

PROGRAM READ_INTEX

! Description:

! - Read profiles in file 'diverse_52profiles_601.dat'
! + The profiles are provided on 101 pressure levels
! - Interpolate profiles on user-specified levels (pres_fin(nlevels_fin))
! + Interpolation linear in LOG(P)
! - Write output file

! History:

! Version	! Date	! Comment
! 1	! 14/05/2003	! Frederic Chevallier (ECMWF)
! 2	! 30/06/2003	! Axel von Engel (Uni Bremen)
!	!	! -tried to fix header problems
!	!	! -input file name changed to
!	!	! diverse_52profiles_601.dat
! 3	! 30/06/2003	! Input filename changed to
!	!	! diverse_52profiles_101L.dat

! Code description:

! Language: Fortran 90.
! Software Standards: "European Standards for Writing and Documenting
! Exchangeable Fortran 90 code".

! Declarations:

Implicit None

!- Local variables

Integer, parameter :: dr = kind(1.0d0) ! definition of a double precision real
Integer, parameter :: nbatm = 52 ! number of profiles to process
Integer, parameter :: nlevels_ini = 101 ! number of pressure levels in initial grid
Integer, parameter :: nlevels_fin = 43 ! number of pressure levels in final grid
! Please customise !!!

Integer :: jdat, iatm, ilev
Real(dr) :: rlat, rlon, psurf, rlsn
Character(len=3) :: nprof

Real(dr), dimension(nlevels_ini) :: pres_ini, temp_ini ! pressure, temperature
Real(dr), dimension(nlevels_ini) :: spechum_ini, specoz_ini ! specific concentrations
(kg/kg)

Real(dr), dimension(nlevels_fin) :: pres_fin, temp_fin ! pressure, temperature
Real(dr), dimension(nlevels_fin) :: spechum_fin, specoz_fin ! specific concentrations
(kg/kg)

!- Pressure grid to which the atmospheric profile will be interpolated (hPa)

!- Please customise !!!

Data pres_fin / &
0.100, 0.290, 0.690, 1.420, 2.611, 4.407, 6.950, 10.370, 14.810, 20.400,
&
27.260, 35.510, 45.290, 56.730, 69.970, 85.180, 102.050, 122.040, 143.840, 167.950,
&
194.360, 222.940, 253.710, 286.600, 321.500, 358.280, 396.810, 436.950, 478.540, 521.460,
&
565.540, 610.600, 656.430, 702.730, 749.120, 795.090, 839.950, 882.800, 922.460, 957.440,
&
985.880, 1005.430, 1013.250 /

!- End of header -----

!- Open input file

```

! Open(1,file='data52_profs_101L.dat',form='formatted',status='old')
Open(1,file='diverse_52profiles_101L.dat',form='formatted',status='old')

!- Open output file
Open(2,file='data52_profs_43L.dat',form='formatted')

!- Pass header
Read(1,*)
Read(1,*)
Read(1,*)
! Read(1,*)

!- Begin loop on profiles
iatmloop : Do iatm = 1, nbatm
  Read(1,'(a3,i11,4e16.6)') &
    nprof, &                ! profile number (e.g. #01)
    jdat, &                 ! date (yyyymmddhh)
    rlat, &                 ! latitude (deg)
    rlon, &                 ! longitude (deg)
    psurf, &                ! surface pressure (hPa)
    rlsn                     ! land/see mask (1=land, 0=sea)

  if (iatm .eq. 1) Read(1,*)

  Read(1,'(4e16.6)') ( &
    pres_ini(ilev), &      ! Pressure (hPa)
    temp_ini(ilev), &     ! temperature (K)
    spechum_ini(ilev), &  ! specific humidity (kg/kg)
    specoz_ini(ilev), &   ! specific ozone (kg/kg)
    ilev = 1, nlevels_ini )

  ! Interpolate to given pressure grid
  Call rttov_intex(nlevels_ini, nlevels_fin, pres_ini, pres_fin, temp_ini, temp_fin)
  Call rttov_intex(nlevels_ini, nlevels_fin, pres_ini, pres_fin, spechum_ini, spechum_fin)
  Call rttov_intex(nlevels_ini, nlevels_fin, pres_ini, pres_fin, specoz_ini , specoz_fin )

  ! Write output
  Write(2,'(a3,i11,4e16.6)') &
    nprof, &                ! profile number (e.g. #01)
    jdat, &                 ! date (yyyymmddhh)
    rlat, &                 ! latitude (deg)
    rlon, &                 ! longitude (deg)
    psurf, &                ! surface pressure (hPa)
    rlsn                     ! land/see mask (1=land, 0=sea)
!
  Do ilev = 1, nlevels_fin
    Write(2,'(400e16.6)') &
      pres_fin(ilev), &    ! Pressure (hPa)
      temp_fin(ilev), &   ! temperature (K)
      spechum_fin(ilev), & ! specific humidity (kg/kg)
      specoz_fin(ilev)    ! specific ozone (kg/kg)
  Enddo
!- End loop on profiles
Enddo iatmloop

Close(1)
Close(2)

END PROGRAM READ_INTEX

!
Subroutine rttov_INTEX(      &
  & klevi ,& ! in
  & klevf ,& ! in

```

```

& presi ,& ! in
& presf ,& ! in
& veci ,& ! in
& vecf ) ! out

! Description:
! To interpolate the array vec from the presi levels to presf levels
!
! Copyright:
!
! This software was developed within the context of
! the EUMETSAT Satellite Application Facility on
! Numerical Weather Prediction (NWP SAF), under the
! Cooperation Agreement dated 25 November 1998, between
! EUMETSAT and the Met Office, UK, by one or more partners
! within the NWP SAF. The partners in the NWP SAF are
! the Met Office, ECMWF, KNMI and MeteoFrance.
!
! Copyright 2002, EUMETSAT, All Rights Reserved.
!
!
! Method:
! Linear interpolation in ln(P)
!
! Current Code Owner: SAF NWP
!
! History:
! Version   Date       Comment
! -----   ----       -
! 1         09/2002    ECMWF

Implicit None

Integer, parameter :: dr = kind(1.0d0)      ! definition of a double precision real
!
! Subroutine arguments
!
Integer, Intent(in) :: klevi      ! number of levels of the initial grid
Integer, Intent(in) :: klevf     ! number of levels of the final grid
!
Real(dr), Intent(in), Dimension(klevi) :: presi ! initial grid
Real(dr), Intent(in), Dimension(klevf) :: presf ! final grid
!
Real(dr), Intent(in), Dimension(klevi) :: veci ! initial vec array
Real(dr), Intent(out), Dimension(klevf) :: vecf ! final vec array
!
! Local scalars :
!
Integer :: jki, jkf
Real(dr)  :: slope, t1, t2, p1, p2, lp1, lp2
Real(dr), Dimension(klevi) :: lpresi
Real(dr), Dimension(klevf) :: lpresf
!
!- End of header -----

vecf(:) = -1000.
lpresi(:) = Log( presi(:) )
lpresf(:) = Log( presf(:) )

Do jkf = 1, klevf
  Do jki = 1, klevi-1
    p1 = presi(jki)
    p2 = presi(jki+1)
    lp1 = lpresi(jki)
    lp2 = lpresi(jki+1)
  
```

```
    If (presf(jkf) >= p1 .And. presf(jkf) < p2) Then
      t1 = veci(jki)
      t2 = veci(jki+1)
      slope = (t1-t2)/(lp1-lp2)
      If (t2 == 0.) slope = 0.
      vecf(jkf) = t1 + slope*(lpresf(jkf)-lp1)
      !
    Else If (jki == 1 .And. presf(jkf) < p1) Then
      vecf(jkf) = veci(jki)
    Else If (jki == (klevi-1) .And. vecf(jkf) == -1000. ) Then
      vecf(jkf) = veci(klevi)
    End If
  End Do
End Do

Return
End Subroutine rttov_INTEX
```