

Exercises “Working with ICON”

Exercise 5

Problem 14

Add a variable `qtrc_ttr(:, :, :, :)` to the type `t_echam_phy_tend` so that the tracer tendency due to your new tracer submodule can be written to the output. You also need a component `TYPE(t_ptr_3d), ALLOCATABLE :: qtrc_ttr_ptr(:)` in the same type. Adding the tracers to the “stream” has two steps: (i) add `qtrc_ttr` as a whole for all tracers, (ii) add references to individual tracers such that these can be written to an output file for each tracer individually. (i) is done by a call to the `add_var` subroutine (e.g. after the call for `qtrc_vdf`) that stores the tracer tendencies due to vertical diffusion. By tradition, a prefix `prefix` is used in the name that should be `prefix//'qtrc_ttr'`. The shape of the field must be `shape_trc=(/kproma, klev, kblks, ktracer/)`.

For the reference, allocate first the tendency for all tracers:

```
ALLOCATE(tend% qtrc_ttr_ptr(ktracer))
```

e.g. after the allocation of `tend%qtrc_vdf_ptr`. In a loop over all tracers `jtrc=1,ktracer`, use the `add_ref` subroutine to get a reference to individual tracers (e.g. after the call for `vdf`). Syntax:

```
add_ref(<fad>_list, '<name_collective_list>', &  
& '<individual_tracer_name>', &  
& <anchor in list, e.g.> tend%qtrc_ttr_ptr(jtrc)%p, &  
& GRID_UNSTRUCTURED_CELL, ZA_REFERENCE, &  
& cf_desc (for individual tracer), &  
& grib2_desc, shape3d, vert_interp=...)
```

You can now introduce the variable `tend%qtrc_ttr` into `interface_echam_ttr` and use `echam_phy_config(:)%fc_ttr` and `echam_phy_config(:)%lparamcpl` to complete the interface for the case of a diagnostic process and the various operator splitting modes.

Problem 15

Run the program for three months similar to problem 9, write the tracers and their tendencies and look at the results with `ncview`.