

Synopsis of FLake Routines*

October 10, 2008

The synopsis presents a short description of the program units of the lake model FLake and of the atmospheric surface-layer parameterization scheme SfcFlx that is used to compute fluxes of momentum and heat at the air-water interface. A detailed scientific description of FLake is given in Mironov (2008). Further useful comments can be found directly in the source codes of FLake and of SfcFlx. Some results from single-column experiments with FLake and SfcFlx are described in Mironov et al. (2003). A thermodynamic sea ice model (parameterization scheme) for the global NWP system GME (Majewski et al. 2002) of the German Weather Service (DWD), that is broadly similar to the ice parameterization scheme of FLake, is described in Mironov and Ritter (2004).

FLake can be used in a stand-alone mode as a single-column lake model, or as a lake parameterization module in three-dimensional numerical weather prediction (NWP) or climate modelling systems. A convenient means to organise communication between FLake and a driving model that calls FLake is provided by FLake Interface. FLake Interface is a communication routine. Its details differ with the application, and it is the user responsibility to organise the use of FLake in the most convenient way. An example of FLake Interface for single-column applications is briefly described in what follows.

Conventions

The synopsis follows typographic conventions outlined below.

- Bold** is used for file names.
- SMALL CAPS** is used for names of modules and routines.
- Slanted* is used for names of variables, parameters, arrays, etc.

The FORTRAN 90 source code of FLake and SfcFlx contains a number of comment lines. Lines embraced/marked with “!_dm” are DM’s comments that may appear to be helpful to a user. Lines embraced/marked with “!_dev” are to be replaced (in the future versions of FLake) as improved physical parameterizations are developed and tested. Lines embraced/marked with “!_dbg” are used for debugging purposes only. Lines embraced/marked with “!_nu” are not used.

*Corresponding author address: Dmitrii V. Mironov, Deutscher Wetterdienst, Meteorologische Analyse und Modellierung, FE14, Frankfurter Str. 135, D-63067 Offenbach am Main, Germany. Phone +49-69-8062 2705, fax +49-69-8062 3721, e-mail dmitrii.mironov@dwd.de.

Modules FLAKE and SFCFLX described below contain a number of routines (subroutines and functions). The source codes of those routines are stored in separate include files that have an extension **incf**. In the FLake version implemented into the limited-area NWP system COSMO (previously referred to as LM, Steppeler et al. 2003), no separate **incf** files are used.

References

- Majewski, D., D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner, 2002: Icosahedral-hexagonal gridpoint model GME: description and high-resolution tests. *Mon. Weather Rev.*, **130**, 319–338.
- Mironov, D. V., 2008: Parameterization of lakes in numerical weather prediction. Description of a lake model. *COSMO Technical Report*, No. 11, Deutscher Wetterdienst, Offenbach am Main, Germany, 41 pp.
- Mironov, D., and B. Ritter, 2004: *A New Sea Ice Model for GME*. Technical Note, Deutscher Wetterdienst, Offenbach am Main, Germany, 12 pp.
- Mironov, D., A. Terzhevik, F. Beyrich, E. Heise, and H. Lohse, 2003: A two-layer lake model for use in numerical weather prediction. *Proc. of the Baltic HIRLAM Workshop*, 17 – 20 November 2003, St. Petersburg, Russia, 83–85.
- Steppeler, J., G. Doms, U. Schättler, H. W. Bitzer, A. Gassmann, U. Damrath, and G. Gregoric, 2003: Meso-gamma scale forecasts using the non-hydrostatic model LM. *Meteorol. Atmos. Phys.*, **82**, 75–96.

FLake Interface

FLake Interface represents a convenient means to organise communication between FLake and a driving model that calls FLake. Details of FLake Interface differ with the application, and it is the user responsibility to organise the use of FLake in the most convenient way. An example of FLake Interface for single-column applications is described below. FLake Interface is a communication routine that does not contain any Flake physics. It does, however, contain a number of variables that have to do with the physics of the atmosphere-lake interface, namely, the surface albedo of water, ice and snow with respect to solar radiation. It also contains the optical characteristics of water, ice and snow. Values of these variables may be computed within FLake Interface. The details are given below.

SUBROUTINE FLAKE_INTERFACE

Modules used: DATA_PARAMETERS, FLAKE, FLAKE_ALBEDO_REF,
FLAKE_DERIVEDTYPES, FLAKE_PARAMETERS, FLAKE_PARAMOPTIC_REF, SFCFLX

Contained in: a user-specified module, can also be used as a stand-alone routine

Called by: a driving model

File: **src_flake_interface_1D.f90**

The input arguments with INTENT (IN):

<i>dMsnowdt_in</i>	rate of snow accumulation ($\text{kg}\cdot\text{m}^{-2}\text{s}^{-1}$)
<i>L_atm_in</i>	solar radiation flux at the surface ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$)
<i>Q_atm_lw_in</i>	long-wave radiation flux from the atmosphere ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$)
<i>height_u_in</i>	height above the lake surface where the wind speed is measured (m)
<i>height_tq_in</i>	height where temperature and humidity are measured (m)
<i>U_a_in</i>	wind speed at $z=\text{height}_u_in$ ($\text{m}\cdot\text{s}^{-1}$)
<i>T_a_in</i>	air temperature at $z=\text{height}_tq_in$ (K)
<i>q_a_in</i>	air specific humidity at $z=\text{height}_tq_in$ (dimensionless)
<i>P_a_in</i>	surface air pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$)
<i>depth_w</i>	lake depth (m)
<i>fetch</i>	wind fetch (m)
<i>depth_bs</i>	depth of the thermally active layer of bottom sediments (m)
<i>T_bs</i>	temperature at the outer edge of the thermally active layer of bottom sediments (K)
<i>par_Coriolis</i>	the Coriolis parameter (s^{-1})
<i>del_time</i>	time step (s)
<i>T_snow_in</i>	temperature at the air-snow interface (K)

<i>T_ice_in</i>	temperature at the snow-ice or air-ice interface (K)
<i>T_mnw_in</i>	mean temperature of the water column (K)
<i>T_wML_in</i>	mixed-layer temperature (K)
<i>T_bot_in</i>	temperature at the water-bottom sediment interface (K)
<i>T_B1_in</i>	temperature at the bottom of the upper layer of the sediment (K)
<i>C_T_in</i>	shape factor with respect to the temperature profile in the thermocline (dimensionless)
<i>h_snow_in</i>	snow thickness (m)
<i>h_ice_in</i>	ice thickness (m)
<i>h_ML_in</i>	thickness of the mixed-layer (m)
<i>H_B1_in</i>	thickness of the upper layer of bottom sediments (m)
<i>T_sfc_p</i>	surface temperature at the previous time step (K) that is equal to the temperature of the surface of water, of ice or of snow

The input/output arguments with INTENT (INOUT):

<i>albedo_water</i>	water surface albedo with respect to solar radiation (dimensionless)
<i>albedo_ice</i>	ice surface albedo with respect to solar radiation (dimensionless)
<i>albedo_snow</i>	snow surface albedo with respect to solar radiation (dimensionless)
<i>opticpar_water</i>	optical characteristics of water (variable of type <i>opticpar_medium</i>)
<i>opticpar_ice</i>	optical characteristics of ice (variable of type <i>opticpar_medium</i>)
<i>opticpar_snow</i>	optical characteristics of snow (variable of type <i>opticpar_medium</i>)

The output arguments with INTENT (OUT):

<i>T_snow_out</i>	temperature at the air-snow interface (K)
<i>T_ice_out</i>	temperature at the snow-ice or air-ice interface (K)
<i>T_mnw_out</i>	mean temperature of the water column (K)
<i>T_wML_out</i>	mixed-layer temperature (K)
<i>T_bot_out</i>	temperature at the water-bottom sediment interface (K)
<i>T_B1_out</i>	temperature at the bottom of the upper layer of the sediment (K)
<i>C_T_out</i>	shape factor with respect to the temperature profile in the thermocline (dimensionless)
<i>h_snow_out</i>	snow thickness (m)
<i>h_ice_out</i>	ice thickness (m)
<i>h_ML_out</i>	thickness of the mixed-layer (m)
<i>H_B1_out</i>	thickness of the upper layer of bottom sediments (m)

T_{sfc_n} the updated surface temperature (K), that is the temperature of the surface of water, of ice or of snow, as the case may be

Description: FLake Interface organises a time step of FLake. FLake routines are called and various quantities are computed in an appropriate order. The USE statements ensure a correct data exchange between various FLake routines. A few local variables declared in FLAKE_INTERFACE are used to store intermediate results. FLAKE_INTERFACE is a plug-compatible routine from the viewpoint of a driving model. All one has to do to organise time stepping is to call FLAKE_INTERFACE in a DO loop with respect to time, passing the necessary information through the input arguments and getting the results through the output arguments of FLAKE_INTERFACE. In this way, FLake can essentially be used as a black box.

A time step of FLake is organised as follows. First, the values of albedo of water, of ice and of snow with respect to solar radiation, and the optical characteristics of water, of ice and of snow are set/computed. Notice these variables are the input arguments of FLAKE_INTERFACE with INTENT (INOUT). They can be either passed to FLAKE_INTERFACE from the driving model, or set/computed within FLAKE_INTERFACE, e.g. using default values offered by FLake. In order to compute the ice surface albedo with respect to solar radiation, an empirical formulation is used that relates the ice albedo to the ice surface temperature. That formulation is meant to implicitly account, in a rather approximate way, for the seasonal changes of the ice albedo (see Mironov and Ritter 2004). Although provision is made to model the evolution of snow cover over the lake ice, the snow module of FLake has not been sufficiently tested and is not used at the moment. The snow albedo is formally set equal to the ice albedo. Parameters *opticpar_ice_opaque* and *opticpar_snow_opaque* (module FLAKE_PARAMOPTIC_REF) are used to specify the optical characteristics of ice and snow. This is a recommended choice.

Next, the initial values of FLake prognostic variables are set. That is, FLake variables with the suffix “_p_flk” are assigned the values of the corresponding input arguments of FLAKE_INTERFACE with the suffix “_in”.

Then, the rate of snow accumulation is set by assigning the FLake variable *dMsnowdt_flk* a value of the input argument *dMsnowdt_in*. As mentioned above, the snow module of FLake is not used at the time being. No switch is required to deactivate the snow module. All one has to do is to pass a zero value of *dMsnowdt_in* to FLAKE_INTERFACE. This choice can be hard-coded by explicitly setting *dMsnowdt_flk* to zero in FLAKE_INTERFACE.

Then, the fluxes of heat and of momentum are computed. The downward flux of solar radiation and the downward long-wave radiation flux from the atmosphere are external parameters for the lake model. They do not depend of the surface temperature and are, therefore, not part of the solution. These fluxes are provided to FLAKE_INTERFACE by the driving model through the input arguments *L_atm_in* and *Q_atm_lw_in*. Fluxes of long-wave radiation from the surface, of sensible and latent heat and of momentum depend on the surface temperature and are, therefore, part of the solution. These fluxes are computed in FLAKE_INTERFACE. Notice the sign of fluxes. The SfcFlx routines assign a positive sign to fluxes if they are directed upwards, whereas in the FLake

routines the downward fluxes are considered to be positive.

The FLake variable L_{atm_flk} is assigned a value of the input argument L_{atm_in} . The subroutine FLAKE_RADFLUX is then called to compute the flux of solar radiation at various depths and the solar-radiation fluxes averaged over the mixed layer and over the thermocline are computed. Notice that the effect of albedo is accounted for within FLAKE_RADFLUX, whereas L_{atm_in} is the incident flux of solar radiation at the lake surface.

As different from solar heating that has a volumetric character, the sum of the long-wave radiation fluxes and of the fluxes of sensible and latent heat enters the FLake equations through the boundary condition at the lake surface. This surface flux is stored in the FLake variable Q_{w_flk} . Q_{w_flk} is first assigned a value of the input argument $Q_{atm_lw_in}$. Then the upward long-wave radiation flux from the surface computed with the function SFCFLX_LWRADWSFC is subtracted. Then the subroutine SFCFLX_MOMSENLAT is called to compute fluxes of momentum and of sensible and latent heat. The surface momentum flux is used to compute the friction velocity in the surface layer of water. The fluxes of sensible and latent heat are added (with the appropriate sign) to Q_{w_flk} . The flux calculation is completed by setting the heat flux at the air-snow interface, Q_{snow_flk} , and the heat flux at the snow-ice or air-ice interface, Q_{ice_flk} , as dependent on whether ice and/or snow is present. If ice is present, Q_{w_flk} is reset to zero.

Then the subroutine FLAKE_DRIVER is called that advances FLake variables one time step. After this call, the FLake prognostic variables with the suffix “_n_flk” contain updated values.

Finally, the output arguments of FLAKE_INTERFACE, i.e. the variables with the suffix “_out”, are assigned the values of the corresponding FLake variables with the suffix “_n_flk”. The updated surface temperature T_{sfc_n} is set equal to either the water-surface temperature, or to the surface temperature of ice or snow. Since the skin effects are not considered in the present version of FLake, the water surface temperature is equal to the mixed-layer temperature.

Given the above example of FLake Interface for single-column applications, developing a FLake Interface for use in a three-dimensional NWP, climate modelling or other numerical prediction system is straightforward. The above calculations are organised in a DO loop over horizontal grid-points, where the calculations are actually performed for the grid-points where lakes are present. In this way, a driving atmospheric model should call FLake Interface only once, passing the required variables and parameters to FLake Interface as two-dimensional fields. A user can use her/his own routines to compute fluxes of momentum and of heat instead of the SfcFlx routines.

Routines of the Lake Model FLake

MODULE DATA_PARAMETERS

Modules used: none

Contains: none

Used by: all FLake and SfcFlx modules and procedures

File: **data_parameters.f90**

Description: In this module, that is taken from COSMO, global kind parameters are defined. Values of integer parameters *ireals* and *iintegers* are set, using intrinsic procedures SELECTED_REAL_KIND and KIND, respectively.

MODULE FLAKE

Modules used: DATA_PARAMETERS

Contains: SUBROUTINE FLAKE_DRIVER, SUBROUTINE FLAKE_RADFLUX,
FUNCTION FLAKE_BUOYPAR, FUNCTION FLAKE_SNOWDENSITY,
FUNCTION FLAKE_SNOWHEATCONDUCT

Used by: FLAKE_INTERFACE

File: **flake.f90**

Description: Main program unit of the lake model FLake. FLake variables that are directly accessible to all FLake procedures contained in this module are declared. These variables should be made accessible to FLake Interface (or to any other user-specified routine that calls FLake procedures) through the respective USE statement. All variables declared in module FLAKE have a suffix “_flk”. Two variables are declared for each FLake prognostic quantity. Those with the suffix “_p_flk” contain values of prognostic quantities at the previous time step, whereas the variables with the suffix “_n_flk” contain updated values.

MODULE FLAKE_ALBEDO_REF

Modules used: DATA_PARAMETERS

Contains: none

Used by: FLAKE_INTERFACE

File: **flake_albedo_ref.f90**

Description: Reference (default) values of albedo for the lake water, lake ice, and snow are specified. As in FLAKE_PARAMOPTIC_REF, two ice categories, viz. white ice and blue ice, and two snow categories, viz. dry snow and melting snow, are used. A value of the dimensionless constant *c_albice_MR* in the interpolation formula for the ice surface albedo with respect to solar radiation is specified (see FLAKE_INTERFACE).

MODULE FLAKE_CONFIGURE

Modules used: DATA_PARAMETERS

Contains: none

Used by: FLAKE_DRIVER

File: **flake_configure.f90**

Description: Switches that configure FLake are set. At the moment, the only switch used is *lflk_botsed_use*. Setting *lflk_botsed_use*=TRUE. indicates that the bottom-sediment scheme is used to compute the depth penetrated by the thermal wave, the temperature at this depth, and the bottom heat flux. Otherwise, the heat flux at the water-bottom sediment interface is set to zero, the depth penetrated by the thermal wave is set to a default value specified in this module (parameter *rflk_depth_bs_ref*), and the temperature at that depth is set to the temperature of maximum density of the fresh water.

MODULE FLAKE_DERIVEDTYPES

Modules used: DATA_PARAMETERS

Contains: none

Used by: FLAKE_INTERFACE, FLAKE_PARAMOPTIC_REF, FLAKE_RADFLUX

File: **flake_derivedtypes.f90**

Description: Derived type *opticpar_medium* used by other FLake program units is defined. Parameters (variables) of type *opticpar_medium* are used to specify optical characteristics of lake water, assuming an exponential decay law for the flux of solar radiation. Each parameter (variable) of type *opticpar_medium* consists of an integer (KIND=iintegers) *nband_optic* that specifies the number of wave-length bands used, and of two real (KIND=ireals) one-dimensional arrays of length *nband_optic_max* that specify fractions of total solar radiation flux and extinction coefficients for each wave-length band, *frac_optic* and *extincoef_optic*, respectively. An integer (KIND=iintegers) parameter *nband_optic_max* is a maximum allowable value of the wave-length bands.

MODULE FLAKE_PARAMETERS

Modules used: DATA_PARAMETERS

Contains: none

Used by: FLAKE_INTERFACE, FLAKE_BUOYPAR, FLAKE_DRIVER, FLAKE_RADFLUX, FLAKE_SNOWDENSITY, FLAKE_SNOWHEATCONDUCT, SFCFLX

File: **flake_parameters.f90**

Description: Values of empirical (disposable) constants and parameters of FLake, of several thermodynamic parameters, and of security constants are specified. Recommended values are given. They are determined on the basis of physical reasoning,

empirical evidence, and sensitivity experiments. A user should exercise caution when changing values of disposable parameters of FLake (see FLake description).

MODULE FLAKE_PARAMOPTIC_REF

Modules used: DATA_PARAMETERS, FLAKE_DERIVEDTYPES

Contains: none

Used by: FLAKE_INTERFACE

File: **flake_paramoptic_ref.f90**

Description: Reference (default) values of the optical characteristics of the lake water, lake ice, and snow are specified, assuming an exponential decay law for the flux of solar radiation. These default values may be used if no information about the optical characteristics of the lake in question is available. In the simplest one-band exponential approximation of the decay law, parameter *opticpar_water_ref*, the extinction coefficient for water is set to a large value, leading to the absorption of 95% of the incoming radiation within the uppermost 1 m of the lake water. An example of two-band approximation, parameter *opticpar_water_trans*, suitable for transparent lake water is also given. Very large values of the extinction coefficients for ice and snow, parameters *opticpar_ice_opaque* and *opticpar_snow_opaque*, respectively, are used to prevent penetration of the solar radiation through the snow-ice cover. The above parameters specifying the optical characteristics of the medium in question are of derived type *opticpar_medium* that is defined in module FLAKE_DERIVEDTYPES. In case the number of wave-length bands used (integer *nband_optic*) exceeds one, a user must ensure that fractions of total solar radiation flux add up to 1. Unused fractions (i.e. elements of the array *frac_optic* with indices from *nband_optic*+1 through *nband_optic_max*) are set to zero. Unused extinction coefficients are set to a large value. Notice that the first extinction coefficient (element of the array *extincoef_optic* with index 1) is treated as typical extinction coefficient for the lake in question. It is used in subroutine FLAKE_DRIVER to estimate the equilibrium depth of the surface-cooling-driven convectively-mixed layer whose deepening is arrested by the volumetric solar heating.

SUBROUTINE FLAKE_DRIVER

Modules used: FLAKE_PARAMETERS, FLAKE_CONFIGURE (DATA_PARAMETERS is used by FLAKE)

Contained in: MODULE FLAKE

Called by: FLAKE_INTERFACE

File: **flake_driver.incf**

Description: Main driving routine of the lake model FLake where computations are performed. It advances FLake variables one time step. At the moment, the Euler explicit scheme is used.

Computations start with setting the time-rate-of-change of prognostic variables to zero

(local variables with the suffix “_dt”), and setting prognostic variables (suffix “_n_flk”) to their values at the previous time step (suffix “_p_flk”). This is a security measure required to avoid spurious changes of prognostic variables when FLake is used within a three-dimensional driving model, e.g. to avoid spurious generation of ice at the neighbouring lake points. Next, the heat flux through the ice-water interface, a generalised heat flux scale utilised in calculations of the convective boundary-layer evolution, and the heat flux through the water-bottom sediment interface are computed. Then, computations proceed to advance the thickness and the temperature of ice and of snow, the mean temperature of the water column, the mixed-layer depth, the mixed-layer temperature, the bottom temperature, and the shape factor with respect to the temperature profile in the thermocline. Then, the depth of the upper layer of bottom sediments penetrated by the thermal wave and the temperature at that depth are computed. Finally, the lake surface temperature is updated. That is, it is set equal to either the water-surface temperature, or to the surface temperature of ice or snow.

The source code of FLAKE_DRIVER is furnished with comment lines that should help a user to understand the code structure. See also the scientific description of FLake in Mironov (2008).

The input arguments of FLAKE_DRIVER with INTENT (IN) are the lake depth (m), depth of the thermally active layer of bottom sediments (m), temperature at the outer edge of the thermally active layer of bottom sediments (K), the Coriolis parameter (s^{-1}), typical extinction coefficient of the lake water with respect to solar radiation (m^{-1}), time step (s), and the surface temperature at the previous time step (K) that is equal to the temperature of the surface of water, of ice or of snow.

The output argument of FLAKE_DRIVER with INTENT (OUT) is the updated surface temperature (K), that is the temperature of the surface of water, of ice or of snow, as the case may be. Other results of calculations are available to the FLake routines through the variables declared in module FLAKE (suffix “_flk”).

SUBROUTINE FLAKE_RADFLUX

Modules used: FLAKE_DERIVEDTYPES, FLAKE_PARAMETERS (DATA_PARAMETERS is used by FLAKE)

Contained in: MODULE FLAKE

Called by: FLAKE_INTERFACE

File: **flake_radflux.incf**

Description: Computes the flux of solar radiation at the snow-ice, ice-water, air-water, mixed layer-thermocline and water column-bottom sediment interfaces, the mean radiation flux over the mixed layer, and the mean radiation flux over the thermocline. An exponential decay law for the flux of solar radiation is used.

The input arguments of FLAKE_RADFLUX with INTENT (IN) are the lake depth (m), albedo of the water surface (dimensionless), albedo of the ice surface (dimensionless), albedo of the snow surface (dimensionless), and three parameters of derived type *opticpar_medium*, namely, *opticpar_water*, *opticpar_ice* and *opticpar_snow*, that specify op-

tical characteristics of water, ice and snow, respectively. Derived type *opticpar_medium* is defined in module FLAKE_DERIVEDTYPES. Variables used in calculations are declared in module FLAKE (suffix “_flk”).

Subroutine FLAKE_RADFLUX has no output arguments. Results of calculations are available to other routines through the variables declared in module FLAKE (suffix “_flk”).

FUNCTION FLAKE_BUOYPAR

Contained in: MODULE FLAKE

Modules used: FLAKE_PARAMETERS (DATA_PARAMETERS is used by FLAKE)

Called by: FLAKE_DRIVER

File: **flake_buoypar.incf**

Description: Computes the buoyancy parameter ($\text{m}\cdot\text{s}^{-2}\text{K}^{-1}$) as function of the water temperature (K). A quadratic equation of state for the fresh-water is used.

FUNCTION FLAKE_SNOWDENSITY

Contained in: MODULE FLAKE

Modules used: FLAKE_PARAMETERS (DATA_PARAMETERS is used by FLAKE)

Called by: FLAKE_DRIVER, FLAKE_SNOWHEATCONDUCT

File: **flake_snowdensity.incf**

Description: Computes snow density ($\text{kg}\cdot\text{m}^{-3}$) as function of snow thickness (m). A simple empirical approximation is utilised that is similar to the approximation used in an old two-layer soil model of COSMO. It should be replaced as an improved parameterization becomes available.

FUNCTION FLAKE_SNOWHEATCONDUCT

Contained in: MODULE FLAKE

Modules used: FLAKE_PARAMETERS (DATA_PARAMETERS is used by FLAKE)

Called by: FLAKE_DRIVER

File: **flake_snowheatconduct.incf**

Description: Computes snow heat conductivity ($\text{J}\cdot\text{m}^{-1}\cdot\text{s}^{-1}\text{K}^{-1} = \text{kg}\cdot\text{m}\cdot\text{s}^{-3}\text{K}^{-1}$) as function of snow thickness (m). A simple empirical approximation is utilised that is similar to the approximation used in an old two-layer soil model of COSMO. It should be replaced as an improved parameterization becomes available.

Routines of the Surface-Layer Scheme SfcFlx

MODULE SFCFLX

Modules used: DATA_PARAMETERS, FLAKE_PARAMETERS

Contains: SUBROUTINE SFCFLX_MOMSENLAT, SUBROUTINE SFCFLX_ROUGHNESS,
FUNCTION SFCFLX_LWRADATM, FUNCTION SFCFLX_LWRADWSFC,
FUNCTION SFCFLX_RHOAIR, FUNCTION SFCFLX_SATWVPRES,
FUNCTION SFCFLX_SPECHUM, FUNCTION SFCFLX_WVPRESWETBULB

Used by: FLAKE_INTERFACE

File: **SfcFlx.f90**

Description: Main program unit of the atmospheric surface-layer parameterization scheme SfcFlx. Values of dimensionless constants and parameters, of several thermodynamic parameters, and of security and utility constants are specified. A number of variables are declared. Notice that several parameters specified in module FLAKE_PARAMETERS are used.

SUBROUTINE SFCFLX_MOMSENLAT

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS is used by SFCFLX)

Called by: FLAKE_INTERFACE

File: **SfcFlx_momsenlat.incf**

Description: Computes fluxes of momentum and of sensible and latent heat at the air-water or air-ice (air-snow) interface. The Monin-Obukhov similarity relations are used to compute turbulent fluxes. The surface roughness lengths with respect to wind speed, potential temperature and specific humidity are computed in SFCFLX_ROUGHNESS. The transcendental equations are solved iteratively using the Newton method. A “1/3” power law in terms of Nusselt and Rayleigh numbers is used to compute fluxes of sensible and latent heat in case of free convection. In case of strong static stability in the surface air layer, when the gradient Richardson number exceeds its critical value and the Monin-Obukhov similarity relations yield zero fluxes, crude estimates of fluxes of momentum and of sensible and latent heat are obtained, assuming that the transport of momentum, heat and mass in the surface air layer is controlled by the molecular transfer mechanisms. A decision between turbulent and molecular fluxes and between fluxes in forced and free convection is made on the basis of flux magnitude. For example, two estimates of the sensible heat flux are computed, one with the Monin-Obukhov similarity relations and the other one with the free convection heat-mass transfer law. The estimate is then chosen that is greater in magnitude. This procedure ensures a gradual transition between various flow regimes.

The input arguments of SFCFLX_MOMSENLAT with INTENT (IN) are height above the surface where the wind speed is measured/computed (m), height above the surface

where temperature and humidity are measured/computed (m), (typical) wind fetch (m), wind speed ($\text{m}\cdot\text{s}^{-1}$), air temperature (K), air specific humidity (dimensionless), surface temperature (K), surface air pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$), and ice thickness (m). The surface temperature is the temperature of the surface of water, of ice or of snow, as the case may be.

The output arguments of `SFCFLX_MOMSENLAT` with `INTENT (OUT)` are momentum flux ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$), sensible heat flux ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$), latent heat flux ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$), and flux of water vapour ($\text{kg}\cdot\text{m}^{-2}\text{s}^{-1}$). All fluxes are positive when directed upwards.

SUBROUTINE `SFCFLX_ROUGHNESS`

Contained in: `MODULE SFCFLX`

Modules used: none (`DATA_PARAMETERS` is used by `SFCFLX`)

Called by: `SFCFLX_MOMSENLAT`

File: **`SfcFlx_roughness.incf`**

Description: Computes the water-surface or the ice-surface roughness lengths with respect to wind speed, potential temperature, and specific humidity. Apart from the roughness lengths, the fetch-dependent Charnock parameter and the threshold value of friction velocity in the surface air layer, that is used to discriminate between the aerodynamically smooth and aerodynamically rough regimes of the flow over the water surface, are computed.

The input arguments of `SFCFLX_ROUGHNESS` with `INTENT (IN)` are typical wind fetch (m), wind speed ($\text{m}\cdot\text{s}^{-1}$), friction velocity in the surface air layer ($\text{m}\cdot\text{s}^{-1}$), and ice thickness (m).

The output arguments of `SFCFLX_ROUGHNESS` with `INTENT (OUT)` are the roughness length (m) with respect to wind speed, the roughness length (m) with respect to potential temperature, the roughness length (m) with respect to specific humidity, the fetch-dependent Charnock parameter (dimensionless), and the threshold value of friction velocity ($\text{m}\cdot\text{s}^{-1}$) in the surface air layer.

FUNCTION `SFCFLX_LWRADATM`

Contained in: `MODULE SFCFLX`

Modules used: none (`DATA_PARAMETERS` is used by `SFCFLX`)

Called by: – (any user-specified routine)

File: **`SfcFlx_lwradatm.incf`**

Description: Computes the downward long-wave radiation flux ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$) from the atmosphere as function of air temperature (K), water vapour pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$), total cloud cover (fraction of one), and low-level cloud cover (fraction of one). Simple empirical formulations for the water-vapour correction function and the

cloud correction function are used. If no data on low-level cloud cover is available, that is only total cloud cover is known, the low-level cloud cover must be set to a negative value (e.g. -1).

FUNCTION SFCFLX_LWRADWSFC

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS is used by SFCFLX)

Called by: FLAKE_INTERFACE

File: **SfcFlx_lwradwsfc.incf**

Description: Computes the upward long-wave radiation flux ($\text{W}\cdot\text{m}^{-2} = \text{kg}\cdot\text{s}^{-3}$) from the surface as function of temperature (K). The surface emissivity with respect to the long-wave radiation is specified in module SFCFLX.

FUNCTION SFCFLX_RHOAIR

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS is used by SFCFLX)

Called by: SFCFLX_MOMSENLAT

File: **SfcFlx_rhoair.incf**

Description: Computes the air density ($\text{kg}\cdot\text{m}^{-3}$) as function of temperature (K), specific humidity (dimensionless), and pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$).

FUNCTION SFCFLX_SATWVPRES

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS and FLAKE_PARAMETERS are used by SFCFLX)

Called by: SFCFLX_MOMSENLAT

File: **SfcFlx_satwvpres.incf**

Description: Computes saturation water vapour pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$) over the water surface or over the ice surface as function of temperature (K) and ice thickness (m). A minimum ice thickness used to discriminate between the water surface and the ice surface is specified in module FLAKE_PARAMETERS.

FUNCTION SFCFLX_SPECHUM

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS is used by SFCFLX)

Called by: SFCFLX_MOMSENLAT

File: **SfcFlx_spechum.incf**

Description: Computes specific humidity (dimensionless) as function of water vapour pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$) and air pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$).

FUNCTION SFCFLX_WVPRESWETBULB

Contained in: MODULE SFCFLX

Modules used: none (DATA_PARAMETERS is used by SFCFLX)

Called by: – (any user-specified routine)

File: **SfcFlx_wvpreswetbulb.incf**

Description: Computes water vapour pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$) as function of air temperature (K), wet-bulb temperature (K), saturation vapour pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$) at wet-bulb temperature, and air pressure ($\text{N}\cdot\text{m}^{-2} = \text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-2}$).