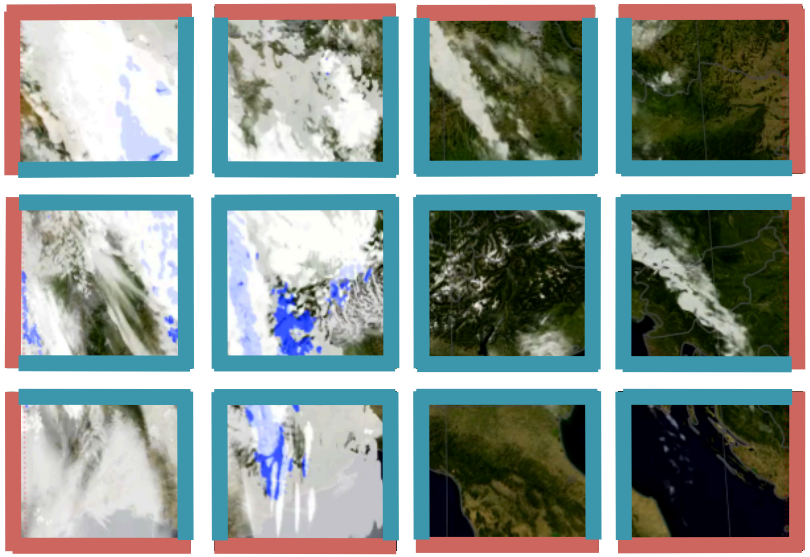


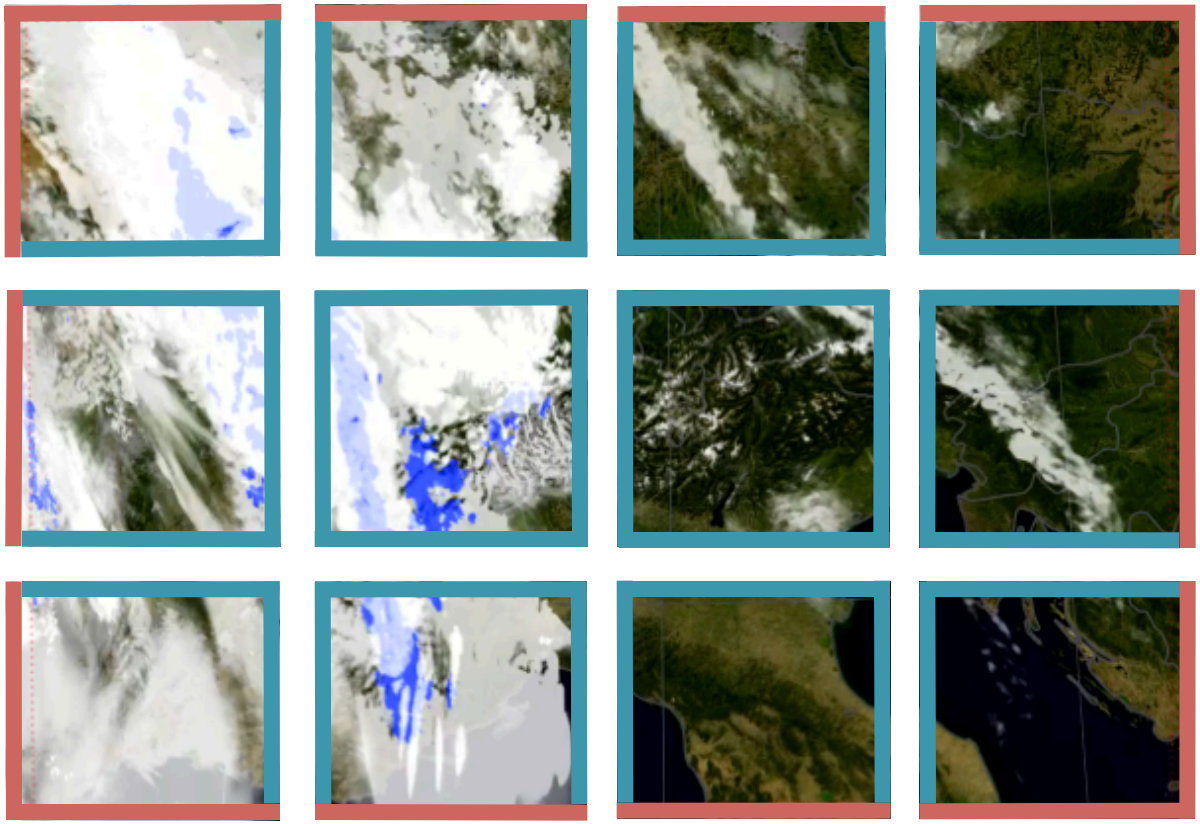
Usage of the
New boundary condition module src_lbc.f90

Anne Roches, Centre for Climate Systems Modeling (C2SM), ETH
Oliver Fuhrer, Federal Institute of Meteorology and Climatology MeteoSwiss
Carlos Osuna, Centre for Climate Systems Modeling (C2SM), ETH

Menu

- Reminder
- Usage in…
  - initialize_loop()
  - fast_waves_runge_kutta()
- Status and Outlook

Scope



BC

This talk!

Halo region

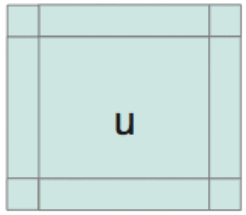
Talk at next COSMO meeting ☺

The Problem

An example:

PE at the NW corner

initialize_loop
…
...
src_runge_kutta
…
...



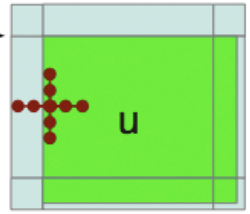
Initializes u for all domain (including boundaries)

stencil compute u

exchg_boundaries(u(nnew), nlines=2)

u modified at compute domain
→ need a halo update
But no BC applied!

hori_diffussion



uses u at the boundaries

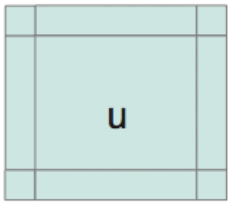
Risk: if some stencil modifies boundaries between initialize_loop & hori_diffusion.

The Problem

PE at the NW corner

Actually fast_waves modifies the boundaries!

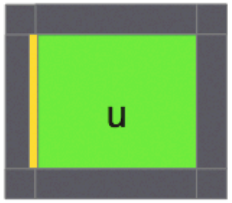
initialize_loop
…
…
src_runge_kutta



Initializes u for all domain
(including boundaries)

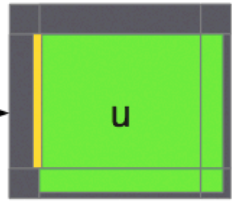
fast_waves
…

1 line BC



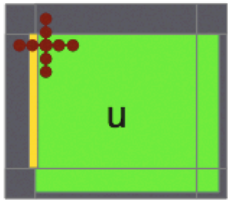
u = u(nnow)
Apply 1 line West/East BC

exchg_boundaries(u(nnew))



u modified at compute domain
→ need a halo update
But no BC applied!

hori_diffussion



uses u at the boundaries

The Proposal

PE at the NW corner

Ideally:

initialize_loop
…
…
src_runge_kutta

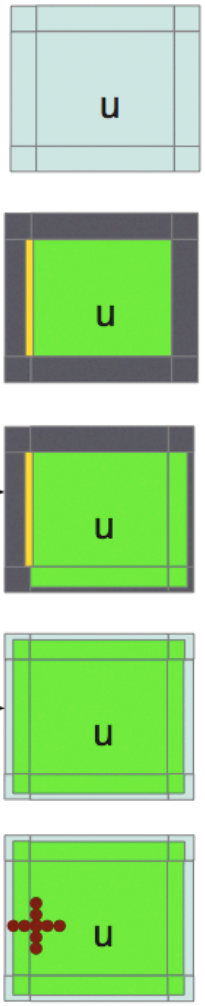
fast_waves
…

1 line BC

exchg_boundaries(u(nnew))

BC (u(nnew))

hori_diffussion



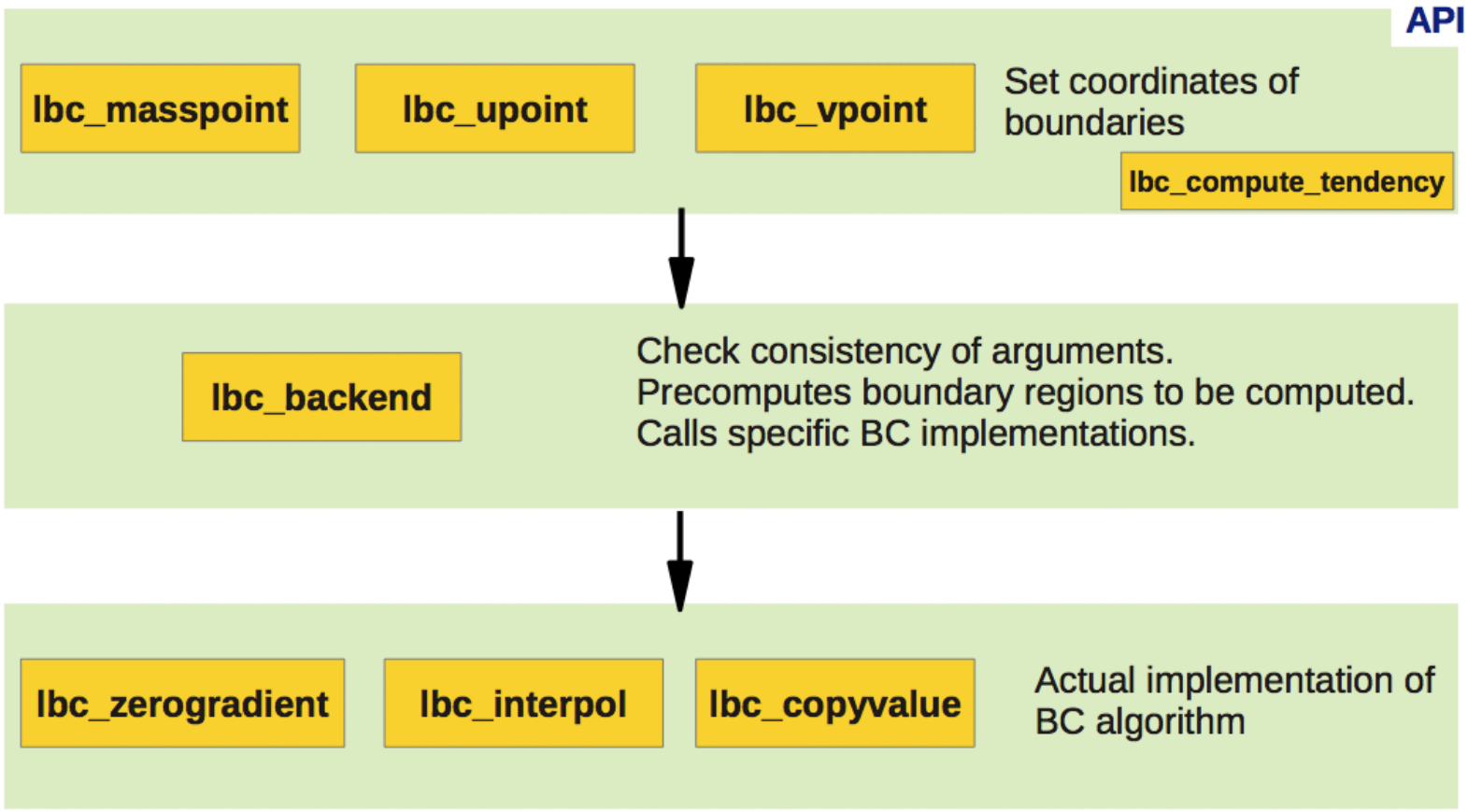
Initializes u for all domain
(including boundaries)

u = u(nnow)
Apply 1 line West/East BC

u modified at compute domain
→ need a halo update
But no BC applied!

uses u at the boundaries

src_lbc.f90 Module



initialize_loop()

```
!-------------------------------------------------------------
!- Section 3: Initialize new time level with boundary values
!-------------------------------------------------------------

  ! factors for linear time interpolation
  z2 = REAL (ntstep+1-nlastbound, wp) / REAL (nincbound, wp)
  z2 = MIN ( 1.0_wp , z2 )
  z1 = 1.0_wp - z2

  ! fields of the atmosphere
  u (:,:,:,nnew) = z1 * u_bd (:,:,:,nbd1) + z2 * u_bd (:,:,:,nbd2)
  v (:,:,:,nnew) = z1 * v_bd (:,:,:,nbd1) + z2 * v_bd (:,:,:,nbd2)
  t (:,:,:,nnew) = z1 * t_bd (:,:,:,nbd1) + z2 * t_bd (:,:,:,nbd2)
  pp(:,:,:,nnew) = z1 * pp_bd(:,:,:,nbd1) + z2 * pp_bd(:,:,:,nbd2)
  IF (.NOT. lw_freeslip) THEN
    w(:,:,:,nnew) = z1 *  w_bd(:,:,:,nbd1) + z2 *  w_bd(:,:,:,
  ENDIF
```

- Inefficient (memory bandwidth bound code)
- Usage of u() in compute domain before another update is an error!

```
!-------------------------------------------------------------
!- Section 5.3: perform boundary update for prognostic variables
!-------------------------------------------------------------

CALL lbc_upoint( BCType_Interpolate, u(:,:,:,nnew),                  &
                 bd1=u_bd(:,:,:,nbd1), bd2=u_bd(:,:,:,nbd2) )
CALL lbc_vpoint( BCType_Interpolate, v(:,:,:,nnew),                  &
                 bd1=v_bd(:,:,:,nbd1), bd2=v_bd(:,:,:,nbd2) )
CALL lbc_masspoint( BCType_Interpolate, t(:,:,:,nnew),               &
                    bd1=t_bd(:,:,:,nbd1), bd2=t_bd(:,:,:,nbd2) )
CALL lbc_masspoint( BCType_Interpolate, pp(:,:,:,nnew),              &
                    bd1=pp_bd(:,:,:,nbd1), bd2=pp_bd(:,:,:,nbd2) )
IF (.NOT. lw_freeslip) THEN
  CALL lbc_masspoint( BCType_Interpolate, w(:,:,:,nnew),             &
                      bd1=w_bd(:,:,:,nbd1), bd2=w_bd(:,:,:,nbd2) )
END IF
```

- Only boundary points are updated
- Weights are computed automatically
- Type of BC / staggering is obvious
- Easy to search for BCs in code

initialize_loop()

```
!- Section 1: Time-related organizational variables
!- Section 2: Reinitialize the tendencies
!- Section 3: Swap time levels and retrieve pointers again
!- Section 4: Initialize some variables (special treatment) for new time step
  !- Section 4.1: update time invariant param. on whole domain using BC data
  !- Section 4.2:  set IC for QI in case QI is not present in the laf
  !- Section 4.3:  compute surface variables for special cases
  !- Section 4.7:  provisory section to compute t(nnew) in case of LF
!- Section 5: Update boundary values for the new time step
  !- Section 5.1: define mask for the boundary points (at masspoint)
  !- Section 5.2: input of new boundary values, if necessary
  !- Section 5.3: perform boundary update for prognostic variables
  !- Section 5.4: perform boundary update for tracers
  !- Section 5.5: perform boundary update for ART species
  !- Section 5.6: perform boundary update for surface fields
  !- Section 5.6.1: surface variables over land points
  !- Section 5.6.2: surface variables over water points
  !- Section 5.6.3: surface water vapor content
  !- Section 5.6.4: weighted surface temperature
!- Section 6: Recompute non-prognostic variables
!  Section 7: Reinitialize vmax_10m
!  Section 8: Check for NaN's
!  Section 9: Check for cold pools
```

Full fields modified

Only BC modified

initialize_loop()

- **Code where full fields are being modified should be checked by code owners**

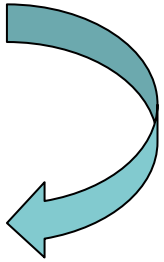
- Often, this code should be moved inside a dummy physics routine or inside the I/O code
  - Example: TERRA
    - If TERRA is switched of (lsoil = .false.) some variables require initialization

- The remaining code may also be indicative of an error
  - Example 1: Assimilation
    - Usage of ps(nnew) before it is computed
  - Example 2: LF core
    - Usage of t(nnew) before it is computed

fast_waves_runge_kutta()

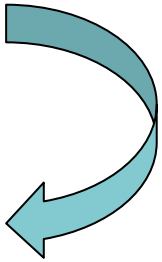
```
! zpi has to be set to 0 (at least at the boundary)
zpi(:,:,:) = 0.0_ireals
```

```
! FUO_BC: BC for divergence is zero, although these values should not be used
!         in the computations for the fast-wave solver

CALL lbc_masspoint( BCType_Value, zdiv, value=0.0_ireals )
```



fast_waves_runge_kutta()

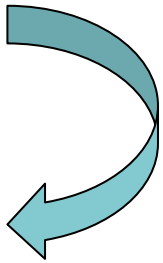
```
IF ( .NOT.lradlbc_u .AND. .NOT.lperi_x) THEN
  ! west
  IF (my_cart_neigh(1) == -1) THEN
    DO  k = 1, ke
      DO  j = jstartu, jendu
        u(istartu-1,j,k,nnew) = u(istartu-1,j,k,nnew) + zubdt_west(j,k)
      ENDDO
    END DO
  ENDIF
  ! east
  IF (my_cart_neigh(3) == -1) THEN
    DO  k = 1, ke
      DO  j = jstartu, jendu
        u(iendu+1,j,k,nnew) = u(iendu+1,j,k,nnew) + zubdt_east(j,k)
      ENDDO
    END DO
  ENDIF
END IF

IF ( .NOT.lradlbc_v  .AND. .NOT.lperi_y) THEN
  ! south
  IF (my_cart_neigh(4) == -1) THEN
    DO  k = 1, ke
      DO i = istartv, iendv
        v(i,jstartv-1,k,nnew) = v(i,jstartv-1,k,nnew) + zvbdt_south(i,k)
      ENDDO
    END DO
  ENDIF
  ! north
  IF (my_cart_neigh(2) == -1) THEN
    DO  k = 1, ke
      DO i = istartv, iendv
        v(i,jendv+1,k,nnew) = v(i,jendv+1,k,nnew) + zvbdt_north(i,k)
      ENDDO
    END DO
  ENDIF
END IF
```



```
! FUO_BC: Set boundary values for u, v from the boundary tendencies

IF ( .NOT. lradlbc_t ) THEN
  CALL lbc_masspoint( BCType_Tendency, t(:,:,:,nnew), tend=t_bd_tend, dt=dts )
END IF
IF ( .NOT. lradlbc_pp ) THEN
  CALL lbc_masspoint( BCType_Tendency, pp(:,:,:,nnew), tend=pp_bd_tend, dt=dts )
END IF
```

fast_waves_runge_kutta()

```
IF (lw_freeslip .AND. .NOT.lperi_x ) THEN

  IF (my_cart_neigh(1) == -1) THEN
    DO  k = 1, ke1
      DO i = 1, nboundlines
        DO  j = jstart, jend
          w(i,j,k,nnew) = w(istart,j,k,nnew)
        ENDDO
      ENDDO
    ENDDO
  ENDIF

  IF (my_cart_neigh(3) == -1) THEN
    DO  k = 1, ke1
      DO i = ie-nboundlines+1, ie
        DO  j = jstart, jend
          w(i,j,k,nnew) = w(iend  ,j,k,nnew)
        ENDDO
      ENDDO
    ENDDO
  ENDIF

ENDIF

IF ( lw_freeslip .AND. .NOT.lperi_y ) THEN

  IF (my_cart_neigh(4) == -1) THEN
    DO  k = 1, ke1
      DO  j = 1, nboundlines
        w(:,j,k,nnew) = w(:,jstart,k,nnew)
      ENDDO
    ENDDO
  ENDIF

  IF (my_cart_neigh(2) == -1) THEN
    DO  k = 1, ke1
      DO  j = je-nboundlines+1, je
        w(:,j,k,nnew) = w(:,jend  ,k,nnew)
      ENDDO
    ENDDO
  ENDIF

ENDIF
```



```
! FUO_BC: apply a freeslip (zero-gradient) boundary condition to w

IF ( lw_freeslip ) THEN
  CALL lbc_masspoint( BCType_ZeroGradient, w(:,:,:,nnew) )
ENDIF
```

Performance

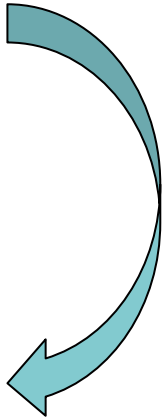
- COSMO-2 production run on Cray XE6 using GNU compiler

ORIG (v5.0 without src_lbc.f90 in initialize_loop()):

```
 ------------------------------------------------------------------------
  Local timers:
   NCOMP_PE= 1075
 ------------------------------------------------------------------------
  Id      Tag           Ncalls         min[s]         max[s]       mean[s]
   1          timeloop    5940     1772.3080      1772.3290      1772.3219
  10              init    5940       79.1440       103.4520        97.6740
  11               lbc    5940        7.5290        22.6570        18.1932
  20           physics    5940      157.0980       292.4790       206.2071
  50          dynamics    5940     1074.2300      1231.6060      1144.9825
 ------------------------------------------------------------------------
```

MODIF (v5.0 with src_lbc.f90 in initialize_loop()):

```
 ------------------------------------------------------------------------
  Local timers:
   NCOMP_PE= 1075
 ------------------------------------------------------------------------
  Id      Tag           Ncalls         min[s]         max[s]       mean[s]
   1          timeloop    5940     1753.1510      1753.1740      1753.1647
  10              init    5940       72.3250        89.4840        79.5422
  11               lbc    5940        0.3370        11.2850         1.5331
  20           physics    5940      157.5660       300.5750       206.9448
  50          dynamics    5940     1073.5370      1233.1070      1145.2752
 ------------------------------------------------------------------------
```



Status

- Routines switched to src_lbc.f90
  - initialize_loop()
  - set_trcr_special_bc()
  - advection_pd()
  - fast_waves_runge_kutta()

- The module src_lbc.f90 has been presented and sent for review after GM12 (that's 23.5 months ago!!!)

Status

Perferctive maintenance (does not change results)

- ☑ Information of WG2 and WG6 coordinators
- ☑ Design, development
- ☑ Presentation
- ☑ 4-eyes principle
- ☑ Coding standards respected
- ☑ Passes the technical testsuite (over 30 configurations)
- ☑ Internal unit-test
- ☐ Provide documentation (what form?)
- ☐ Nominate a responsible person

Take home messages

- There are errors concerning the BCs in the current code
- It is far from obvious to find these errors and to find out where BCs are being applied to each fields
- src_lbc.f90 module can help by making BCs…
  - less error prone
  - more explicit
  - more concise
  - more efficient
  - easier to port to GPUs
- BCs should be grouped with exchg_boundaries() as much as possible
- Review it and include it in COSMO 5.2