

How-To use the ECHAM5-SCM

Rebekka Posselt

rebekka.posselt@env.ethz.ch

Andreas Rhodin

andreas.rhodin@dwd.de

May 8, 2008

Contents

1	Introduction	1
2	Set-up	2
2.1	Run-Script	2
2.1.1	Original set-up and description (Andreas Rhodin)	2
2.1.2	CHANGED and EXTENDED set-up (Rebekka Posselt)	4
2.2	Forcing file	6
2.3	Tools	7
3	Implementation	8
3.1	mo_column.f90	8
3.1.1	Initialization	8
3.1.2	Providing the forcing terms and performing the temporal integration	8
3.2	mo_relaxation.f90	9
3.3	columnctl.inc	10

1 Introduction

The Single Column Model (SCM) is not a separate model, but the full model run in a special mode. The implementation makes use the facilities of the parallel code to operate only on part of the domain, in this case at one grid-point only.

The SCM version only considers the vertical axis (1D), so horizontal processes and horizontal tendencies cannot be calculated. These terms can be derived by two different ways:

1. by comparison of the results from the SCM run with those from a preceding 3D-model run. The differences are added to the terms calculated by the SCM in subsequent runs so that without any changes to the code the 3D-model trajectory is reproduced. Now the code may be changed in order to investigate effect of modified parameterizations, assuming that the tendencies due to horizontal processes remain unaffected.
2. by obtaining thermodynamic profiles and large-scale tendencies from field campaigns (e.g., ARM IOP's) (usually with the help of mesoscale models (e.g., RACMO, COSMO)) so that comparisons and evaluations of the SCM results with observations are possible.

To run ECHAM in SCM mode the usual input files are required (initial or restart files). Use of restart files is recommended, because the fields in the initial files are not well balanced. The SCM is activated by setting parameters in the `&COLUMNCTL` namelist. Output of any quantity used in the physical parameterizations can easily be performed.

2 Set-up

2.1 Run-Script

2.1.1 Original set-up and description (Andreas Rhodin)

The settings for the SCM are passed to the model via the `&COLUMNCTL` namelist. To provide the minimum functionality only the keywords `comode`, `lat_1d` and `lon_1d` are required.

`comode= (default: ' ')`

The character string `comode` specifies the mode the model is run in. Valid values are:

- ' ': The SCM is not run. A usual 3D-model run is performed. This has the same effect as if the namelist were not present.
- '**traject**': A 3D-model run is performed. The forcing fields required by a subsequent SCM run are written to the file `forcing`.
- '**resid**': A SCM run is performed. For each time step the forcing fields from a 3D-model run are read from the file `forcing` and compared to the respective terms calculated by the SCM. The `residui` (3D1D model results) are added to the column model values so that it follows the same trajectory as the 3D-model. They are also written to the file `residui` to be used by a subsequent SCM run.
- '**add**': A SCM run is performed. For each time step the `residui` fields produced by a preceding SCM run (with `comode='resid'`) are read from file `residui` and added to the values calculated by the SCM so that it follows the same trajectory as the 3D-model as long as no parameterizations are changed.
- '**free**': The SCM is run without forcing or correction terms added. In this case the forcing file must provide only one (or `nforce`, see below) time steps for the SCM initialization.
- '**force**': A SCM run is performed. For each time step the forcing fields from a 3D-model run are read from the file `forcing` to overwrite the values calculated by the SCM, so that it is reset to the 3D-model trajectory.

`nforce= (default:1)`

During the first `nforce` time steps of a SCM integration the variables are overwritten by those from the forcing file, even if `comode` is set to `add` or `free`. If `comode='free'` is specified `nforce` must be at least 1, because the SCM cannot retrieve all its initial values from the restart or initial files.

`lat_1d=, lon_1d=`

Specify the latitude and longitude index of the grid-point the SCM is run at (`comode='resid'`, `'add'`, `'free'`, `'force'`). In mode `comode='traject'` up to 100 longitude/latitude index pairs may be provided to specify the locations the forcing file is written for.

`lat_1db=, lon_1db=`

Only important for `'traject'` mode. One longitude/latitude index pair may be given to specify a region (min, max index) the forcing file is written for. These namelist variables may be used instead of `lat_1d`, `lon_1d` in `'traject'`-mode.

`for_xxx=0,1 or 2 (default:cf. Table 1)`

These flags can be used to specify individually which variables are written to or read from the forcing file (cf. Table 1).

0: the variable is neither read nor written to the forcing file.

name	meaning	type	flag	default
required at model start if vertical coordinates (VCT) are changed				
Q	specific humidity	prognostic variable	for_q	(0)
XL	cloud water	prognostic variable	for_x	(0)
XI	cloud ice	prognostic variable	for_x	(0)
XT	tracer	prognostic variable	for_xt	(0)
applied after spectral computations				
T	temperature	prognostic variable	for_t	(1)
U	wind component	prognostic variable	for_uv	(1)
V	wind component	prognostic variable	for_uv	(1)
ALPS	log surface pressure	prognostic variable	for_ps	(1)
DTM	temperature gradient	term in prognostic equation	for_dt	(for_def)
DTL	temperature gradient	term in prognostic equation	for_dt	(for_def)
VO	vorticity	term in prognostic equation	for_vo	(for def)
D	divergence	term in prognostic equation	for_d	(for def)
DALPSL	log surface pressure gradient	term in prognostic equation	for_dp	(for_def)
DALPSM	log surface pressure gradient	term in prognostic equation	for_dp	(for_def)
UO,UL,DUO	zonal means	not used	for_zm	(0)
applied after transport calculations				
QE	specific humidity	tendency	for_tr	(for_def)
XE	specific humidity	tendency	for_tr	(for_def)
XTEnnn	tracer	tendency	for_tr	(for_def)
TE	temperature tendency	additional tendency term	for_te	(0)
ALPS TR	log surface pressure	prognostic variable	for_trp	(for_def)
ALPSM1	log surface pressure (t- Δ t)	prognostic variable	for_trp	(for_def)

Table 1: Fields read or written by the SCM. Additional fields may be added for diagnostics by calling the routine write column. The prognostic variables T, U, V, ALPS are processed in any case. The use of the other variables can be switched on or off by setting the respective flags (0 or 1). Most variables are used for calculating contributions to the prognostic equations (calculated in subroutine dyn). Other provide tendencies or adjustments due to the transport scheme. The impact of switching off these contributions is shown in a example below.

- 1:** the variable is written to the forcing file (in mode 'traject') and read and modified during a SCM run (in mode 'force' or 'add').
- 2:** the variable is written to the forcing file (and to the result files as well) but not modified during a SCM run. This option is useful if additional variables shall be investigated (plotted) from a SCM run which were not present in the forcing file.

Note: All fields used in subsequent SCM runs (`for_xxx=1`) must have been written to the forcing file before. The settings during the SCM run with `comode='add'` must be the same as those with `comode='resid'`.

ug_1d, vg_1d= (default:0)

The contribution of the pressure gradient term to the prognostic equation for the wind field is calculated in spectral space and cannot be considered by the SCM. In `comode='force'` or `comode='add'` mode this term is considered by the residui or forcing fields. In `comode='free'` mode these namelist variables may be used in order to prescribe the pressure gradient in terms of geostrophic wind components.

file	written by	read by	contents
forcing	3D-model	1D-model	3D model trajectory
residui	1D-model	1D-model	differences 3D-1D trajectory
result1	1D-model	1D model	results before corrections applied
result2	1D-model	1D model	results after corrections applied

Table 2: Files used by the SCM. Presently unformatted files are used to store the forcing and residuum files. NETCDF should be used in the future. Files forcing and residui are used to drive the SCM. Files result1 and result2 are written only for diagnostic purposes.

nudguv(nlev), nudgt(nlev), nudgp (default:0)

In 'add' mode, in addition to adding the residui, the variables u-wind, v-wind, temperature and log surface pressure may be relaxed to the values present in the forcing file with the weights given by these weighting coefficients.

k_vg= (default:0)

This Keyword is used to prescribe a profile for the geostrophic wind components. Above model level `k_vg` the geostrophic wind is set to the wind field at the last forcing time step (`nforce`). Below that level the geostrophic wind is set to the value at level `k_vg`.

Note: Setting `k_vg`, `ug_1d`, `vg_1d` $\neq 0$ makes sense only in `comode='free'` mode.

uvfac= (default:-1)

`uvfac=0` if `u`, `v` from the forcing file are not scaled by `gl_sqcst` (Corioles-Parameter), i.e. if real winds are used to run the model. Within the SCM the wind speeds are then scaled by the Corioles-Parameter `gl_sqcst`.

lrewind= (default:.false.)

When set to `.true.`, the forcing and the residui file is rewound at each time step before reading. A constant forcing may be achieved with this option.

lvctch= (default:.false.)

When set to `.true.`, the Vertical Coordinate Table (VCT) in the forcing file may be changed in order to run the SCM with a different vertical resolution. In order to work properly the namelist variables must be set to 1 and thus be available in the forcing file: `for_q`, `for_x`, `for_xt`.

2.1.2 CHANGED and EXTENDED set-up (Rebekka Posselt)

If the SCM is supposed to run by initial and boundary conditions obtained from field campaigns or other models it is sometimes necessary to force the SCM with tendencies only or to apply a nudging (e.g., every 3 h when measurement were taken). This is not possible with the original version of the SCM. Thus, the `&COLUMNCTL` (and the `mo_column.f90`) is extended, so that it is possible to force or nudge the SCM. The extended version only applies to the following variables: `T`, `Q`, `U`, `V`, `PS`, `XL`, `XI` plus the according tendencies. The rest of the possible forcing variables (cf. Table 1) are treated by the original set-up. The new set-up was developed for `comode='force'`.

usecol= (default:.false.)

This switch decides whether to use the original way to run the SCM (`.false.`, see above: `for_xxx`) or the extended version (`.true.`).

`col_xxx= (default: 0,0,)`

Treated variables include `xxx=ps,t,q,uv,x`. A pair of numbers is passed to the model. Thereby, the first number N_1 set the mode/type of the SCM simulation, whereas, the second number N_2 determine the treatment of the according tendencies.

first number N_1 : mode/type of the SCM simulation

0: is similar to the original `comode='force'`. The variable is read every time step from the forcing file and the actual model values are overwritten.

1: A forcing with the large-scale tendencies is applied. The variables are read in only for `nforce` time steps. The tendencies calculated in the model are substituted by the forcing tendencies (according to the settings of the second number N_2).

>1: This initiates a nudged simulation. The nudging time step τ is determined by $N_1 \times \Delta t$. The tendency treatment is again determined by N_2 . A nudging tendency is added to the model tendencies.

second number N_2 : treatment of tendencies

0: Only the horizontal tendencies are given in the forcing file. The vertical forcing is given by the model. Both tendencies are added.

1: Horizontal and vertical tendencies are given in the forcing file. The model tendencies are overwritten.

else: No tendencies are read in. Only the tendencies from the model are used. With `col_xxx=1,-1` a 'free' run for the considered variable is achieved.

`col_nud= (default:1)`

Determines the type of nudging. Additional treatments can easily be added to *calc_relax* (in *mo_relaxation*)

1: Newtonian relaxation of variable X with relaxation time $\tau = N_1 \Delta t$. Add a nudging tendency to the tendency calculated by the SCM to relax the model values to the observations. The larger τ the weaker the relaxation become. Typical values for τ are 1 – 3 h.

$$\left. \frac{\partial X}{\partial t} \right|_{nud} = \left. \frac{\partial X}{\partial t} \right|_{SCM} + \frac{X_{obs} - X}{\tau} \quad (1)$$

2: Weighted average to obtain new variable value. No additional tendency from nudging. Included for test reasons only!

$$\left. \frac{\partial X}{\partial t} \right|_{nud} = \left. \frac{\partial X}{\partial t} \right|_{SCM} \quad X|_{nud} = \frac{X_{obs} \Delta t + X(\tau - \Delta t)}{\tau} \quad (2)$$

else: Prescribe variable. No additional tendency from nudging.

$$\left. \frac{\partial X}{\partial t} \right|_{nud} = \left. \frac{\partial X}{\partial t} \right|_{SCM} \quad X|_{nud} = X_{obs} \quad (3)$$

`set_slm= (default:-1)`

Set/change the Land-Sea-Mask for the SCM column.

>0: SLM is set to the specified value. If `set_slm` is greater than 1 it is set to 1.

<0: The SLM from the model is used.

2.2 Forcing file

The format of the forcing (as well as the residui and result file) is unformatted Fortran. The postprocessing utility *chua* (see Sect. 2.3) may be used to convert it to formatted (ASCII) and back. The forcing file may be manipulated in order to prescribe a forcing or residui stemming from another source than the 3D ECHAM model (e.g., other models, observations, ...).

The information for each variable within the forcing file consists of a header with name (*character*16*), rank and size of each dimension (generally *lon,lev,lat*) of the array, followed by the data (in Fortran storage order) with the uppermost level being the first value. The formatted version looks as follows (for one column of a 19 layer ECHAM5 forcing file):

```
NSTEP          1 1 0 0          ! timestep (1)
0.0000000000000000
ILON_JLAT      2 1 2 0          ! lon,lat index (1,2)
3.0000000000000000 7.0000000000000000
DTIME          2 1 1 0          ! time step increment (1,1)
2400.000000000000
AK             2 1 20 0         ! vertical coefficient table AK (1,20)
0.0000000000000000 2000.0000000000000 ...
BK            2 1 20 0         ! vertical coefficient table BK (1,20)
0.0000000000000000 0.0000000000000000 ...
ALPS          2 1 1 0          ! log surface pressure (1,1)
11.5031042431194
T             3 1 19 1         ! temperature (1,19)
208.475218772137 210.814337925290 210.661626850253 211.816094269480
215.250933186430 216.483978410409 217.702115863208 220.455787209049
227.804338494870 237.847872950005 247.713113169838 255.573587455114
262.506973124186 266.371413040499 271.769232444394 275.522679939573
277.106050882993 277.120185422363 278.258184846864
U             3 1 19 1         ! wind speed U (1,19)
...
```

The information for each variable in the unformatted file consists of one record with name (*character*16*), rank (*real*8*) and size of each dimension (*real*8(3)*) and one record with the data (*real*8*).

The forcing file written by the 3D ECHAM may hold information for more than one column. In this case the first index of the arrays is larger than 1.

When preparing the forcing file from observation one has to take care of the correct sequence, shape and units of the variables. The forcing file should at least contain the values for the surface pressure, temperature, moisture and wind profiles.

```
NSTEP          1 1 0 0          ! timestep (1)
ILON_JLAT      2 1 2 0          ! lon-lat [as ECHAM grid points]
DTIME          2 1 1 0          ! time step length [s]
AK             2 1 32 0         ! vertical coefficient table AK
BK            2 1 32 0         ! vertical coefficient table BK
ALPS          2 1 1 0          ! log surface pressure
PS+           2 1 1 0          ! log surface pressure tendency
T             3 1 31 1         ! Temperature [K]
U             3 1 31 1         ! wind speed U [m/s]
V             3 1 31 1         ! wind speed V [m/s]
Q             3 1 31 1         ! specific humidity Q [kg/kg]
XL            3 1 31 1         ! cloud water content XL [kg/kg]
```

XI	3 1 31 1	! cloud ice content XI [kg/kg]
VO	3 1 31 1	! vorticity [1/s]
D	3 1 31 1	! divergence [1/s]
SST	2 1 1 0	! sea surface temperature [K]
PT	3 1 32 1	! pressure tendency large-scale subsidence [Pa/s]
VOM	3 1 31 1	! meridional horizontal wind tendency [m/s/s]
VOL	3 1 31 1	! zonal/longitudinal horizontal wind tendency [m/s/s]
TTE	3 1 31 1	! temperature tendency [K/s]
QE	3 1 31 1	! moisture tendency [kg/kg/s]
XLTE	3 1 31 1	! cloud water tendency [kg/kg/s]
XITE	3 1 31 1	! cloud ice tendency [kg/kg/s]

To obtain the correct sign of the large-scale vertical velocity within the model the large-scale subsidence PT should be used so that a subsidence is negative and rising is positive (the same direction as if the vertical velocity would have been in m/s and not in Pa/s)! Furthermore, PT has to be given on the full levels (i.e., 32 instead of 31 levels).

With the new ECHAM5.5 it sometimes happens that the SCM has problems if AK and BK are given. As long as the standard level definitions are used (L19 or L31,lvctch=.false.) these parameters are not necessary and can be removed from the forcing file.

2.3 Tools

Some postprocessing utilities are available in /echam5/post/column. For the installation read the README file within that directory. With every new version of ECHAM the post-processing tools should be recompiled to make them available. The make process puts the executables in the *data*-directory.

post_1d

USAGE: `post 1d | gnuplot`

Visualization script based on gnuplot. `post 1d` should be called from the directory /echam5/data, piping its standard output to gnuplot. The contents of the files used by the column model (Table 2) may be displayed (time series at one level or vertical cross sections at one time). In addition the time increment calculated by the SCM ($1d_{tend} = result1(t)\Delta result2(t\Delta 4t)$) is displayed.

chua

The (unformatted) forcing file may be converted to ASCII and back again. USAGE: `chua 'file name'` The name of the formatted file will be 'file name.asc'. If 'file name' is not present, but 'file name.asc', the latter will be generated from the former. If the unformatted file holds multiple columns, only one column will be converted. the program will ask for the column indices.

chvct

The vertical coordinates (Vertical Coordinates Table) of the forcing file may be changed. The fields are interpolated accordingly.

USAGE: `chvct ifile ofile table`

The forcing file 'ofile' will be generated from 'ifile' using the vertical coordinates table 'table'. An example VCT is given in 'ab_ ecwf 32'. The new VCT is written to the forcing file. The fields will be interpolated linearly. In order to work properly the namelist variables must be set: `for_q=1, for_x=1, for_xt=1` while writing the forcing file and running the SCM. `lvctch=.true.` while running the SCM.

Additional diagnostics

In order to write additional fields to the SCM files (for diagnostics only, no forcing is applied) the generic subroutine `write_column` (defined in module `mo_column`) may be called with the following interfaces:

```
call write_column (x(:,:,:), name)      (lon,lev,lat)-fields.  
call write_column (x(:,:) , name)      (lon,lat)-fields.  
call write_column (x(:,:) , name, jlat) (lon,lev)-fields in row buffer scan.  
call write_column (x(:)   , name, jlat) (lon)-fields in row buffer scan.
```

`x` is the field to be plotted for diagnostics. `name` must be a unique character string (16 significant characters), different from those given in Table 1. `jlat` is the latitude index (local with respect to the processor element in parallel mode). This interface must be used if the routine is called from within the latitude scan (from the physical parameterization routines).

3 Implementation

3.1 `mo_column.f90`

All SCM specific routines are gathered in the module `mo_column`.

3.1.1 Initialization

Two specific initialization routines are called from routine `init_decomposition`. The routine `decompose` has got two optional parameters `lats_1d`, `lons_1d` to be passed in case of a column model run.

CALL `inicolumn` (`lcolumn`): This routine reads the namelist `COLUMNCTL` and stores the values read in `mo_column`. The logical intent(out) parameter `lcolumn` indicates, whether ECHAM shall run in SCM mode.

CALL `decompose` (... , `lats_1d=lat_1d`, `lons_1d=lon_1d`): If `inicolumn` returned `lcolumn=.true.` then subroutine `decompose` has to be called with the optional parameters `lats_1d` and `lons_1d`. The actual values `lat_1d`, `lon_1d` are public variables in `mo_column`.

CALL `setcolumn`: This routine is called to perform the final initialization of module `mo_column` after subroutine `decompose` has been called.

CALL `resetcolumn`: This routine should be called in order to deallocate the allocatable arrays in `mo_column` and to close the SCM specific files.

3.1.2 Providing the forcing terms and performing the temporal integration

Three routines are called from routine `scan1` in order to read or write the terms which cannot be calculated by the SCM (cf. Table 1) or to perform temporal integration otherwise performed in spectral space.

CALL `get_col_ffti`: To be called after the spectral computations (*fourier_to_gridpoint*) to provide updated values of the prognostic variables (temperature, wind components, log. surface pressure) or terms used in the prognostic equations (vorticity divergence, horizontal gradients of temperature, log. surface pressure). This routine also updates the geostrophic wind as indicated by the namelist variables `ug_1d`, `vg_1d`, `k_vg`.

CHANGES for `usecol=.true.`: variables (PS, T, Q, U, V, XL, XI) can be nudged.

CALL `get_col_tran`: To be called after the transport scheme calculations. This routine provides updated values of the moisture, cloud water and tracer tendencies as well as updated surface pressure (old and actual value).

CHANGES for usecol=.true.: moisture and cloud water/ice tendencies can be read in as only horizontal or as horizontal+vertical tendencies

CALL `get_col_dyn`: To be called after the spectral computations (*fourier_to_gridpoint*) and before the transport scheme calculations. This routine provides updated values of the horizontal wind and temperature.

CHANGES for usecol=.true.: horizontal wind and temperature tendencies can be read in as only horizontal or as horizontal+vertical tendencies

CALL `cal_col_exp1 (ztodt, jlat)`: To be called before the spectral computations (*sil*). This routine performs explicit integration of temperature, log. surface pressure and wind components if the SCM is run:

```
u_scb    (:,:,jlat) = um1    (:,:,jlat) + ztodt * (vom - zcorio*vg(:,:,jlat))
v_scb    (:,:,jlat) = vm1    (:,:,jlat) + ztodt * (vol + zcorio*ug(:,:,jlat))
t_scb    (:,:,jlat) = tm1    (:,:,jlat) + ztodt * te
alps_scb (:, jlat)  = alpsm1(:, jlat) + ztodt * alpse
```

The updated values are stored in the scan buffer to be used in the next time step. If the forcing flags `for_XXX` (cf. Table 4.1) are set to zero, the respective prescribed fields (DTM, DTL, VO, D, DALPSL, DALPSM) are set to zero, i.e. these terms are not considered in the prognostic equations. Otherwise the values stay on their last value provided or updated from the forcing or `residui` file, respectively.

3.2 `mo_relaxation.f90`

`calc_relax (varntn, varn, var, relaxtime, TYPE)`

This routine does the calculation of the additional nudging tendency. Used variables include:

- `varntn` - additional tendency due to relaxation (OUT)
- `varn` - observed values (IN)
- `var` - model values (INOUT)
- `relaxtime` - relaxation time step (=nrelaxtime*time_step_len) (IN)
- `TYPE` - relaxation type (IN) (can be extended)
 - 1) Newtonian relaxation
 - 2) weighted average
 - def) prescribe

`get_forcing (var, varntn, varn, colvar, nforce, varchar, type)`

Obtain the variable profiles/values from the forcing file according to the chosen type/mode. As long as `nforce>0` the profiles will be prescribed.

- `var` - model variable (changed for "prescribe") (INOUT)
- `varntn` - additional tendency due to relaxation (changed for "nudge") (INOUT)
- `varn` - observed variable (used for "prescribe" and "nudge") (IN)
- `colvar` - determines SCM mode (see 2.1.2) (IN)
 - 0) "prescribe" variables
 - 1) "force" with variable tendencies only
 - 2,3,...,N) "nudge" with nrelaxtime=N

- varchar - variable name (character) (IN)

get_tendency3 (varte, varth, colvar)

This subroutine is used to obtain the tendency profiles.

- varte - model tendency (INOUT)
- varth - observed tendency (IN)
- colvar - determines tendency type (IN)
 - 0) only horizontal tendencies are given, vertical tendencies from model
 - 1) horizontal and vertical tendencies are given, overwrites internal model tendencies
 - else) only model tendencies are used (= "free" run)

m_relaxinit

Initializes/Allocate the additional variables (varth - nudging tendency, varth - observed tendency, varn - observed variables). Is called in *scan1*. Treatment is similar to that in *mo_scan_buffer* (m_bufscan).

cleanup_relax

Deallocate the additional variables. Is called in *control*. Treatment is similar to that in *mo_scan_buffer* (cleanup_scanbuffer).

3.3 columnctl.inc

Within the *columnctl.inc* the additional settings for the &COLUMNCTL are added.

```

usecol,  &! switch if col_ (T) or for_ (F) is used  (default=.false.)
col_t,   &! INTEGER (2) temperatur                  (default=(1,0))
        !     1: presc(0), force(1), nudge(N)
        !     2: tte hor (0), hor+ver(1)
col_q, col_uv, col_x, col_ps, & ! similar to col_t
set_slm, &! set Land-Sea-Mask for column
col_nud  ! nudging type 1 = Newtonian relax
        !                 2 = weighted mean

```