

Sample of Fortran coding by Youkyoung Jang

A Fortran program in the below, **add_ttend.F90**, is to apply a new temperature tendency in a global climate model. Its purpose is to correct bias of diabatic heating in the model by adding differences between observations and the model heating. Several experiments with this code have been done and their results are presented at [Virtual Workshop of bias corrections in subseasonal to interannual predictions](#), NOAA, September, 2014.

The name of input data is 'diff_forModel_final.nc' and forcing is **xxh** (correcting heating rate).

Step 1. add_ttend_init()

: called from phys_init subroutine

Read and add **xxh** in an initialization of physics package. Temperature and heating rate before and after adding **xxh** are also added to check up if the forcing (xxh) is properly added.

Step 2. add_ttend_adv(state)

: called from advnce subroutine

Add **xxh** from step 1 to update time interpolation.

Step 3. add_ttend_run(state, ptend, xxh)

: called from tphysac subroutine

Add **xxh** in the temperature tendency equation

Only add_ttend.F90 is given for a requested sample coding. To work out this program in the global model, other programs have been modified to call it. If you are interested in all modified programs, please let me know.

add_ttend.F90

```
module add_ttend
```

```
use shr_kind_mod, only:r8=>shr_kind_r8
```

```
use tracer_data, only: trfld, trfile, trcdata_init, advance_trcdata
```

```
use ppgrid, only: begchunk, endchunk, pcols, pver
```

```
use physconst, only: cpair
```

```
implicit none
```

```
private
```

```
save
```

```
public :: &
```

```
add_ttend_init, &
```

```
add_ttend_adv, &
```

```
add_ttend_run
```

```
type(trfld),pointer :: fields(:)
type(trfile) :: file
```

```
! input data: diff_forModel_final.nc
! xxh: my forcing, heating rate for correction, be added in temperature
tendency
```

```
character(len=256) :: datapath
='/glade/u/home/yjiang/glade/netcdf_test/diff'
character(len=256) :: filename = 'diff_forModel_final.nc'
character(len=256) :: filelist = ''
character(len=32) :: datatype = 'CYCLICAL'
logical :: rmv_file = .false.
integer :: cycle_yr = 2000
integer :: fixed_ynd = 0
integer :: fixed_tod = 0
character(len=32) :: specifier(1) = 'XXH: xxh'
```

```
contains
```

```
subroutine add_ttend_init()
```

```
!
! Should be called from phys_init subroutine in physpkg module
!
```

```
use tracer_data, only : trcdata_init
use cam_history, only : addfld, add_default, phys_decomp
use ppgrid,          only : pcols, pver, begchunk, endchunk
```

```
call trcdata_init(specifier, filename, filelist, datapath, fields,
file, &
    rmv_file,cycle_yr, fixed_ynd,fixed_tod, datatype)
```

```
! add fields in output
! : forcing(xxh), variables (tmp-temperature/s-heating) before and
after adding
! xxh
! saving fields in output by call outfld
! xxh : in add_ttend_adv subroutine in this program
! other fields : in tphysac subroutine in tphysac.F90
```

```
!1.XXH
```

```
call addfld('XXH' , 'K/day', pver , 'A', 'Additional applied
heating', &
```

```
    phys_decomp)
```

```
call add_default('XXH', 1, ' ')
```

```
!2.tmpbefore
```

```
call addfld('tmpbefore' , 'K/day', pver , 'I', 'before heating
applying ', &
```

```
    phys_decomp)
```

```
call add_default('tmpbefore', 1, ' ')
```

```
!3. tmpafter
```

```
call addfld('tmpafter' , 'K/day', pver , 'I', 'after heating
applying ', &
```

```

        phys_decomp)
    call add_default('tmpafter', 1, ' ')
!-----
!4.sbefore
    call addfld('sbefore ', 'K/day', pver, 'I', 'before heating
applying ', &
        phys_decomp)
    call add_default('sbefore', 1, ' ')
!5. safter
    call addfld('safter ', 'K/day', pver, 'I', 'after heating
applying ', &
        phys_decomp)
    call add_default('safter', 1, ' ')
end subroutine add_ttend_init

```

```

subroutine add_ttend_adv(state)
!
! Should be called from advnce subroutine
!
    use tracer_data,    only : advance_trcdatanew
    use physics_types,  only : physics_state
    use ppgrid,         only : begchunk, endchunk
    use ppgrid,         only : pcols, pver
    use cam_history,    only : outfld
    use phys_buffer,    only : pbuf_size_max, pbuf_fld

    implicit none

    type(physics_state), intent(in)    :: state(begchunk:endchunk)

    integer :: c, ncol, i
    real(r8) :: outdata(pcols, pver)

! new subroutine by yjang is called not to have time-vertical
interpolation existing in CESM

    call advance_trcdatanew(fields, file, state)

!$OMP PARALLEL DO PRIVATE (C, NCOL, OUTDATA)
    do c = begchunk, endchunk
        ncol = state(c)%ncol
        outdata(:, ncol) = fields(1)%data(:, ncol, c)
        call outfld('XXH', outdata(:, ncol), ncol, state(c)%lchnk)
    enddo
end subroutine add_ttend_adv

```

```

subroutine add_ttend_run(state, ptend, xxh)
!
! Should be called in tphysba or tphysac.

```

```

! A Call to physics_update should follow it to apply the tendencies set
here to the state
!
  use physics_types,    only : physics_state, physics_ptend,
physics_ptend_init
  use ppgrid,          only : pcols, pver

  type(physics_state),intent(in) :: state
  type(physics_ptend),intent(out) :: ptend

  integer :: c, ncol, i, lchnk
  real(r8) :: xxh(pcols,pver)

  lchnk = state%lchnk
  ncol = state%ncol

  xxh(:,ncol,:) = fields(1)%data(:,ncol,:,lchnk)

end subroutine add_ttend_run
end module add_ttend

```