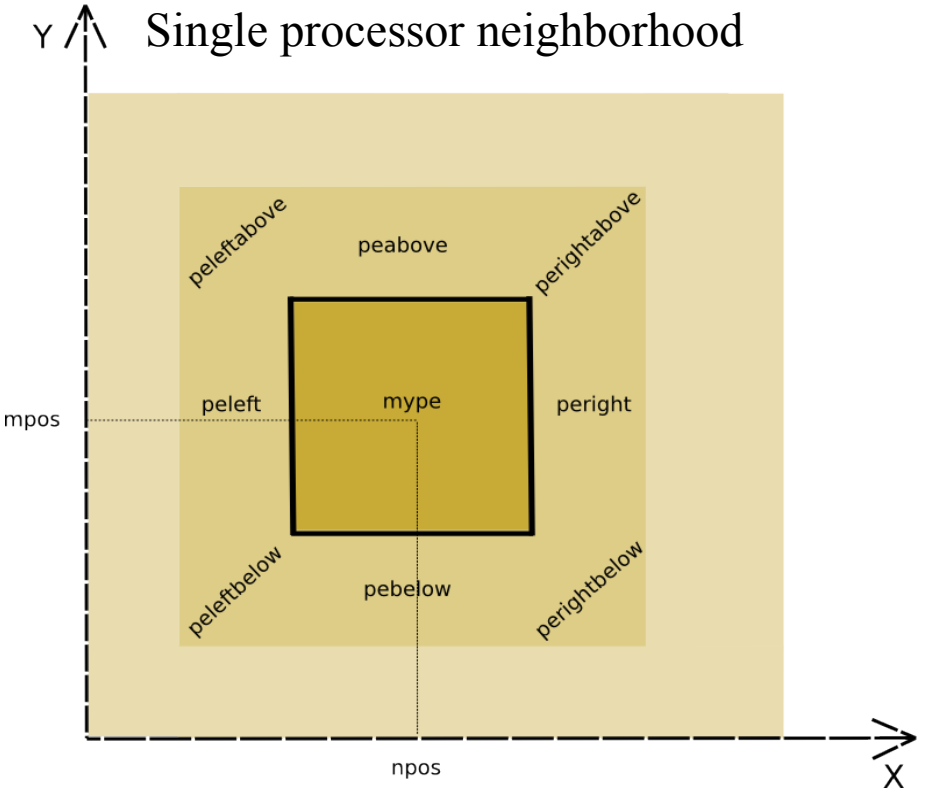
MPI communication is basically organized in COSMO (init_procgrid). A new subroutine (emulate_geomset) is used to initialize EULAG's MPI variables following COSMO structure. This ensures identical domain decomposition in both models, i.e. the same grid points lie on the same processor domain.

=> n/nproc must be integer in EULAG

Domain decomposition in EULAG



Single processor neighborhood



C&E coupling – model configuration

A new configuration of EULAG dyn. core via namelists

run_ideal configuration script:

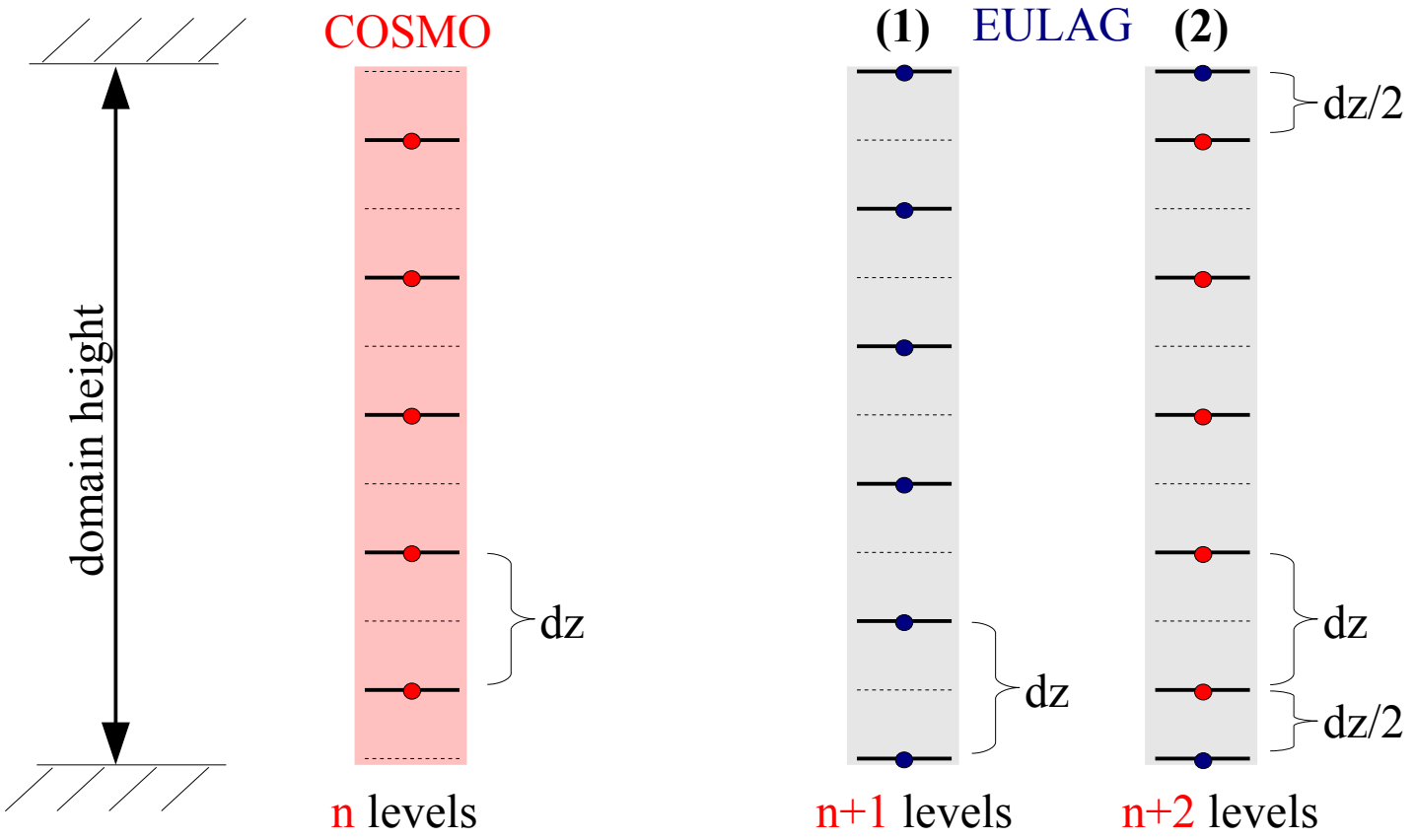
```
cat > INPUT_EULAG << end_input_eulag
 &EULAGCTL

   !-------------------------------------------------------------------
   ! Basic (data_param)
   !-------------------------------------------------------------------
   ! ipsinc = 0 anelastic system
   !          1 pseudoincompressible (Durran) equations.
   ! ideep  = 0 shallow atmosphere approximation: special
   !            for small spheres, re Wedi & Smolar QJR, 2009
   !          1 (default) depp atmosphere
   !-------------------------------------------------------------------
   ipsinc = 0,
   ideep  = 0,

   !-------------------------------------------------------------------
   ! GCRK (data_param)
   !-------------------------------------------------------------------
   ! lrd  - set the size of Krylov subspace lord=lrd
   ! itmn - minimum number of outer cycles of pressure iterations
   ! iprc - use preconditioneer (default on), igrid=1 (B grid) -> iprc=0
   ! ispcpr - use spectral preconditioneer (default off)
   !-------------------------------------------------------------------
   lrd    = 3,
   itmn   = 1,
   iprc   = 1,
   Ispcpr = 0,

(...)
```

The parameters from EULAGCTL namelist are loaded in organize_eulag subroutine. COSMO makefile with some modifications (additional EULAG's modules) is used to compile the code.

C&E coupling – grid adaptation

Open question: what is the best option for coupling COSMO staggered C-grid and EULAG unstaggered A-grid?

There are at least two options for distribution of vertical levels:
(1) interpolation from COSMO to EULAG levels
(2) 1:1 transformation for mass levels with interpolation of the velocities only (operational?)



C&E coupling – grid adaptation

Horizontal grid

In the current version of the C&E model only Cartesian grid is implemented. Slight modification to the code was required as COSMO offers dx and dy as dlon*R_earth and dlat*R_earth.

EULAG's horizontal mesh is constructed as:

```
x(i)=dx*i
y(j)=dy*j
```

For idealized experiments with one-directional flow, it turned out that when dx and dy are of real type with all the numbers important (e.g. dx=99.999384845134...), some gridboxes have a slightly different length. This, in turn, results in generation of noise velocity in direction perpendicular to the flow. For rounded dx and dy the problem disappeared.

The next step: couple terrain-following coordinates and a curvilinear framework (mountains on a sphere). Possible problems with metric terms.

C&E coupling – data initialization in EULAG

Initialization of EULAG environment

Environmental temperature (src_artifdata->organize_eulag):

```
! 0. Import data from t(:,:,:,nnew), t is full temperature!!!
  the(:,:,2:l-1) = (t(:,:,1:ke,nnew))*(p0sl/(p0(:,:,1:ke)+pp(:,:,1:ke, nnew)))**rdocp

! 1. Extrapolate
  the(:,:,1) = 0.5*(3*the(:,:,  2) - the(:,:,  3))
  the(:,:,l) = 0.5*(3*the(:,:,l-1) - the(:,:,l-2))

! 2. Invert the array in z-direction
  the(:,:,1:l) = the(:,:,l:1:-1)

! 3. Call update in order to get correct data in halo
  CALL update(the,np,mp,1,np,mp,ih)
```

Temperature perturbation (src_runge_kutta):

```
!t+t0 is full temperature (with thermals etc.)
 th(:,:,2:l-1) = (t0(:,:,1:ke) + t(:,:,1:ke,nnow))*(p0sl/(p0(:,:,1:ke)          &
                    +pp(:,:,1:ke, nnow)))**rdocp
! 1. Extrapolate
  th(:,:,1) = 0.5*(3*th(:,:,  2) - th(:,:,  3))
  th(:,:,l) = 0.5*(3*th(:,:,l-1) - th(:,:,l-2))

! 2. Invert th array in z-direction
  th(:,:,1:l) = th(:,:,l:1:-1)

! 3. Call update in order to get correct data in halo
  CALL update(th,np,mp,1,np,mp,ih)

! 4. Switch to potential temperature perturbation
  th(:,:,:) = th(:,:,:) - the(:,:,:)
```

C&E coupling – COSMO forcings in EULAG

COSMO forcings from diffusion

- tested for other processes switched off (also for *_tens=0._ireals)
- slight modification of boundary values was required:

In implicit_vert_diffusion_uvwt:

```
! modification of boundary forcings (Eulag):
    ttens(:,:,1)=ttens(:,:,2)
    ttens(:,:,ke)=ttens(:,:,ke-1)
(...)
    utens(:,:,1) = utens(:,:,2)
    utens(:,:,ke) = utens(:,:,ke-1)
(...)
    vtens(:,:,1) = vtens(:,:,2)
    vtens(:,:,ke) = vtens(:,:,ke-1)
(...)
    wtens(:,:,1) = wtens(:,:,2)
    wtens(:,:,ke) = wtens(:,:,ke-1)
```

Without above the boundary forcings were significantly different from those at inner grid points what resulted in gradual cooling at the top and warming at the bottom of the domain(?). Horizontal diffusion seems to work well without any modifications.

C&E coupling – time step in EULAG

Centered in time (2^{nd} order accuracy) time integration:

$$\Psi^{n+1} = \Psi^{n} + \mathrm{MPDATA}(\Psi^{n} + 0.5\Delta t\, R^{n} , u^{n+1/2}) + 0.5\Delta t\, R^{n+1}$$

If $R = R_2(O(\Delta t^2)) + R_1(O(\Delta t))$, then:

$$\Psi^{n+1} = \Psi^{n} + \mathrm{MPDATA}(\Psi^{n} + 0.5\Delta t\, R_2^{n} + \Delta t\, R_1^{n}, u^{n+1/2}) + 0.5\Delta t\, R_2^{n+1}$$

The forcings from COSMO are summed and translated to EULAG's grid, where are integrated with the 1st order of accuracy (R_1):

```
(...)
    u_eulag(i,j,k,0)= u_eulag(i,j,k,0)+fx(i,j,k)*dth +fx_cosmo(i,j,k)*dt
    th(i,j,k) = th(i,j,k) +ft(i,j,k)*dth +ft_cosmo(i,j,k)*dt
(...)
```

Future work

- curvilinear framework (spherical grid, terrain-following coordinates)
- consistent data initialization (contravariant vs. geographical velocities)
- implementation of physics (bl turbulence, radiation, moist processes, etc.)
- extensive testing (!)
- extension of vertical domain (up to 100km?)
- reformulation of open boundary conditions

Thank you!

IMGW 01-673 Warszawa, ul. Podleśna 61
tel.: + 48 22 56 94 361
fax: + 48 22 56 94 356
marcin.kurowski@imgw.pl
www.imgw.pl