

■ [Namelist integ](#)

- [jsflx](#)
- [jshrt](#)
- [nrdc](#)
- [nxr, nyr](#)
- [therdc, phirdc](#)
- [jzrdc](#)
- [ncam](#)
- [nxc, nyc](#)
- [jzcam](#)
- [xcam, ycam](#)
- [rmincam](#)
- [rmidcam](#)
- [rmaxcam](#)
- [thecam, phicam](#)
- [ahcam](#)
- [avcam](#)
- [amcam](#)
- [diacam](#)
- [npot](#)
- [nppot](#)
- [nqpot](#)
- [jspot](#)
- [jzpot](#)
- [xpot, ypot](#)
- [rminpot](#)
- [rmidpot](#)
- [rmaxpot](#)
- [thepot, phipot](#)
- [pminpot, pmaxpot](#)
- [qmaxpot](#)
- [diapot](#)

■ [Namelist model](#)

- [nx, ny, nz](#)
- [iz3l](#)
- [nz3](#)
- [xbin, ybin](#)
- [jrfr](#)
- [jsph](#)
- [rplanet](#)
- [jtoplev](#)
- [zgrd](#)
- [drfr](#)
- [dens](#)
- [tmpa1d](#)
- [ext1st, omg1st](#)
- [jpf1st](#)
- [ext2nd, omg2nd, jpf2nd](#)
- [ext3rd, omg3rd, jpf3rd](#)
- [nkd](#)
- [wkd](#)
- [absg1d](#)

- [Namelist mdlphs](#)
  - [filephs](#)
  - [npf](#)
  - [nang](#)
  - [ang](#)
  - [phs](#)
- [Namelist mdlisfc](#)
  - [filesfc](#)
  - [nxb, nyb](#)
  - [tmps2d](#)
  - [jsfc2d](#)
  - [psfc2d](#)
- [Namelist mdla3d](#)
  - [ifmt3d](#)
  - [npar](#)
  - [filea3d](#)
  - [tmpa3d](#)
  - [absg3d](#)
  - [extp3d, omgp3d](#)
  - [jfp3d](#)
- [External input file filephs](#)
- [External input file filesfc](#)
- [External input file filea3d](#)
  - [Sequential access format](#)
  - [Direct access format](#)

[Edit](#)

## Namelist integ

### jsflx

[integer scalar]

Switch for calculating fluxes (area-averaged fluxes). They are computed as follows:

0: no flux output  
 1: at top and bottom of the atmosphere (TOA and BOA, respectively)  
 2: at interfaces of all layers except internal interfaces of 3-D inhomogeneous layers  
 3: at interfaces of all layers

The number of interfaces at which fluxes are computed will be

0 for jsflx=0,  
 2 for jsflx=1,  
 $nz - nz3 + 2$  for jsflx=2,  
 $nz + 1$  for jsflx=3.

See also nz and nz3.

### jsprt

[integer scalar]

Switch for calculating heating rates. If jsprt=1, then heating rates are computed for all nz layers.

### nrdc

[integer scalar]

Number of area-averaged radiances, one of which is specified by altitude and direction.

### nxr, nyr

[integer scalar]

Number of pixels along x and y axes, respectively, for pixel-averaged radiances.

### therdc, phirdc

[real 1-D array, {dat(irdc), irdc=1, nrdc}]

Radiance zenith and azimuth angles, respectively, in degree. The angles corresponds to the light transport vectors.

### **jzrdc**

[integer 1-D array, {dat(irdc), irdc=1, nrdc}]

Layer interface number that specifies the altitude at which the area-averaged radiances are sampled. Should be larger than or equal to 0 and smaller than or equal to  $nz$ . The interface  $jzrdc=0$  corresponds to the bottom of atmosphere (BOA). If  $jzrdc > nz$ , then  $jzrdc$  will be reset to  $nz$ , by the code. The horizontal location for radiance sampling is defined at the sampling level. Note that this differs from the method that is used for map projection of satellite observation image pixels, which are mapped at the ground level (not at the satellite level).

### **ncam**

[integer scalar]

Number of cameras. One camera image is composed of angular-averaged radiances viewing from a point.

### **nxc, nyc**

[integer scalar]

Numbers of pixels along horizontal and vertical directions, respectively, of camera images.

### **jzcam**

[integer 1-D array, {dat(icam), icam=1, ncam}]

Layer interface number at which each camera is located. Should be between 0 and  $nz$ .

### **xcam, ycam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Relative horizontal location (0 to 1) along x and y-axes, respectively, for the camera. The x-coordinate between 0 and  $xmax$  corresponds to  $xcam = 0$  to 1. It is also similar for y-coordinate.

### **rmincam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Minimum distance (m) of the view region of the camera. Light sources from the domain closer to the camera is forced to set as it is from the point of the limited distance,  $rmincam$ . For simulation of physically-correct images, this limit should be set as 0. However, simulated images will be very noisy when scattering medium is present near the camera. Such a situation frequently appears when Rayleigh scattering is taken into account. For that reason, light from source emission or scattering event near the camera should be adjusted for obtaining smooth, better-looking camera image.

### **rmidcam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Moderate distance (m) for the camera radiance.

### **rmaxcam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Maximum distance (m) of the view region of camera. This limits distance between emission/scattering point and the camera. For simulation of physically-correct images, this limit should be infinity. However, such a simulation is not feasible. See also  $rmincam$ .

### **thecam, phicam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Zenith and azimuth angles, respectively, in degree, of direction of the center of camera image. The angles correspond to the camera-to-target vector.

### **ahcam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Maximum view angle (degrees) along camera image's horizontal (scan) direction. The horizontal direction is perpendicular to the vertical direction at the image center.

### **avcam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Maximum view angle (degrees) along camera image's vertical (track) direction. At the image center, a plane along the vertical direction of the image includes the zenith and nadir direction in the spatial domain (the Cartesian coordinate system).

### **amcam**

[real 1-D array, {dat(icam), icam=1, ncam}]

Maximum angle (degrees) of the field of view of the camera, from the center direction. Should be less than 90 degrees.

## diacam

[real 1-D array, {dat(icam), icam=1, ncam}]

Diameter (in meter) of the camera lens surface. The surface is assumed as spherical. If diacam=0, then the coordinates of the camera focus point should be (xcam, ycam). Otherwise (diacam >0), the point is at

$$x = xcam - D/2 * \sin(\text{thecam}) * \cos(\text{phicam}), \text{ and}$$
$$y = ycam - D/2 * \sin(\text{thecam}) * \sin(\text{phicam}),$$

where D is diameter of the lens surface sphere:

$$D = \text{diacam} / \sin(\text{amcam}).$$

See also thecam, phicam, and amcam.

## npot

[integer scalar]

Number of detectors for the local fluxes.

## nppot

[integer scalar]

Number of pathlength bins for the local fluxes. See also pminpot and pmaxpot.

## nqpot

[integer scalar]

Number of view angle bins for the local fluxes. See also qmaxpot.

## jspot

[integer 1-D array, {dat(ipot), ipot=1, npot}]

Switch for integration of plane flux (jspot = 0) or spherical flux (jspot = 1).

## jzpot

[integer 1-D array, {dat(ipot), ipot=1, npot}]

Index of layer boundary at which the local flux is sampled. Should be between 0 and nz.

## xpot, ypot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Relative horizontal location (0 to 1) along x and y-axes, respectively, for the local flux. The x-coordinate between 0 to xmax corresponds to xpot = 0 to 1. It is also similar for y-coordinate.

## rminpot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Minimum distance (m) of the view domain of the local flux. Emission/scattering from regions closer to the detector is forced to set as it is from the point of the limit distance, rminpot. For simulation of physically-correct results, this limit should be 0. However, simulated results will be very noisy if scattering medium is present near the detector. Such a situation frequently appears when Rayleigh scattering is taken into account. For this reason, light sources from emission/scattering events that occurs near the detector should be adjusted for obtaining better-looking results.

## rmidpot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Moderate distance (m) for the local flux.

## rmaxpot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Maximum distance (m) of the view domain of the local flux. This limits distance between scattering point and the detector. For simulation of physically-correct results, this limit should not be set as infinity. However, such a simulation is highly time-consuming. See also rminpot.

## thepot, phipot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Zenith and azimuth angles, respectively, in degree, of direction of the center of the local flux detector. The angles correspond to detector-to-target vector.

## pminpot, pmaxpot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Minimum and maximum limits of pathlength for sampled local flux, where the pathlength is integrated flight length of photon between the source emission point and the detector sampling point. The pathlength range is divided into bins, number of which is

nppot.

### qmaxpot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Maximum angle (degrees) of the field of view of the local flux detector, from the center direction. The field of view is conical angular region. Note, however, that qmaxpot can be larger than 90 degrees. When qmaxpot=180 degrees, for example, then the field of view is the globe. The angular field is divided into bins, number of which is npot.

### diapot

[real 1-D array, {dat(ipot), ipot=1, npot}]

Diameter (in meter) of the lens surface of the local flux detector. The detector surface is assumed as spherical. If diapot=0, then the coordinates of the detector focus point should be (xpot, ypot). Otherwise (diapot > 0), the point is at

$$\begin{aligned}x &= xpot - D/2 * \sin(\text{thepot}) * \cos(\text{phipot}), \text{ and} \\y &= ypot - D/2 * \sin(\text{thepot}) * \sin(\text{phipot}),\end{aligned}$$

where D is diameter of the lens surface sphere:

$$D = \text{diapot} / \sin(\text{qmaxpot}).$$

See also thepot, phipot, and qmaxpot.

[Edit](#)

## Namelist model

### nx, ny, nz

[integer scalar]

Numbers of grids along x, y, and z-axis, respectively, in the spatial domain.

### iz3l

[integer scalar]

Lowest index of z-coordinate grid (or layer), of horizontally inhomogeneous layers. Layers for the z-index iz=iz3l to iz3l + nz3 - 1 are defined as horizontally inhomogeneous, and for these layers 3-D data of optical properties are given. See also nz3.

### nz3

[integer scalar]

Number of horizontally inhomogeneous layers, which is defined for the z-grid index

$$iz=iz3l \text{ to } iz3l + nz3 - 1.$$

See also iz3l.

### xbin, ybin

[real scalar]

Grid spacing along x and y-axis, respectively.

### jfr

[integer scalar]

Switch for atmospheric refraction. If jfr = 1, then refraction is simulated at each atmospheric layer boundary. Reflection can also occur when the refraction is taken into account. Otherwise (jfr = 0), the refraction/reflection is neglected.

### jsph

[integer scalar]

Switch for correction of atmospheric sphericity effects on radiance and air mass factors. Should be 0 or 1. The correction is implemented only for the line of sight (the path between the last scattering point and the radiance detector). This may influence on results of radiance of direction near the horizon (within 3 degrees) under some special conditions (e.g., very optically thin atmosphere).

### rplanet

[real scalar]

Radius (in meter) of the planet, for the sphericity correction on the radiances and air mass factors. Used only when jsph=1. The planet surface should correspond to z = 0. For the earth, the radius is 6.74e+6 m approximately.

### jtplev

[integer scalar]

Defines a method to specify temperature data, as follows:

0: temperature data are given for layers (grids), and the temperature is assumed as constant in each layer.

1: temperature data are given for layer interfaces, and the temperature is

assumed to vary linearly along z-coordinate in each layer. Required number of z-axis levels for temperature data is  $nz$  for  $jtmplev=0$  and  $nz+1$  for  $jtmplev=1$ . Temperature data are used only when the source includes thermal emission ( $jdefsrc = 2$  or  $3$ ).

### **zgrd**

[real 1-D array, {dat(iz), iz=0, nz}]

Z-coordinate values (in meter) of layer boundaries. The top and bottom of a layer with index  $iz$  corresponds to  $z=zgrd(iz)$  and  $zgrd(iz+1)$ , respectively. The BOA (surface) corresponds to  $zgrd(0)$ , and TOA  $zgrd(nz)$ .

### **drfr**

[real 1-D array, {dat(iz), iz=1, nz}]

Anomalies in atmospheric refractive index from unity, for each layer. Used only when  $jfr=1$ . Give null if the index is unity. Actual values of refractive indexes should be  $1+drfr$ .

### **dens**

[real 1-D array, {dat(iz), iz=1, nz}]

Gas density (in relative unit) for air mass factor calculations, used only when calculating area-averaged radiance. This is used as a weighting function for average air mass factors of radiance.

### **tmpa1d**

[real 1-D array, {dat(iz), iz=1, nz}] when  $jtmplev=0$

[real 1-D array, {dat(iz), iz=0, nz}] when  $jtmplev=1$

Horizontally-averaged atmospheric temperature [K] defined at layers if  $jtmplev=0$ , or at layer interfaces if  $jtmplev=1$ . Used for thermal emission when  $jdefsrc = 2$  or  $3$ . When  $jtmplev=1$ , an interface of  $iz=0$  corresponds to BOA, and that of  $iz=nz$  corresponds to TOA. Note that  $tmpa1d$  is atmospheric temperature even when  $iz=0$  (BOA). The surface temperature should be given in `namelist mdlscf`. For horizontally inhomogeneous layer, data of perturbation in temperature will be specified in `namelist mdla3d`.

### **ext1st, omg1st**

[real 1-D array, {dat(iz), iz=1, nz}]

Extinction coefficients (/m) and single scattering albedo, respectively, of the first kind optical medium, which can be gas, aerosol, hydrometeor, or their mixture. Note that all optical properties are assumed as constant in every cells.

### **jpf1st**

[integer 1-D array, {dat(iz), iz=1, nz}]

indexes of phase functions of the first kind scattering medium. Should be between 0 and  $npf$ . If  $jpf1st = 0$ , then predefined Rayleigh phase function is used.

### **ext2nd, omg2nd, jpf2nd**

[real 1-D array, {dat(iz), iz=1, nz}]

As `ext1st`, `omg1st`, and `jpf1st`, respectively, but for the second kind optical medium.

### **ext3rd, omg3rd, jpf3rd**

[real 1-D array, {dat(iz), iz=1, nz}]

As `ext1st`, `omg1st`, and `jpf1st`, respectively, but for the third kind optical medium.

### **nkd**

[integer scalar]

Number of terms for the k-distribution used as the gaseous absorption model. Give  $nkd=1$  for monochromatic computation.

### **wkd**

[real 1-D array, {dat(ikd), ikd=1, nkd}]

Weight coefficients of the correlated k-distribution. Give  $wkd(1)=1$  for monochromatic computation.

### **absg1d**

[real 2-D array, {(dat(ikd, iz), ikd=1, nkd), iz=1, nz}]

Horizontally-averaged gaseous absorption coefficients (/m) of the correlated k-distribution. Perturbations of the absorption coefficient in the horizontally inhomogeneous layers can be given in `namelist mdla3d`.

[Edit](#)

## **Namelist mdlphs**

### **filephs**

[character string scalar]

Filename for external file input of phase function data. Leave it as default '' if no external input file is needed. The name should have relative path from a directory where the configuration file exists. The external file input will override on input from the following

arguments of namelist mdlphs.

## npf

[integer scalar]

Number of phase functions

## nang

[integer 1-D array, {dat(ipf), ipf=1, npf}]

Numbers of angle grids for the phase function. The followings are exceptions:

- If nang=1, then Heyney-Greenstein phase function with asymmetry parameter value of phs(iang, ipf) for iang=1.
- If nang=0, then Rayleigh scattering phase function is used.

## ang

[integer 2-D array, {{dat(iang, ipf), iang=1, nang), ipf=1, npf}]

Angles in degree for the phase function.

## phs

[integer 2-D array, {{dat(iang, ipf), iang=1, nang), ipf=1, npf}]

Phase functions. Relative values are acceptable, because the phase functions are automatically normalized in the code. See also nang.

[Edit](#)

## Namelist mdlsrc

### filesfc

[character string scalar]

Filename for external file input of surface model data. Leave it as default '' if no external input file is needed. The name should have relative path from a directory where the configuration file exists. The external file input will override on input from the following arguments of namelist mdlsrc. The file can be the same as filephs.

### nxb, nyb

[integer, scalar]

Numbers of (X, Y) pixels for the surface properties. This is now independent of the atmosphere. For example, if nxb = 1 and nyb = 1, then surface properties are constant in the full domain.

### trmps2d

[real 2-D array, {{dat(ixb, iyb), ixb=1, nxb), iyb=1, nyb}]

The surface temperature (in K). Used for calculation of thermal emission when jdefsrc = 2 or 3.

### jsfc2d

[integer 2-D array, {{dat(ixb, iyb), ixb=1, nxb), iyb=1, nyb}]

Index for surface BRDF model of each pixel at (ixb, iyb):

- 0: black
- 1: Lambertian
- 2: diffuse-specular mixture (DSM) model
- 3: Rahman-Pinty-Verstraete (RPV) model
- 4: Li-Sparse-Ross-Thick (LSRT) model

This is neglected if jsfc2d is given in the file, filesfc.

### psfc2d

[real 2-D array, {{{dat(ixb, iyb, ip), ixb=1, nxb), iyb=1, nyb), ip=1, 5}]

Surface parameters of each pixel at (ixb, iyb). These are neglected if psfc2d is given in the file, filesfc. Definitions of psfc2d for respective ip differ by the BRDF model (jsfc2d), as follows:

- jsfc2d = 0 (black): No specification is required for psfc2d, which is dummy parameter.
- jsfc2d = 1 (Lambertian):
  - psfc2d(ip=1): albedo
- jsfc2d = 2 (DSM): The surface reflection is modeled by a microscopical mixture of diffusive and specular reflectors.
  - psfc2d(ip=1): diffuse albedo (a\_d)
  - psfc2d(ip=2): fraction of diffuse reflectors (f\_d); Fraction of specular reflection is 1 - psfc2d(ip=2)
  - psfc2d(ip=3): real part of refractive index of underlying material (e.g., water); Should be given as positive.
  - psfc2d(ip=4): imaginary part of refractive index of underlying material (e.g., water); Should be given as positive.

psfc2d(ip=5): variance of micro-scopic surface slope (tangent of the inclination angle). Give zero for flat surface. This is used

- for specular reflection. According to Cox and Munk's (1954) parameterization for ocean surface,

$$\text{psfc2d}(ip=5) = 0.00512 * V + 0.003.$$

where V is wind velocity (m/s) at 10 m above the ocean surface.

The DSM model albedo follows

$$a = f\_d * a\_d + (1 - f\_d) * a\_s,$$

where a\_s is albedo for specular reflection. BRDF of the DSM model follows the similar equation.

- jsfc2d = 3 (RPV): Rahman-Pinty-Verstraete (RPV) BRDF model
  - psfc2d(ip=1): the first parameter, alpha
  - psfc2d(ip=2): the exponent parameter, k
  - psfc2d(ip=3): the asymmetry parameter, THETA
  - psfc2d(ip=4): delta

The RPV BRDF, R, is given as

$$\pi * R = \alpha * [u_0 * u_1 / (u_0 - 1) / (u_1 - 1)]^{(k - 1)} * HG(\text{THETA}) * F(G)$$

where  $\pi=3.141592653$ ,  $u_0$  and  $u_1$  are respectively cosines of incidence and reflection vectors, and HG is the Heyney-Greenstein function. The function F is as follows:

$$F(G) = 1 + (1 + G) / (\text{delta} + G),$$

where G is a geometrical parameter.

- jsfc2d = 4 (LSRT): Li-Sparse-Ross-Thick (LSRT) BRDF model
  - psfc2d(ip=1): "k\_L", weight of Lambertian reflection
  - psfc2d(ip=2): "k\_g", weight of geometrical optics reflection
  - psfc2d(ip=3): "k\_v", weight of volume scattering

The LSRT BRDF, R, is given by

$$\pi * R = k\_L * \pi + k\_g * K\_geo + k\_v * K\_vol$$

where K\_geo and K\_vol are geometrical and volume scattering kernel functions.

[Edit](#)

## Namelist mdla3d

### ifmt3d

[integer scalar]

Switch for format of the external input file, filea3d:

- 0: Sequential-access text format,
- 1: Direct-access binary format.

### npar

[integer scalar]

Number of scattering media in horizontally inhomogeneous layers. A medium can be a mixture of gases, aerosols and hydrometeors or a size-bin of their size distributions.

### filea3d

[character string scalar]

Filename of an external file of 3-D atmospheric model data. Leave it as default '' if no external input file is needed. The name should have relative file name path from the directory where the configuration file exists. The external file input will override on input from the following arguments of namelist mdla3d (except ifmt3d and npar). The file can be the same as filephs and/or filesfc when sequential access format (ifmt3d=0).

### tmpa3d

[real 3-D array, {{{(dat(ix, iy, iz3), ix=1, nx), iy=1, ny), iz3=1, nz3}}] when jtmplev=0

[real 3-D array, {{{(dat(ix, iy, iz3), ix=1, nx), iy=1, ny), iz3=0, nz3}}] when jtmplev=1

3-D distributions of perturbations of atmospheric temperature [K] at layers (jtmplev=0) or layer interfaces (jtmplev=1) of horizontally inhomogeneous layers. Used for calculation of thermal emission when jdefsrc = 2 or 3. When jtmplev=0, an interface of index iz3=0 corresponds to the bottom of the lowest inhomogeneous layer, and iz3=nz3 corresponds to the top of the highest inhomogeneous layer. Actual temperatures are represented as follows,

```
temperature(ix, iy, iz) = tmpa1d(iz) + tmpa3d(ix, iy, iz3), where iz = iz3l + iz3 - 1.
```

See also tmpa1d and iz3l.

### absg3d

[real 4-D array, {{{(dat(ix, iy, iz3, ikd), ix=1, nx), iy=1, ny), iz3=1, nz3), ikd=1, nkd}}]

3-D distributions of perturbations of gaseous absorption coefficient of the correlated k-distribution. Actual absorption coefficients are represented as follows,

```
coefficient(ix, iy, iz, ikd) = absg1d(iz, ikd) + absg3d(ix, iy, iz3, ikd),
```

where  $iz = iz3l + iz3 - 1$ . Note that the total coefficient (absg1d + absg3d) may not be less than zero.

### extp3d, omgp3d

[real 4-D array, {{{(dat(ix, iy, iz3, ipar), ix=1, nx), iy=1, ny), iz3=1, nz3), ipar=1, npar}}]

Extinction coefficients and single scattering albedos, respectively, of media in horizontally inhomogeneous layers with  $iz3=1$  to  $nz3$ . A medium can be a mixture of gases, aerosols and hydrometeors or a size-bin of their size distributions.

### jpfp3d

[integer 4-D array, {{{(dat(ix, iy, iz3, ipar), ix=1, nx), iy=1, ny), iz3=1, nz3), ipar=1, npar}}]

Indexes of phase functions of media in horizontally inhomogeneous layers. Should be between 0 and npf. If jpfp3d=0, then predefined Rayleigh phase function is used.

[Edit](#)

## External input file filephs

The file for phase function data should be sequential access file. The data block to be read is recognized after a line containing the following "signature":

```
%mdlphs
```

After the signature is found, the data are read by the following Fortran codes;

```
read (iu, *) npf      ! number of phase functions
do ipf = 1, npf
  read (iu, *)
  read (iu, *) nang(ipf)      ! number of angles
  read (iu, *)
  do iang = 1, nang(ipf)
    read (iu, *) ang(iang, ipf), phs(iang, ipf)      ! angle in degree, and phase function
  end do
end do
```

Example data files can be found in the "examples" directory. See also namelist mdlphsfor details of each variable.

[Edit](#)

## External input file filesfc

The file for surface property data should be sequential access file. The data block to be read is recognized after a line that contains the following line:

```
%mdlisfc
```

After the signature is found, the data are read by the following Fortran codes;

```
read (iu, *)
read (iu, *) ((tmps2d(ixb, iyb), ixb = 1, nxb), iyb = 1, nyb) ! temperature
read (iu, *)
read (iu, *) ((jsfc2d(ixb, iyb), ixb = 1, nxb), iyb = 1, nyb) ! surface BRDF model index
read (iu, *, end=1)
do ip = 1, 5
  read (iu, *, err=1, end=1) ((psfc2d(ixb, iyb, ip), ixb = 1, nxb), iyb = 1, nyb)
  !// surface parameters
end do
1 continue
```

Example data files can be found in the "examples" directory. See also namelist mdlisfcfor details of each variable.

[Edit](#)

## External input file filea3d

The file for 3-D atmospheric property data should be sequential access text format if ifmt3d=0 or direct access binary format if ifmt3d=1.

### Sequential access format

When sequential access text format (ifmt3d=0), the data block to be read is recognized after a line that contains the following line:

```
%mdlphs
```

After the signature is found, the data are read by the following Fortran codes:

```
read (iu, *)
if (jtmplev .eq. 0) then
  iz3min = 1
else
  iz3min = 0
end if
do iz3 = iz3min, nz3
  read (iu, *) ((tmpa3d(ix, iy, iz3), ix = 1, nx), iy = 1, ny) ! temperature
end do
do ikd = 1, nkd
  read (iu, *)
  do iz3 = 1, nz3
    read (iu, *) ((absg3d(ix, iy, iz3, ikd), ix = 1, nx), iy = 1, ny)
    !// gas absorption coefficient
  end do
end do
do ipar = 1, npar
  read (iu, *)
  do iz3 = 1, nz3
    read (iu, *) ((extp3d(ix, iy, iz3, ipar), ix = 1, nx), iy = 1, ny)
    !// extinction coefficient
  end do
  read (iu, *)
  do iz3 = 1, nz3
    read (iu, *) ((omgp3d(ix, iy, iz3, ipar), ix = 1, nx), iy = 1, ny)
    !// single scattering albedo
  end do
  read (iu, *)
  do iz3 = 1, nz3
    read (iu, *) ((rpfp3d(ix, iy), ix = 1, nx), iy = 1, ny)
    !// index of phase function
    do iy = 1, ny
      do ix = 1, nx
        jfp3d(ix, iy, iz3, ipar) = int(rpfp3d(ix, iy) + 0.00001)
        !// real-to-integer conversion
      end do
    end do
  end do
end do
end do
```

Note that in the last part of the above code, the real values for rpfp3d are rounded to integer values for jfp3d. See also namelist mdla3d for details of each variable. Example data files can be found in the "examples" directory.

### Direct access format

When direct access binary format (ifmt3d=1), record length is set as nrec=nbyte\*nx\*ny. The data are read as the following Fortran codes:

```
irec = 1
if (jtmplev .eq. 0) then
  iz3min = 1
else
  iz3min = 0
end if
do iz3 = iz3min, nz3
  read (iu, rec=irec) ((tmpa3d(ix, iy, iz3), ix = 1, nx), iy = 1, ny) ! temperature
  irec = irec + 1
end do
do ikd = 1, nkd
  do iz3 = 1, nz3
    read (iu, rec=irec) ((absg3d(ix, iy, iz3, ikd), ix = 1, nx), iy = 1, ny)
    !// gas absorption coefficient
    irec = irec + 1
  end do
end do
```

```
    end do
end do
do ipar = 1, npar
do iz3 = 1, nz3
read (iu, rec=irec) ((extp3d(ix, iy, iz3, ipar), ix = 1, nx), iy = 1, ny)
  !// extinction coefficient
  irec = irec + 1
end do
do iz3 = 1, nz3
read (iu, rec=irec) ((omgp3d(ix, iy, iz3, ipar), ix = 1, nx), iy = 1, ny)
  !// single scattering albedo
  irec = irec + 1
end do
do iz3 = 1, nz3
read (iu, rec=irec) ((rpfp3d(ix, iy), ix = 1, nx), iy = 1, ny)
  !// index of phase function
do iy = 1, ny
do ix = 1, nx
  jfpf3d(ix, iy, iz3, ipar) = int(rpfp3d(ix, iy) + 0.00001)
  !// real-to-integer conversion
end do
end do
  irec = irec + 1
end do
end do
```

[Edit](#)

[Next](#)

---

---