TSA migration to “block structure” and ICON input

Preface

- The current TERRA code (5.06b_4) is unified for COSMO and ICON and designed for the so called “block structure”.
- Terra-standalone (TSA) is modified to use this new TERRA code as well for ICON input data.
- All the current data statements and modules of COSMO needed for TSA are taken from the current COSMO version. Of course there are some changes or modifications necessary. This is listed below together with the new routines.
- For the turbulence scheme the SR parturs_new is brought to “block structure”, the new turbulence code is NOT used.

TSA code structure

- Main program: terra_TSA.f90
- terra_TSA contains the following files with a lot of subroutines:

```
CONTAINS
INCLUDE 'terra_io.f90'          ! input and output routines
INCLUDE 'terra_lmparam.f90'     ! additional LM parameterization needed by the soil module
INCLUDE 'terra_lmenv.f90'       ! subroutines to establish the fields and variables
                                ! needed by the program
```

- This is kept for the moment.

New subroutines

- New TERRA code is integrated:
  - **sfc_terra_data.f90**
  - **sfc_terra.f90**
  - **sfc_terra_init.f90**
  - **sfc_utilities.f90**
  - turb_data.f90
- New in terra_lmenv.f90:

- **allocate_block_fields**

- New in terra_lmparam.f90:
  - **parturs_newblock**

- New in terra_io.f90 for reading ICON fields:
  - **read_const_fields_icon**
  - **read_initial_fields_icon**
  - **read_icongrib** (new for read_lmgrib)

- New**: tsa_sfc_interface.f90** is an extract of sfc_interface.f90 with these SR:
  - **tsa_sfc_init**
  - **tsa_sfc_init_copy** (imode=1: copyToBlock, imode=2: copyFromBlock)
  - **tsa_sfc_organize**
  - **tsa_sfc_finalize**
  - sfc_in_wkarr_alloc
  - sfc_in_wkarr_dealloc

- For the new routines implemented for eccodes for GRIB-IO see documentation I; here are some additions:

  ➢ Module gribio.f90
    - (new) read_grib_eccodes (substitute for read_grib) is adapted to ICON fields.

  ➢ Module terra_io.f90
    - **get_grib_info_icon** for meta data of ICON
    - grbout_eccodes also is adapted to ICON fields.

<u>Modified data modules / subroutines / main program:</u>

- terra_TSA:
  - Introduce kind_parameters (wp)
  - Additions for skin temperature par. (itype_canopy=2)
  - Add t_snow_mult, rho_snow_mult (multi layer snow)

  - USE data_block_fields
  - Add new itype parameter: itype_hydcond, itype_canopy, itype_mire
  - nlgw not used anymore
  - data_runcontrol: use nproma, nlastproma and nblock
  - data_soil is replaced by sfc_terra_data
  - src_soil_multlay is replaced by (tsa_)sfc_interface, sfc_terra_init and sfc_terra
  - Replace terra_multlay by tsa_sfc_organize, calling terra_init for nt=0 and terra for nt>0

- data_terra_standalone: meta data for ICON unstructured grid
- read_namelist: Added ymodel, nproma (→ nblock, nlastproma according to array length), itype…., idiag_snowfrac
- allocate_fields (in terra_lmenv.f90): adapted for ICON and some new fields needed for new TERRA code
- init_variables (in terra_lmenv.f90): define arrays for copying to and from block structure
- clean_up (in terra_lmenv.f90): deallocation of block fields added
- near_surface (in terra_lmparam.f90): NOT used from new COSMO version, adapted to ICON fields

REMARKS:

GRIDS

- Fields rlon and rlat of the desired grid are computed due to NL input. rlon_in/rlat_in are only read in in read_const_fields for comparison to decide if the grid/area coincides with the desired one. According to this comparison, the fields match, will be cropped or interpolated. So the read in fields will be manipulated to result in the new desired fields (according to NL grid input), which will be processed further. After this is done, the read in fields will be deallocated.
- In other routines (read_initial_fields, read_lmgrib) first the grid parameters are read for the first field (get_grib_info) and then compared to the desired grid (defined per NL). Assumption: All the fields have the same grid! If the fields have to be

interpolated (spatial), rlon_in, rlat_in will be computed out of the meta data (coming from get_grib_info). Then the fields will be manipulated as already described above.

- In each routine the grids will be compared and adapted to the desired one.
- So, the input grid of constant data, initial data and forcing data could be different in principle.

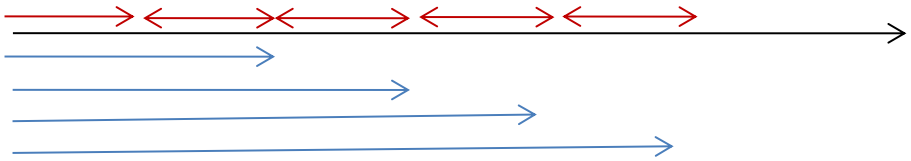
FIELDS

- There exist lists of required fields (see Table at the end):
    - Constant fields. NL switches (with default) control which fields are needed:. lz0local (F), lvegadapt (T), lstomata (F), lconstvegalb (T), lgettcl (F). Not all switches are implemented yet or there is no correct coding or shortName definitions are missing.
    - Initial conditions. Besides T_SO/ W_SO (alternative: fields from old 2 layer scheme) other surface/ground fields are read in including FR_LAND and SOILTYP. If missing, there are defaults set. Controlling switch is lmulti_in (T). It is not clear to me, why T_G is required (new implemented by GDM) as it is content of met. forcing data.
    - Meteorological forcing data. ntype_atminput controls the input of atmospheric wind, temperature, pressure and moisture data at the surface (2m, 10m, lowest model level). For ntype_raininput=1 precipitation is read in (ntype_raininput=2: observations are taken); ntype_radinput controls the input of the radiation fields including T_G. It is **new**, that missing precipitation or radiation fields are set to zero.

TIMES

- NL METFORCING parameter lhourly_data, default is .FALSE., stands for hourly data of precipitation and radiation (GRIB input). It is not checked, when lhourly_data=.TRUE., if this is correct (check GRIB meta data). It is assumed (!) that this is true.
- It seems, that every hour it is tried to read a new forcing file. Therefore it is assumed in read_grib, that the forecast time is in hour (no check of time unit) – nave=ipds18 (vv). At the start (n=0), 2 files are read in for ydate_ini and ydate_ini+1hour.
- Time checks are only by file name, which means that it is believed that the content of a file has the time stamp written in the file name.
- There are always two times available (old, new; new –old = 1hour)); between these times there is a linear interpolation for the fields wind, temperature, moisture and pressure. For precipitation and radiation rates, the average for that time interval is

taken. It seems to be possible, that radiation/precipitation data valid exactly for this interval are provided (hourly data) or data valid for forecast/nudging times being an average or accumulation since forecast/nudging start.

time axis with 

hourly data in red, else (accum or avg since start) in blue. To get the correct average rate for the appropriate hour, the (red) hourly data are taken or for the (blue) data since start the appropriate difference has to be taken.

THINGS to THINK of

- CHECKS of grid / time stamp: Are the input fields sufficiently checked or should there be other/additional checks?
- CHECK of missing fields: Should there be an abort if certain fields are missing? Or should there be a presetting to a certain value? How can it be achieved that there is no calculatio with undefined fields?
- Physical context: TOT_RAIN/TOT_SNOW ? Rates or amounts, instant or averages for radiation? What about runoff definitions?

TSA TABLE: List of required fields according to NAMLIST parameters

(some with defaults, if missing)

Field (shortName)	Required (checked)	Additional per NAMELIST parameter	Default for field	Remarks
I. constant fields *(read_const_fields)*				
RLON	✓			)needed if fields have
RLAT	✓			)to be interpolatd
HSURF	✓			
FR_LAND	✓			
SOILTYP	✓			
Z0	✓			
Z0 (alternative)		lz0local (default:F)		Not implemented yet, no shortName !!
PLCOV_MN	✓	lvegadapt (default: T)		
PLCOV_MX	✓	lvegadapt (default: T)		
LAI_MN	✓	lvegadapt (default: T)		
LAI_MX	✓	lvegadapt (default: T)		
PLCOV (alt.)		lvegadapt=.FALSE.		
LAI (alt.)		lvegadapt=.FALSE.		
ROOTDP	✓			
RSTOM_MN		lstomata		Set to .FALSE. by GDM in "init _variables"
RSTOM_MX		lstomata		"
RSTOM (alt.)		lstomata		No shortNames !!
"VEGALB" = albedo of vegetation ! (code for soil resistence?)		.NOT. lconstvegalb		Not implemented, no shortName !!
T_2M_CL (GRIB2) T_SO(clim.lev)		lgettcl (default: F) + lmulti_layer		"T_CL": Normally taken from initial condition file
T_CL_M		lgettcl + .NOT.lmulti_layer		"
II. initial conditions *(read_initial_fields)*				
T_G (Initiated by GDM; **WHY?** Is read in by read_metforc)	✓		Use tgcom	If missing, set default. "tgcom" needs T_SNOW, W_SNOW and T_SO(0), FR_LAND
W_I	✓			
QV_S	(✓)		0.0	If missing, set default.
W_SNOW	✓			
RHO_SNOW	(✓)		250.	If missing, set default.
FRESHSNW	(✓)		0.6	If missing, set default.
T_SNOW	✓			
FR_LAND				Read again, needed if new area compared to constant fields.

SOILTYP				"
T_SO	✓	lmulti_in (default:T)		lmulti_layer =.TRUE
W_SO	✓			
T_S		lmulti_in=.FALSE.		
T_M				
T_CL_M				
W_G1				
W_G2				
W_CL				
TCH	(✓)		1.0	If missing, set default.
TCM	(✓)		1.0	If missing, set default.
III. meteorological forcing data *(read_metfor + read_lmgrib)* For every NL par. (the defaults are highlighted) the required fields are coloured in a block, e.g. one blue, one orange, one brown and the green block are needed!				
U		ntype_atminput=1		) if ke_model>0 then for
V		"		) lowest model layer,
T		"		) else no level
QV		"		) constraint ?? 💣
PS		"		) **Only option for ICON?**
SP_10M		ntype_atminput=2		Data from **nudging** run
T_2M		"		"
TD_2M		"		"
PS		"		"
SP_10M		ntype_atminput=3		Data from **analysis**
T_2M		"		"
RELHUM_2M		"		"
PS		"		"
RAIN_GSP		ntype_raininput=1		Amounts or rates?!
RAIN_CON		"		**NEW**: If missing,
SNOW_GSP		"		fields
SNOW_CON		"		are set to zero
TOT_PREC		"		**ICON**: alternative to components
SOBS_RAD/ASOB_S		ntype_radinput=1		Average or instant ?!
THBS_RAD/ATHB_S		"		**NEW**: If missing, fields
ALB_RAD		"		are set to zero
T_G		"		Is computed if missing???
SOBS_RAD/ASOB_S		ntype_radinput=2		
THBS_RAD/ATHB_S		"		
ALB_RAD		"		
T_S		"		T_G is computed
T_SNOW		"		
W_SNOW		"		
PABS_RAD/APAB_S		lpar=.TRUE.	PAPS_RAD=0.5 SOBS_RAD	If missing, use default (lpar=.FALSE.)
PABS_RAD/APAB_S		lpar=.FALSE.	PAPS_RAD=0.5 SOBS_RAD	

(✓) : Field required; if missing, use default.