

---

# CODES FOR CHAPTER 20

---

## **fskdist.f90**

Program fskdist is a Fortran90 code to calculate full spectrum  $k$ -distributions for a number of Planck function temperatures and a single gas property state (temperature, partial and total pressures), for a gas mixture containing CO<sub>2</sub>, H<sub>2</sub>O, CH<sub>4</sub> and soot; weight functions  $a(T, T_{\text{ref}}, g)$  are calculated, as well. The spectral absorption coefficient is either calculated directly from the HITRAN or HITEMP databases, or is supplied by the user. The user should scan the code for OPEN statements, identifying input (HITRAN/HITEMP and/or absorption coefficient) and output files.

Input:

- Tref = reference temperature (temperature of gas for evaluation of absorption coefficient, and also used as reference Planck function temperature), in K,
- Tmin = minimum temperature for which a  $k$ -distribution and weight functions  $a(T, T_{\text{ref}}, g)$  are to be calculated, in K,
- Tmax = maximum temperature for which a  $k$ -distribution and weight functions  $a(T, T_{\text{ref}}, g)$  are to be calculated, in K,
- numT = number of different temperatures to be considered; equally spaced between Tmin and Tmax,
- P = total pressure of gas mixture, bar,
- xmfr(3) = mole fraction vector; xmfr(1)= mole fraction of CO<sub>2</sub>, xmfr(2)= mole fraction of H<sub>2</sub>O, xmfr(3)= mole fraction of CH<sub>4</sub>; note that for any xmfr < 10<sup>-3</sup> the specie is neglected.
- fvsoot = volume fraction of soot,
- nsoot, ksoot = complex index of refraction for the soot; its absorption coefficient is assumed linear in wavenumber, using
- wvnm\_b = minimum wavenumber considered, cm<sup>-1</sup>,
- wvnm\_e = maximum wavenumber considered, cm<sup>-1</sup>,
- wvnmst = wavenumber step (equally spaced) with which the absorption coefficient for the mixture is calculated from the HITRAN or HITEMP database, cm<sup>-1</sup>,
- kdmin = minimum  $k$ -value to be considered for  $k$ -distribution, cm<sup>-1</sup>,
- kdmax = maximum  $k$ -value to be considered for  $k$ -distribution (kdmax ≤ 0 sets kdmax=kmax, i.e., the maximum absorption coefficient found across the spectrum), cm<sup>-1</sup>; allows for globally fixed  $k$ -values independent of  $k$ -distribution (useful for mixing),
- n\_pwrk = number of different  $k$ -bin values considered in the construction of the  $k$ -distribution,
- pwr = exponent for  $k$ -bin values spacing:  $k$ -bins are equally spaced in  $k^{\text{pwr}}$  between kdmin and kdmax.
- nq = number of quadrature points for radiative calculations, i.e., the number of  $(k, g)$ -pairs desired for RTE evaluations to be performed before spectral integration (over cumulative  $k$ -distribution  $g$ ),
- iwr = absorption coefficient switch: iwr=0 to make a single complete run, i.e., evaluating  $\kappa_\eta$  from HITRAN or HITEMP (without storing them), followed by generation of  $k$ -distributions, iwr=1 same but absorption coefficient is stored for future use, and iwr=2: precalculated absorption coefficients are read in and  $k$ -distributions are generated.
- ipl = linear vs. pressure-based absorption coefficient switch:  
 ipl=0: calculate linear absorption coefficient, in cm<sup>-1</sup>  
 ipl=1: calculate pressure-based absorption coefficient (allowed only for single absorbing gas!), in cm<sup>-1</sup> bar<sup>-1</sup>; if the pressure-based absorption coefficient for a dilute gas is desired, set xmfr=1.d-3 (=0.1%)
- ipr = output switch: see under output.

Output:

ipr = 1: a single output file is generated containing a header line (formatted for Tecplot), identifying the variables being printed, and n\_pwrk data lines, each with  $2 \times \text{numT} + 2$  numbers:  $k_i, (\text{numT} + 1) \times g(T_j, k_i)$ , and (slightly smoothened)  $\text{numT} \times a(T_j, T_{\text{ref}}, k_i)$  (including  $T_{\text{ref}}$  for  $g$ .)

ipr = 2: in addition to the ipr=1 output file, a second file is generated, containing a header identifying variables, and nq output lines, each with numT+3 numbers:  $w_i$ ,  $g_i(T_{\text{ref}}, k_i)$ ,  $k_i$ , and numT smoothened  $a(T_j, T_{\text{ref}}, k_i)$ -values (averaged over its  $g$ -range).

**Example:**

We consider the full-spectrum  $k$ -distribution for a pressure-based absorption coefficient (ipl=1) of pure H<sub>2</sub>O, for a vanishingly small mole fraction (xmfr(3)=(/0.0d0,1.0d-3,0.0d0/)). Note that  $x_{\text{H}_2\text{O}}$  has been set to  $10^{-3}$ : the code, when accessing HITRAN or HITEMP, will assume a specie not to be present whenever  $x_i < 10^{-3}$ . The absorption coefficient has been calculated in a previous run (iwr=2), and has been stored in file absch2o-0p-2000K.dat (for a wavenumber range from 50 cm<sup>-1</sup> to 12000 cm<sup>-1</sup> with a  $\Delta\eta = 0.005$  cm<sup>-1</sup>). We will calculate the  $k$ -distributions for 5 temperatures: a reference temperature  $T_{\text{ref}} = T_0 = 2000$  K (at which the absorption coefficient has been evaluated) and 4 equally spaced (Planck function) temperatures between  $T_{\text{min}} = 0$  K and  $T_{\text{max}} = 1500$  K (numT=4): this results in the 4 temperatures of 300 K, 500 K, 1000 K and 1500 K (the first temperature is not 0 K, because temperatures below 300 K are not accepted: any temperature below it is set to 300 K). We will use 500  $k$ -bins (n\_pwrk=500) with pwr=0.1 (this spreads the  $k$ -bins over many orders of magnitude, but places more and more bins into large magnitudes; see output file). We also set kadmin=10<sup>-7</sup> (cm<sup>-1</sup> bar<sup>-1</sup>) and kdmx = -20 < 0 (cm<sup>-1</sup> bar<sup>-1</sup>), i.e., we will consider  $k$ -values between 10<sup>-7</sup> and the maximum value found among the absorption coefficient values. Finally, we set ipr=2 and nq=12, i.e., besides the general  $k$ -distributions we want to also generate truncated  $k$ -distributions ready-made for numerical quadrature, using 12 quadrature points. The top of the program with input parameters, therefore, looks like this:

```
MODULE Key
  IMPLICIT NONE
!HITRAN/HITEMP DATABASE
  INTEGER :: lu
  INTEGER,PARAMETER :: rows=1400000
  DOUBLE PRECISION,PARAMETER :: wvnm_b=50.d0,wvnm_e=12000.d0,wvnmst=0.005d0, &
    kadmin=1.d-7,kdmx=-20.d0
  DOUBLE PRECISION :: data(rows,6)
END MODULE Key

PROGRAM Main
  USE Key
! Input parameters
  INTEGER,PARAMETER :: numT=4,n_pwrk=1000,iwr=2,ipl=1,ipr=2,nq=12
  DOUBLE PRECISION,PARAMETER :: P=1.d0,Tref=2000d0,Tmin=000d0,Tmax=1500d0
  DOUBLE PRECISION,PARAMETER :: xmfr(3)=(/0.0d0,1.0d-3,0.0d0/)
  DOUBLE PRECISION,PARAMETER :: klmin=1.d-9,pwr=0.1d0
  DOUBLE PRECISION,PARAMETER :: fvsoot=0.d-6,nsot=1.89d0,ksot=0.92d0
```

where we have changed the values for wvnm\_b, wvnm\_e, wvnmst, kadmin, kdmx, numT, n\_pwrk, iwr, ipl, ipr, nq, Tref, Tmin, Tmax, xmfr and pwr to fit our needs. Also, in this simulation we have set file names as

```
! Set output file name
  character(256), parameter :: kvsgFile='kvsg2o-0p-2000K.dat'
  character(256), parameter :: kvsgqFile='kvsgqh2o-0p-2000K.dat'
  character(256), parameter :: abscFile='absch2o-0p-2000K.dat'
! Open output files
  OPEN(7,FILE=kvsgFile)
! Header formatted for TECPLOT, for a numT of 4
  write(7,6)
6 format('VARIABLES = k,g0,g1,g2,g3,g4,a1,a2,a3,a4')
  IF(ipr==2) THEN
    OPEN(8,FILE=kvsgqFile,STATUS='unknown')
! Header formatted for readability, for a numT of 4
    write(8,8)
  ENDIF
8 format('      wq',9x,'gq',9x,'kq',8x,'aq1',8x,'aq2',8x,'aq3',8x,'aq4')
! File containing absorption coefficient
  IF(iwr>0) OPEN(9,FILE=abscFile,STATUS='unknown')
```

i.e., the previously calculated absorption coefficient is located in absch2o-0p-2000K.dat, while the long  $k$ -distribution output (500 values) will be put into kvsg2o-0p-2000K.dat, and the short, quadrature-ready output into kvsgqh2o-0p-2000K.dat. Note that the header lines for the output files are formatted for numT=4 (see the two format statements above): they will need to be rewritten for different values of numT.

We will also assume that Numerical Recipes subroutines are available, leaving the following lines unchanged:

```

! Selection of g-values for numerical quadrature, using a Numerical Recipes routine
! If Numerical Recipes is not available, set nq=12, comment out the following 6 lines of code,
! and uncomment the 5-line REAL declaration following it
! Get quadrature coefficients from Numerical Recipes
REAL                :: gqs(nq),wqs(nq),kq(nq),aq(numt,nq),gq(nq),wq(nq),gaujac,alf=3.,bet=0.,sum
sum=0.
CALL GAUJAC(gqs,wqs,nq,alf,bet)
do iq=1,nq
  gq(iq)=0.5*(1.-gqs(iq))
  wq(iq)=wqs(iq)/(2.*(alf+bet+1)*gq(iq)**alf*(1.-gq(iq))**bet)
  sum=sum+wq(iq)
enddo
! Correction to make sum(wq)=1
wq=wq/sum
! End quadrature coefficients from Numerical Recipes
! Selection of precalculated g-values for numerical quadrature, for nq=12,alf=3.,bet=0.
! REAL :: kq(nq),aq(numt,nq), &
!   gq(nq)=(/ 5.120075E-02,1.170678E-01,2.015873E-01,3.007074E-01,4.095012E-01,5.225285E-01, &
!   6.341280E-01,7.387071E-01,8.310236E-01,9.064499E-01,9.612060E-01,9.925594E-01)/,&
!   wq(nq)=(/ 5.556622E-02,7.576839E-02,9.258290E-02,1.048306E-01,1.118451E-01,1.132605E-01, &
!   1.090012E-01,9.927844E-02,8.457905E-02,6.563999E-02,4.341329E-02,1.904792E-02)/

```

This will calculate quadrature points  $gq$  and weights  $wq$  using Gaussian quadrature of moments ( $alf=3$  sets 3rd order moments). For users without access to Numerical Recipes the  $gq$  and  $wq$  calculated here have been put in data statements and may be used instead by following the guidelines above.

The previously calculated absorption coefficient in `absch2o-0p-2000K.dat` has the following form:

```

variables = "absc"
zone i= 2390001
#      50.00000 12000.00000      0.00500
0.51219E-04
0.51323E-04
0.51428E-04
0.51534E-04
0.51642E-04
0.51750E-04
.
.
.

```

It is formatted for easy plotting using Tecplot, and has 2,390,001 absorption coefficient values between  $50\text{ cm}^{-1}$  and  $12000\text{ cm}^{-1}$ , spaced  $0.005\text{ cm}^{-1}$  apart.

The output file `kvsgh2o-0p-2000K.dat` has this form (with the columns for  $a3$  and  $a4$  omitted to fit on the page):

```

VARIABLES = k,g0,g1,g2,g3,g4,a1,a2,a3,a4
1.11334746D-07 7.96747373D-02 5.23869989D-04 2.26395097D-05 1.28391640D-03 2.13171234D-02 5.29826143D-05 2.63832240D-03 1.00906462D-01 4.55777755D-01
1.36266566D-07 8.52815350D-02 5.23869989D-04 2.26735350D-05 1.50824331D-03 2.36291115D-02 1.00848658D-04 4.08710454D-03 1.19960706D-01 4.79094048D-01
1.66115877D-07 9.00036377D-02 5.23869990D-04 2.27003552D-05 1.68726471D-03 2.55265665D-02 2.16767785D-04 7.38497566D-03 1.56250238D-01 5.22293603D-01
2.01726786D-07 9.37613925D-02 5.23869990D-04 2.27194809D-05 1.82135979D-03 2.69993935D-02 4.37361553D-04 1.32407710D-02 2.06442469D-01 5.79223754D-01
2.44067583D-07 1.02974254D-01 5.23934221D-04 8.52859416D-05 5.10437162D-03 3.41894314D-02 8.06580759D-04 2.24516206D-02 2.66272096D-01 6.42366201D-01
2.94246088D-07 1.12533644D-01 5.24022055D-04 1.63032520D-04 8.87822956D-03 4.20105578D-02 1.36144304D-03 3.55584825D-02 3.31229083D-01 7.04508996D-01
3.53526541D-07 1.23839293D-01 5.24099683D-04 2.30932151D-04 1.26253234D-02 5.07848713D-02 2.11775362D-03 5.25013589D-02 3.97103002D-01 7.60165878D-01
4.23348170D-07 1.41672092D-01 5.24204831D-04 3.15420522D-04 1.79307268D-02 6.44793111D-02 3.06402561D-03 7.24871772D-02 4.60296729D-01 8.06256909D-01
5.05345557D-07 1.63682856D-01 6.79267024D-04 4.18076075D-03 3.46648315D-02 8.58464331D-02 4.16735706D-03 9.41999954D-02 5.18053043D-01 8.41979615D-01
6.01370944D-07 1.82498123D-01 7.95460318D-04 6.98729223D-03 4.71695621D-02 1.03117054D-01 5.38432553D-03 1.16228009D-01 5.68674321D-01 8.68168095D-01
7.13518620D-07 1.99342127D-01 8.91344620D-04 9.21172573D-03 5.75711493D-02 1.18257717D-01 6.66353924D-03 1.37375476D-01 6.11507507D-01 8.86483146D-01
8.44151550D-07 2.14065460D-01 9.59688608D-04 1.08096384D-02 6.56348712D-02 1.30896677D-01 7.93423318D-03 1.56618822D-01 6.46421228D-01 9.98637601D-01
9.95930395D-07 2.28639506D-01 1.08555898D-03 1.32219150D-02 7.52270046D-02 1.44043846D-01 9.09287889D-03 1.72823102D-01 6.72970982D-01 9.05811610D-01
1.17184510D-06 2.43521364D-01 1.29626952D-03 1.68374407D-02 8.70200012D-02 1.58100939D-01 1.00091993D-02 1.84611821D-01 6.89951229D-01 9.08452972D-01
1.37524925D-06 2.57984408D-01 1.53627326D-03 2.08886000D-02 9.96127365D-02 1.72243921D-01 1.05612324D-02 1.90674644D-01 6.95893739D-01 9.06568830D-01
1.60989730D-06 2.71057492D-01 1.69859458D-03 2.36947584D-02 1.09121994D-01 1.84158478D-01 1.06844430D-02 1.90389205D-01 6.90296094D-01 9.0338299D-01
1.87998501D-06 2.83372640D-01 1.85241517D-03 2.63590759D-02 1.18246410D-01 1.95447797D-01 1.04050520D-02 1.84318068D-01 6.74688551D-01 8.90640135D-01
2.19019319D-06 2.94707018D-01 1.96152698D-03 2.82804768D-02 1.25440138D-01 2.05304290D-01 9.83672164D-03 1.74204744D-01 6.52715674D-01 8.79138308D-01
2.54573496D-06 3.04817787D-01 2.08359256D-03 2.96742175D-02 1.31095187D-01 2.13677119D-01 9.14489155D-03 1.62446894D-01 6.29124161D-01 8.67876894D-01
2.95240684D-06 3.14363969D-01 2.10218069D-03 3.08225009D-02 1.36202937D-01 2.21508242D-01 8.50562312D-03 1.51354772D-01 6.08270704D-01 8.58652439D-01
4.1664389D-06 3.24015430D-01 2.16685257D-03 3.20098365D-02 1.41584254D-01 2.29613988D-01 8.09474616D-03 1.42583906D-01 5.92969114D-01 8.52530310D-01
3.94557900D-06 3.32537431D-01 2.22666684D-03 3.31359103D-02 1.46613815D-01 2.36881980D-01 8.14093680D-03 1.36979489D-01 5.84160699D-01 8.49726486D-01
4.54710684D-06 3.41328215D-01 2.28089386D-03 3.41619508D-02 1.51377700D-01 2.44180400D-01 9.06733490D-03 1.34847804D-01 5.81367038D-01 8.49835656D-01
5.2295258D-06 3.49624273D-01 2.34055944D-03 3.52255005D-02 1.56195879D-01 2.51245912D-01 1.17210749D-02 1.36508281D-01 5.83534641D-01 8.52222591D-01
6.00374561D-06 3.58035652D-01 2.40321007D-03 3.63548190D-02 1.61112578D-01 2.58434724D-01 1.76275522D-02 1.42894888D-01 5.89833169D-01 8.56375019D-01
6.87908800D-06 3.66653849D-01 2.47755172D-03 3.76261434D-02 1.66416406D-01 2.65944430D-01 2.91050097D-02 1.55922207D-01 6.00116335D-01 8.62094495D-01
7.86769331D-06 3.74674767D-01 2.53082128D-03 3.85922339D-02 1.70965183D-01 2.72759681D-01 4.89979201D-02 1.78317154D-01 6.14936243D-01 8.69494644D-01
8.98236949D-06 3.82226610D-01 2.58843812D-03 3.95972865D-02 1.75407365D-01 2.79220074D-01 7.98603097D-02 2.12745245D-01 6.35149421D-01 8.78842614D-01
1.02372242D-05 3.89227742D-01 2.64363067D-03 4.06129729D-02 1.79956200D-01 2.85480820D-01 1.22726326D-01 2.60415217D-01 6.61281164D-01 8.90321322D-01
.
.
.
1.95195872D+03 9.99998942D-01 1.00000000D+00 1.00000000D+00 9.99999851D-01 9.99999312D-01 4.61364475D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
1.98997421D+03 9.99998962D-01 1.00000000D+00 1.00000000D+00 9.99999854D-01 9.99999326D-01 4.61364475D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01

```

```

2.02865475D+03 9.99998983D-01 1.00000000D+00 1.00000000D+00 9.99998857D-01 9.99999339D-01 4.61364475D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
2.06801066D+03 9.99999004D-01 1.00000000D+00 1.00000000D+00 9.9999860D-01 9.99999353D-01 4.61364475D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
2.10805239D+03 9.99999026D-01 1.00000000D+00 1.00000000D+00 9.9999863D-01 9.99999367D-01 4.61364474D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
2.14879055D+03 9.99999048D-01 1.00000000D+00 1.00000000D+00 9.9999866D-01 9.99999381D-01 4.61364474D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
2.19023589D+03 9.99999070D-01 1.00000000D+00 1.00000000D+00 9.9999869D-01 9.99999396D-01 4.61364473D-09 1.77221960D-04 1.40712612D-01 6.49679776D-01
2.23239930D+03 9.99999093D-01 1.00000000D+00 1.00000000D+00 9.9999872D-01 9.99999411D-01 4.61364469D-09 1.77221959D-04 1.40712612D-01 6.49679776D-01
2.27529180D+03 9.99999116D-01 1.00000000D+00 1.00000000D+00 9.9999876D-01 9.99999426D-01 4.61364458D-09 1.77221957D-04 1.40712612D-01 6.49679775D-01
2.31892460D+03 9.99999140D-01 1.00000000D+00 1.00000000D+00 9.9999879D-01 9.99999441D-01 4.61364431D-09 1.77221951D-04 1.40712610D-01 6.49679773D-01
2.36330903D+03 9.99999164D-01 1.00000000D+00 1.00000000D+00 9.9999882D-01 9.99999457D-01 4.61364375D-09 1.77221940D-04 1.40712607D-01 6.49679768D-01
2.40845656D+03 9.99999188D-01 1.00000000D+00 1.00000000D+00 9.9999886D-01 9.99999472D-01 4.61364270D-09 1.77221919D-04 1.40712602D-01 6.49679760D-01
2.45437885D+03 9.99999213D-01 1.00000000D+00 1.00000000D+00 9.9999889D-01 9.99999489D-01 4.61364091D-09 1.77221882D-04 1.40712592D-01 6.49679745D-01
2.50108769D+03 9.99999238D-01 1.00000000D+00 1.00000000D+00 9.9999893D-01 9.99999505D-01 4.61363821D-09 1.77221827D-04 1.40712578D-01 6.49679723D-01
2.54859502D+03 9.99999264D-01 1.00000000D+00 1.00000000D+00 9.9999896D-01 9.99999522D-01 4.61363456D-09 1.77221753D-04 1.40712558D-01 6.49679694D-01
2.59691297D+03 9.99999290D-01 1.00000000D+00 1.00000000D+00 9.9999900D-01 9.99999539D-01 4.61363017D-09 1.77221664D-04 1.40712535D-01 6.49679658D-01
2.64605380D+03 9.99999328D-01 1.00000000D+00 1.00000000D+00 9.9999905D-01 9.99999564D-01 4.61362544D-09 1.77221568D-04 1.40712509D-01 6.49679619D-01
2.69602995D+03 9.99999375D-01 1.00000000D+00 1.00000000D+00 9.9999912D-01 9.99999594D-01 4.61362089D-09 1.77221476D-04 1.40712485D-01 6.49679582D-01
2.74685401D+03 9.99999422D-01 1.00000000D+00 1.00000000D+00 9.9999919D-01 9.99999625D-01 4.61361698D-09 1.77221396D-04 1.40712464D-01 6.49679551D-01
2.79853875D+03 9.99999471D-01 1.00000000D+00 1.00000000D+00 9.9999926D-01 9.99999656D-01 4.61361398D-09 1.77221335D-04 1.40712448D-01 6.49679526D-01
2.85109710D+03 9.99999520D-01 1.00000000D+00 1.00000000D+00 9.9999932D-01 9.99999688D-01 4.61361191D-09 1.77221293D-04 1.40712437D-01 6.49679509D-01
2.90454215D+03 9.99999570D-01 1.00000000D+00 1.00000000D+00 9.9999939D-01 9.99999720D-01 4.61361065D-09 1.77221268D-04 1.40712430D-01 6.49679499D-01
2.95888719D+03 9.99999620D-01 1.00000000D+00 1.00000000D+00 9.9999947D-01 9.99999753D-01 4.61360995D-09 1.77221254D-04 1.40712427D-01 6.49679493D-01
3.01414566D+03 9.99999672D-01 1.00000000D+00 1.00000000D+00 9.9999954D-01 9.99999787D-01 4.61360961D-09 1.77221247D-04 1.40712425D-01 6.49679491D-01
3.07033118D+03 9.99999724D-01 1.00000000D+00 1.00000000D+00 9.9999961D-01 9.99999821D-01 4.61360946D-09 1.77221244D-04 1.40712424D-01 6.49679489D-01
3.12745754D+03 9.99999778D-01 1.00000000D+00 1.00000000D+00 9.9999969D-01 9.99999856D-01 4.61360940D-09 1.77221242D-04 1.40712424D-01 6.49679489D-01
3.18553874D+03 9.99999832D-01 1.00000000D+00 1.00000000D+00 9.9999976D-01 9.99999891D-01 4.61360938D-09 1.77221242D-04 1.40712423D-01 6.49679489D-01
3.24458892D+03 9.99999887D-01 1.00000000D+00 1.00000000D+00 9.9999984D-01 9.99999927D-01 4.61360937D-09 1.77221242D-04 1.40712423D-01 6.49679489D-01
3.30462243D+03 9.99999943D-01 1.00000000D+00 1.00000000D+00 9.9999992D-01 9.99999963D-01 4.61360937D-09 1.77221242D-04 1.40712423D-01 6.49679489D-01
3.36565380D+03 1.00000000D+00 1.00000000D+00 1.00000000D+00 1.00000000D+00 1.00000000D+00 4.61360937D-09 1.77221242D-04 1.40712423D-01 6.49679489D-01
3.39642078D+03 1.00000000D+00 1.00000000D+00 1.00000000D+00 1.00000000D+00 1.00000000D+00 4.61360937D-09 1.77221242D-04 1.40712423D-01 6.49679489D-01

```

Finally, output file `kvsqgh2o-0p-2000K.dat` contains quadrature  $k$ -values as:

```

wq      gq      kq      aq1      aq2      aq3      aq4
5.5566E-02 5.1201E-02 1.0056E-07 1.1349E-04 4.4089E-03 1.2272E-01 4.8243E-01
7.5768E-02 1.1707E-01 3.1802E-07 1.6206E-03 3.9770E-02 3.4270E-01 7.0910E-01
9.2583E-02 2.0159E-01 7.3344E-07 6.8338E-03 1.3182E-01 5.8699E-01 8.7111E-01
1.0483E-01 3.0071E-01 2.4012E-06 9.8831E-03 1.7233E-01 6.4875E-01 8.7864E-01
1.1185E-01 4.0950E-01 1.4519E-05 3.6367E-01 4.8589E-01 7.4587E-01 9.2075E-01
1.1326E-01 5.2253E-01 6.4322E-05 7.4837E-01 9.0208E-01 1.0097E+00 1.0374E+00
1.0900E-01 6.3413E-01 2.1014E-04 9.5075E-01 1.1213E+00 1.1166E+00 1.0833E+00
9.9278E-02 7.3871E-01 6.6275E-04 9.6959E-01 1.2567E+00 1.2799E+00 1.1654E+00
8.4579E-02 8.3102E-01 1.9721E-03 1.3306E+00 1.6521E+00 1.5328E+00 1.2665E+00
6.5640E-02 9.0645E-01 6.1860E-03 3.0111E+00 2.5667E+00 1.8180E+00 1.3507E+00
4.3413E-02 9.6121E-01 2.4663E-02 4.6993E+00 3.4362E+00 2.0650E+00 1.4178E+00
1.9048E-02 9.9256E-01 1.5850E-01 1.4341E+01 6.5090E+00 2.6153E+00 1.5531E+00

```

Note that the code has an accuracy-checking mechanism built in: if the absorption coefficient is calculated from the HITRAN/HITEMP databases, the Planck-mean absorption coefficient is calculated directly from the database's line intensities, as well as by line-by-line integration of the absorption coefficient; if the discrepancy exceeds 0.5% a message is printed to the screen, warning that `wvns` is too coarse to properly resolve the absorption coefficient. The Planck-mean absorption coefficient is also calculated from the  $k$ - $g$ -distribution. Again, if the discrepancy exceeds 0.5% a message is printed to the screen, warning that  $k$ -bin spacing is too coarse (`n_pwrk` too small) to properly resolve the absorption coefficient. For the above example, the choice of `n_pwrk`=500 results in an error of 1.78%, as indicated by the warning message.

### **fskdco2.f90, fskdh2o.f90**

These subroutines determine single values of the cumulative  $k$ -distribution for CO<sub>2</sub> and H<sub>2</sub>O, respectively, using the correlations of of Modest and Mehta [1] and of Modest and Singh [2].

Input for `fskdco2.f90`:

Tg = Gas temperature, i.e., temperature at which the absorption coefficient is evaluated, (in K)  
Tp = Planck function temperature, i.e., temperature at which  $I_b$  is evaluated, (in K)  
absco = Pressure-based absorption coefficient, (in cm<sup>-1</sup> bar<sup>-1</sup>)

Input for `fskdco2.f90`: same as for `fskdco2.f90` plus

x = Mole fraction of water vapor, (-)

Output for both:

gcal = Cumulative  $k$ -distribution for the input conditions, (-).

### **fskdco2dw.f90, fskdh2odw.f90**

These subroutines determine single values of the cumulative  $k$ -distribution for CO<sub>2</sub> and H<sub>2</sub>O, respectively, using the correlations of Denison and Webb [3,4].

Input for `fskdco2dw.f90`:

Tg = Gas temperature, i.e., temperature at which the absorption coefficient is evaluated, (in K)

$T_p$  = Planck function temperature, i.e., temperature at which  $I_b$  is evaluated, (in K)

$C_{abs}$  = Molar absorption cross-section,  $R_u T_g k / x p$ , (in  $m^2/mol$ )

Input for `fskdh2odw.f90`: same as for `fskdco2.f90` plus

$x$  = Mole fraction of water vapor, (-)

Output for both:

$g_{cal}$  = Cumulative  $k$ -distribution for the input conditions, (-).

#### **kdistmix.f90:**

Subroutine `kdistmix` finds the cumulative  $k$ -distribution for an  $n$ -component mixture from a given set of individual species cumulative  $k$ -distributions (narrow band, wide band, or full spectrum), employing mixing schemes. Three mixing scheme are implemented, namely superposition, multiplication and uncorrelated mixture (Modest and Riazzi [5]). The mixing model is implemented as an independent module. For  $n > 2$  `kdistmix.f90` should be called recursively. To invoke `kdistmix`, the user should call

```
use modkdistmix, only : kdistmix
call kdistmix(k1, g1, k2, g2, k, g, mixmodel, mixNop, mixScheme)
```

Input for subroutine `kdistmix`:

$k_1$  = A double precision array with  $k$ -values for the  $k$ -distributions of the first species, (in  $cm^{-1}$ )

$g_1$  = A double precision array with  $g$ -values for the  $k$ -distributions corresponding to the  $k$ -values in array  $k_1$ , (-). The size of  $g_1$  must be the same as  $k_1$ .

$k_2$  = A double precision array with  $k$ -values for the  $k$ -distributions of the second species, (in  $cm^{-1}$ ). The size of  $k_2$  may be different from  $k_1$ .

$g_2$  = A double precision array with  $g$ -values for the  $k$ -distributions corresponding to the  $k$ -values in array  $k_2$ , (-). The size of  $g_2$  must be the same as  $k_2$ , but may be different from  $g_1$ .

$k$  = A double precision array with  $k$ -values for the  $k$ -distributions of the mixture, (in  $cm^{-1}$ ). The size of  $k$  may be different from  $k_1$  and/or  $k_2$ .

`mixmodel` = An optional integer scalar to specify the mixing model. Valid model numbers are 1 for superposition, 2 for multiplication and 3 for uncorrelated mixture (Modest and Riazzi). If not given, the uncorrelated mixture model will be used.

`mixNop` = An optional integer scalar to specify the minimum number of points for internal calculations. If not given, a value of 256 will be used. This number is only needed for the uncorrelated mixture model.

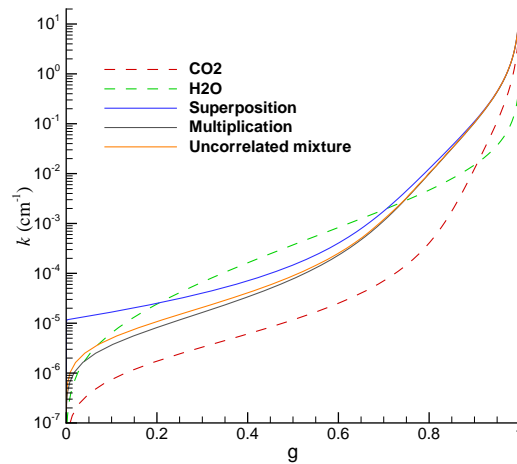
`mixScheme` = An optional integer scalar to specify the integration scheme for the uncorrelated mixture model and is only used for this model. If not given, a value of 0 for the default integration scheme will be used. Currently only the integration model is implemented. This number is reserved for future development.

Output for subroutine `kdistmix`:

$g$  = A double precision array of rank one with  $g$ -values for the mixed  $k$ -distribution corresponding to the  $k$ -values, (-).

#### **Example:**

Consider a mixture of  $CO_2$  and  $H_2O$  with mole fraction of 0.2 and 0.2, respectively. The mixture has a total pressure of 1 bar and temperature of 800K. The Planck function temperature is 1000K. The full-spectrum  $k$ -distribution data are determined from correlation tables. The following program finds the full-spectrum  $k$ -distributions of the mixture using three different mixing models (superposition, multiplication, and uncorrelated). The results are compared in the figure below:



This example also contains a function `kPowerLaw` to generate a list of  $k$ -values from a power law between minimum and maximum values.

Input for function `kPowerLaw`:

`kmin` = Minimum value (in  $\text{cm}^{-1}$ ).  
`kmax` = Maximum value (in  $\text{cm}^{-1}$ ).  
`n` = Number of  $k$ -values desired.  
`pwr` = Exponent for  $k$ -value spacing (see also `fskdist.f90`).

Output for function `kPowerLaw`:

`k` = An array of rank one and size `n` that contains a list of  $k$ -values (in  $\text{cm}^{-1}$ ). A sequence  $k_i$  for  $i = 1, \dots, n$  from a power law of power  $p$  (`pwr`) with a minimum  $k_{\min}$  and a maximum  $k_{\max}$  has  $k_i^p$  equally distributed values between  $k_{\min}^p$  and  $k_{\max}^p$ . A value of 0.1 for the power  $p$  is suggested.

```

program mixTest
use modkdistmix, only : kdistmix
implicit none

! export nb db
real(8),parameter :: P=1.d0,T=800.d0,xCO2=0.2d0,xH2O=0.2d0
real(8),parameter :: Trad=1000.d0
integer :: erflag=0, ik, ib
real(8) :: x, bb1,bb2
integer, parameter :: nop = 128, m=3
real(8),dimension(nop) :: k, gCO2, gH2O, gSup, gMul,gMR
real(8):: kp
real(8), parameter :: kmin=1.d-9, kmax=1.d2
k = kPowerLaw(kmin, kmax, nop, 0.1d0)
gCO2=1.d0;gH2O=1.d0
do ik=1, nop
    kp=k(ik)/xCO2
    call fskdco2(T, Trad, kp, gCO2(ik))
    ! k in correlation is pressure based
end do
do ik=1, nop
    kp=k(ik)/xH2O
    call fskdh2o(T, Trad, kp, xH2O, gH2O(ik))
    ! k in correlation is pressure based
enddo

call kdistmix(k, gCO2, k, gCO2, k, gSup, 1)
call kdistmix(k, gCO2, k, gCO2, k, gMul, 2)
call kdistmix(k, gCO2, k, gCO2, k, gMR, 3)
open(60, file='fskgMix.dat')
do ib = 1, nop
write(60,'(6f12.8)') k(ib),gCO2(ib), gH2O(ib), gSup(ib), gMul(ib), gMR(ib)
enddo
close (60)

contains
function kPowerLaw (kmin,kmax, n, pwr) result(k)
! function generate a list of k-values between kmin and kmax
! according to power law with power "pwr"

```

```

integer, parameter :: dp = kind(1.d0)
real(dp), intent(in) :: kmin, kmax, pwr
integer, intent(in) :: n
real(dp), dimension(n) :: k
real(dp) :: pwrk_min, pwrk_max, pwrk_step
integer :: i
pwrk_min = kmin*pwr
pwrk_max = kmax*pwr
pwrk_step = (pwrk_max-pwrk_min)/real(n-1, dp)
k = (/ (pwrk_min+real(i-1,dp)*pwrk_step, i=1, n) /)
k = k**(1.d0/pwr)
end function kPowerLaw
end program

```

### fskdistmix.f90:

This self-contained Fortran module finds the full spectrum cumulative  $k$ -distribution for a CO<sub>2</sub>-H<sub>2</sub>O mixture, employing the correlations of Modest and Mehta [1] and Modest and Singh [2], using one of three mixing schemes described by equations (20.162) (superposition), (20.163) (multiplication), or (20.167) (uncorrelated mixture).

To invoke the model, the user calls

```

use modfskdistmix, only : fskdistmix
call fskdistmix(xCO2, xH2O, Tg, Tp, kq, gq, m, nop, errflag)

```

Input for subroutine fskdistmix:

xCO2 = CO<sub>2</sub> mole fraction, (-).  
xH2O = H<sub>2</sub>O mole fraction, (-).  
Tg = Gas temperature (in K).  
Tp = Planck function temperature (in K).  
m = Integer to specify mixing model. m = 1 for superposition, 2 for multiplication and 3 for uncorrelated mixture (Modest and Riazzi)  
nop = Integer to specify number of points for internal calculation.  
gq = A double precision array for  $g$ -values (quadrature points).

Output for subroutine fskdistmix:

kq = A double precision array of  $k$ -values for the quadrature points specified by gq.  $kq$  is linear based and has the same size as gq.  
errflag = Error flag. errflag = 0 if no error, errflag = 1 if error is found, such as a wrong model number.

This module also provides a subroutine for quadrature point calculation, generating Gaussian or Chebychev quadrature self-contained Fortran module onto between 0 and 1 and open at both ends. The corresponding quadrature weights are also calculated.

To invoke the quadrature subroutine, the user calls

```

use modfskdistmix, only : quadgen2
call quadgen2(Cheb, g, w, nq)

```

Input for subroutine quadgen2:

Cheb = A logical scalar to switch between Gaussian and Chebychev quadrature schemes. Should be set to True for Chebychev quadrature, False for Gaussian quadrature.  
nop = An integer scalar specifying the number of quadrature points.

Output for subroutine quadgen2:

g = An array of size nop containing quadrature points.  
w = An array of size nop containing quadrature weights.

### Example:

In this example we consider a gas mixture with a total pressure of 1 bar, temperature of 800K. It contains 20% of CO<sub>2</sub> and 20% of H<sub>2</sub>O by mole. The following program finds the full-spectrum  $k$ -distribution of a this mixture subject to 1000K Planck function temperature, using correlation tables and compares results between different mixing models.

```

program mixTest
use modfskdistmix, only : fskdistmix, quadgen2
implicit none
real(8), parameter :: P=1.d0, T=800.d0, xCO2=0.2d0, xH2O=0.2d0
real(8), parameter :: Trad=1000.d0
integer :: erflag=0, ib
integer, parameter :: nq = 16, nopcorr = 1024
real(8), dimension(nq) :: gq, wq
real(8), dimension(nq) :: kqSup, kqMul, kqMR
call quadgen2(.false., gq, wq, nq)
call fskdistmix(xCO2, xH2O, T, Trad, kqSup, gq, 1, nopcorr, erflag)
call fskdistmix(xCO2, xH2O, T, Trad, kqMul, gq, 2, nopcorr, erflag)
call fskdistmix(xCO2, xH2O, T, Trad, kqMR, gq, 3, nopcorr, erflag)

```

```

open(60, file='fsgkCorr.dat')
do ib = 1, nq
write(60,'(5f12.5)') gq(ib),wq(ib), kqSup(ib), kqMul(ib), kqMR(ib)
enddo
close (60)
end program

```

The output quadrature  $g$  points, quadrature weights  $w$ , and  $k$ -values from three mixing models are listed below:

gq	wq	k sup	k mul	k MR
0.07051694	0.13911035	0.00010040	0.00001174	0.00001545
0.20568663	0.13129793	0.00021668	0.000006478	0.00007703
0.33189367	0.12078145	0.00046038	0.00021092	0.00023517
0.44797743	0.11146496	0.00095065	0.00056075	0.00058797
0.55303706	0.09846713	0.00190269	0.00132135	0.00144293
0.64644658	0.08844043	0.00371492	0.00288260	0.00292762
0.72786419	0.07436563	0.00716137	0.00600981	0.00634272
0.79723565	0.06447459	0.01384539	0.01230641	0.01296693
0.85479181	0.05076401	0.02734917	0.02538019	0.02569302
0.90104015	0.04183752	0.05637761	0.05402394	0.05620029
0.93675064	0.02984988	0.12386378	0.12137227	0.12518867
0.96293624	0.02263203	0.29454263	0.29238274	0.30489881
0.98082827	0.01353204	0.76224969	0.76083830	0.76602318
0.99184741	0.00862024	2.17053771	2.16991559	2.16884325
0.99757068	0.00328520	7.21939997	7.21923941	7.21288088
0.99969530	0.00107660	35.31945789	35.31944491	35.28831336

The results are identical to the ones given in the previous example for `kdistrib.f90`, since  $k$ -distributions in that figure were obtained from the correlations.

## References

1. Modest, M. F., and R. S. Mehta: "Full spectrum  $k$ -distribution correlations for CO<sub>2</sub> from the CDSD-1000 spectroscopic databank," *International Journal of Heat and Mass Transfer*, vol. 47, pp. 2487–2491, 2004.
2. Modest, M. F., and V. Singh: "Engineering correlations for full spectrum  $k$ -distribution of H<sub>2</sub>O from the HITEMP spectroscopic databank," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 93, pp. 263–271, 2005.
3. Denison, M. K., and B. W. Webb: "An absorption-line blackbody distribution function for efficient calculation of total gas radiative transfer," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 50, pp. 499–510, 1993.
4. Denison, M. K., and B. W. Webb: "Development and application of an absorption line blackbody distribution function for CO<sub>2</sub>," *International Journal of Heat and Mass Transfer*, vol. 38, pp. 1813–1821, 1995.
5. Modest, M. F., and R. J. Riazzi: "Assembly of full-spectrum  $k$ -distributions from a narrow-band database; effects of mixing gases, gases and nongray absorbing particles, and mixtures with nongray scatterers in nongray enclosures," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 90, no. 2, pp. 169–189, 2005.