How to use my prognostic cloud fraction scheme

Short version:

Use branch

```
https://svn.iac.ethz.ch/external/echam-hammoz/echam6-
hammoz/branches/steffen_muench_clean
```

with the following namelist adjustments in PHYSCTL:

```
lCloudMuench = .true.
nic_cirrus = 3
lMidlevelConvectionFix = .true.
lConvPrecipFromBase = .true.
```

The last two are not necessary but used in all my simulations (see below and explanation in paper).

You might want to deactivate my additional output streams (cirrus, cloud, cover) via the namelist to reduce model output:

```
&SET_STREAM stream='cirrus', lpost=0 /
&SET_STREAM stream='cloud', lpost=0 /
&SET_STREAM stream='cover', lpost=0 /
```

That's it!

———

Long version:

I have two relevant ECHAM6.3-HAM2.3 branches:

```
https://svn.iac.ethz.ch/external/echam-hammoz/echam6-
hammoz/branches/steffen_muench
```

and

```
https://svn.iac.ethz.ch/external/echam-hammoz/echam6-
hammoz/branches/steffen_muench_clean
```

For all normal use the clean branch should be fine as I removed unnecessary code parts and variables in it so that it is cleaner and easier to understand. The non-clean branch was used for my papers but the results should be similar with the clean branch; non-bit identical though as the cleaning also involved some replacements of code parts.
The non-clean branch includes some deactivated prototypes that I coded but did not use for the papers (orographic cirrus implementation, marine statocumulus implementation, using

in-cloud and cloud-free water vapor also in radiation calculations). Note however, that they were not used in the papers for a reason as they are not fully consistent with the model design (orocirrus, marine stratocumulus) or the code should be carefully checked (radiation).

Even the clean branch can be further optimised and cleaned: deactivation of custom output streams (cirrus, cloud, cover), code optimizations, …

I implemented the cloud scheme so that it can be activated just by namelist switches. So my branch can be used also for the other cloud schemes. Note, however, that I made some implementation in my branches that are always active (nucleation scavenging of insoluble dust, convection cloud cover, other bugfixes that I documented on redmine).

In the clean branch in mo_cloud_micro_2m.f90 you can find a nice implementation of the latest version of nic_cirrus=3 in the standard 2M scheme.

File structure:
mo_cloud_muench.f90: the main file of the cloud scheme (called from physc.f90)
mo_cloud_muench_cirrus.f90: the cirrus nucleation scheme (nic_cirrus=3)
mo_cloud_muench_outputStream.f90: here I developed a module to easily create custom output streams for cloud variables and I am using it for my custom output streams
Of cause many small adjustments were made in other files. To see all of them, perform a svn diff between the branch and the tag for ECHAM6.3-HAM2.3

The cloud scheme is activated and controlled from the PHYSCTL namelist.

It is activated by the switch lCloudMuench = .true.
Further nic_cirrus = 3 has to be selected as No. 1 and 2 were not implemented
Finally for all my simulations (except REFERENCE) I implemented two modifications to the convection scheme which are activated by lMidlevelConvectionFix = .true. and lConvPrecipFromBase = .true.
I implemented the standard tuning of the published version in the code so that with these lines you get the model state of the paper:

```
lCloudMuench = .true.
nic_cirrus = 3
lMidlevelConvectionFix = .true.
lConvPrecipFromBase = .true.
```

Then I implemented further namelist switches to control the tuning and model behavior (here with standard values):

```
&PHYSCTL
 lcdnc_progn    = .true.
 ncd_activ      = 2
 nactivpdf      = 0
 lconv          = .true.
 lmfmid         = .true.
 ! Activate CloudMuench
```

```
 lCloudMuench   = .true.
 nic_cirrus     = 3
 lorocirrus     = .false.
 ! Tuning of ECHAM
 lMidlevelConvectionFix  = .true.
 lDeepToShallowConv      = .true.
 lShallowConvInhom       = .false.
 lConvPrecipFromBase      = .true.
 ! The following tuning parameters are only used if they are >= 0.0
 tuningRain              =  1.8
 tuningSnow              =  2.0
 tuningConvEntrDeep      = 2e-4
 tuningConvEntrShallow   = 3e-3
 tuningConvEntrMidlevel  = 1e-4
 tuningConvPrecip        = 2e-4
 tuningConvOvershoot      = 0.2
 tuningWaterInhom         = 0.8
 tuningIceInhom           = 0.7
 tuningSedimentation      = 1.0
 tuningCirrusTemp         = 235.15
 tuningMinIceCrystalRad   = 4.0
 ! Tuning of CloudMuench microphysics
 iMixedPhaseFreezingScheme = 4 ! 0=Off 2=Ickes(2017) 4=DeMott(2015)
 lCirrusDustFreezing       = 1
 sedimentationCoverFactor  = 0.5
 rimingFactor              = 1.0
 tune_iceToSnowRad         = 200e-6
 minActivatedCCN = 40.0
 ! Tuning of CloudMuench cloud formation
 tuningRHcritSurface      = 0.9
 lTurbulentCloudForm      = .true.
 tuningTurbulentMixing    = 0.0
 lConvDetrainedCloudForm  = .true.
 tuningDetrainedMixing    = 0.8
 tuningConvICfactor       = 5.0
 ! Additional diagnostics
 lDiagCREs = .true.
/
```

Explanations:

lMidlevelConvectionFix: remove midlevel height limits
lDeepToShallowConv: redefine deep convection to shallow when convection height is less then 200 hPa (standard in ECHAM)
lShallowConvInhom: Use liquid cloud inhomogenity tuning factor of 0.4 for special type of shallow convection clouds (see Mauritsen et al., 2019; for ECHAM6.3); was never implemented for the standard two-moment scheme; with this switch it can be used for all cloud schemes
lConvPrecipFromBase: Allow formation of convective precipitation from convective cloud base and not only after a certain height
tuningRain: Tuning of stratiform autoconversion of cloud water to rain

tuningSnow: Tuning of stratiform aggregation of cloud ice to snow (standard 2M scheme), or only aggregation of ice crystals (prognostic scheme)
tuningConvEntrDeep: Tuning of deep convection entrainment rate
tuningConvEntrShallow: Tuning of shallow convection entrainment rate
tuningConvEntrMidlevel: Tuning of mid-level convection entrainment rate
tuningConvPrecip: Tuning of convection precipitation formation
tuningConvOvershoot: Tuning of convective overshooting
tuningWaterInhom: liquid cloud inhomogenity factor
tuningIceInhom: ice cloud inhomogenity factor
tuningSedimentation: enhancement factor of ice crystal sedimentation velocity
tuningCirrusTemp: temperature for homogeneous freezing of cloud droplets (cirrus cloud temperature)
tuningMinIceCrystalRad: minimum ice crystal effective radius
iMixedPhaseFreezingScheme: Mixed phase immersion freezing scheme (0=Off 2=Ickes(2017) 4=DeMott(2015))
lCirrusDustFreezing: Heterogeneous nucleation cirrus clouds (1=On, 0=Off)
sedimentationCoverFactor: Factor of how sedimentation is interpreted as vertical advection and cloud overlap (see paper for detailed description)
rimingFactor: Riming enhancement factor
tune_iceToSnowRad: Minimum snowflake radius
tuningRHcritSurface: Critical RH for large-scale cloud formation (constant for all altitudes)
lTurbulentCloudForm: Enable cloud formation by turbulent vertical diffusion
tuningTurbulentMixing: Mixing factor of cloud air with environmental air for turbulent cloud formation
lConvDetrainedCloudForm: Enable cloud formation by convective detrainment
tuningDetrainedMixing: Mixing factor of cloud air with environmental air for convective cloud formation
tuningConvICfactor: Tuning factor for immediate aggregation of ice crystals in convective detrainment (see paper for detailed description)
minActivatedCCN: minimum activated CCN (=minimum CDNC at point of cloud droplet activation)
lDiagCREs: Diagnose cloud radiative effects of individual cloud types (liquid T > 0°C, supercooled liquid, mixed-phase ice, cirrus ice) via radiation double calls

Finally, the non-clean branch needs a slightly different namelist (some more options that I removed during the cleanup):

```
&PHYSCTL
 lcdnc_progn    = .true.
 ncd_activ      = 2
 nactivpdf      = 0
 lconv          = .true.
 lmfmid         = .true.
 ! Activate CloudMuench
 lCloudMuench   = .true.
 nic_cirrus     = 3
 lorocirrus     = .false.
 lRadiationIncloudCloudfreeVapor = .false.
```

```
 ! The following tuning parameters are only used if they are >= 0.0
 ! Tuning of ECHAM
 lMidlevelConvectionFix  = .true.
 lDeepToShallowConv      = .true.
 lShallowConvInhom       = .false.
 tuningRain              =  1.8 ! 15.5  (ECHAM-HAM 10.6 ECHAM 15.0)
 tuningSnow              =  2.0 ! 65.0  (ECHAM-HAM 900. ECHAM 95.0)
 tuningConvEntrDeep      = 2e-4 ! 1e-4  (ECHAM-HAM 2e-4 ECHAM 1e-4)
 tuningConvEntrShallow   = 3e-3 ! 3e-3  (ECHAM-HAM 3e-3 ECHAM 3e-3)
 tuningConvEntrMidlevel  = 1e-4 ! 1e-3  (ECHAM-HAM 1e-4 ECHAM 1e-4)
 tuningConvPrecip        = 2.0e-4 ! 2e-4  (ECHAM-HAM 9e-4 ECHAM
2.5e-4)
 tuningConvOvershoot      = 0.2 ! 0.2   (ECHAM-HAM 0.2  ECHAM 0.2)
 lConvPrecipFromBase      = .true.
 tuningWaterInhom         = 0.8
 tuningIceInhom           = 0.7 ! 0.8   (ECHAM-HAM 0.7  ECHAM 0.8)
 tuningSedimentation      = 1.0
 tuningCirrusTemp         = 235.15
 tuningMinIceCrystalRad   = 4.0
 ! Tuning of CloudMuench
 lLiquidVertCover         = .true.
 lLiquidVertCoverTilting  = .false.
 lCloudDissolve           = .false.
 ! Tuning of CloudMuench microphysics
 mixedPhaseFreezingScheme = 4 ! 2=Ickes(2017) 4=DeMott(2015)
 lMixedPhaseImmFreezing   = 1
 lCirrusDustFreezing      = 1
 sedimentationCoverFactor = 0.5
 rimingFactor             = 1.0
 tune_iceToSnowRad        = 200e-6
 ! Tuning of CloudMuench cloud formation
 tuningRHcritSurface      = 0.9
 tuningRHcritTop          = 0.0
 tuningRHcritMarineInv    = 0.0
 nCirrusDistr             = 2
 tuningCirrusStdDev       = 1.0
 lTurbulentCloudForm      = .true.
 tuningTurbulentMixing    = 0.0
 lConvDetrainedCloudForm  = .true.
 tuningDetrainedMixingMin = 0.8
 tuningConvICfactorMin    = 1.0
 tuningConvICfactorMax    = 5.0
 lConvCloudTracer = .false.
 lmfdd = .true.
 minActivatedCCN = 40.0
 lAggregationUseUpdrafts = .false.
 iIceCrystalTuning = 1
/
```

I implemented my own output streams (files: cirrus, cloud, cover) for all my own diagnostics. They are somewhat sorted by topic but not fully consistent. Unfortunately, I did not implement a namelist switch to completely deactivate it, so it can be deactivated via:

```
&SET_STREAM stream='cirrus', lpost=0 /
&SET_STREAM stream='cloud', lpost=0 /
&SET_STREAM stream='cover', lpost=0 /
```

Use the cloud scheme without HAM but prescribed CCN

As used in my second paper, I made it possible to run the standard and the new two-moment cloud schemes without HAM by prescribing CCN. This can also be controlled by namelist.

Create an ECHAM6 experiment (not ECHAM-HAM): prepare_run.sh -t echam6 …

Then in the settings file:

```
&SUBMODELCTL
 lmethox          = .true.
 lham             = .false.
 lmoz             = .false.
 lhammoz          = .false.
 lccnclim         = .true.
/
&RADCTL
 iaero   = 3     ! 1 for interactive (lham=true), 2 for Tanre 3 for
Kinne
/
&PHYSCTL
 lcdnc_progn    = .true.
 ncd_activ      = 1
 lCloudMuench   = .true. ! or false for standard 2M scheme
 …insert all your standard cloud settings…
 iMixedPhaseFreezingScheme = 0 ! 0=Off 2=Ickes(2017) 4=DeMott(2015)
 lCirrusDustFreezing       = 0
 CCNclimTop               =  50e6
 CCNclimOcean             = 150e6
 CCNclimLand              = 350e6
/
```

Note that the tuning for the prescribed CCN runs is not implemented in the code and has to be set by namelist. Please see the supplementary of my ECS paper for the exact values