MAIC-2

Generated by Doxygen 1.6.3

Wed Apr 20 14:07:47 2011

Contents

Chapter 1

Main Page

1.1 Description

The Mars Atmosphere-Ice Coupler MAIC-2 is a simple, latitudinal model, which consists of a set of parameterisations for the surface temperature, the atmospheric water transport and the surface mass balance (condensation minus evaporation) of water ice. It is driven directly by orbital parameters. A detailed description of the model is given in the publication by Greve et al. (2010).

The model equations of MAIC-2 are discretised by a finite-difference/finite-volume scheme. Coding is done in the programming language Fortran 90.

Required model forcing (as functions of time):

- Obliquity (axial tilt).
- Orbital eccentricity.
- Solar longitude (Ls) of perihelion.

Output (as functions of latitude and time):

- Surface temperature.
- Evaporation rate of water ice.
- Condensation rate of water ice.
- Atmospheric water content.
- Surface mass balance of water ice.
- Ice thickness.

References:

- Greve, R., B. Grieger and O. J. Stenzel. 2010.

  MAIC-2, a latitudinal model for the Martian surface temperature, atmospheric water transport and surface glaciation.

  Planetary and Space Science 58 (6), 931-940.

- MAIC-2 website: http://maic2.greveweb.net/

1.2 Copyright

1.3 License

Chapter 2

Modules Index

2.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 condensation Module Reference

Computation of the condensation rate.

Functions/Subroutines

- subroutine setcondpar (gravity, timescale)

  Setting of parameters.

- subroutine getcond_1 (temp, water, cond, dtime)

  Computation of condensation (removal of water exceeding the saturation pressure at the surface).

- subroutine getcond_2 (temp, water, cond)

  Computation of condensation (continuous, quadratic dependence on humidity).

Variables

- real(dp) g
- real(dp) tau

5.1.1 Detailed Description

Computation of the condensation rate.

5.1.2 Function/Subroutine Documentation

5.1.2.1 subroutine condensation::getcond_1 (real(dp),dimension(:) *temp*, real(dp),dimension(:) *water*, real(dp),dimension(:) *cond*, real(dp) *dtime*)

Computation of condensation (removal of water exceeding the saturation pressure at the surface).

Definition at line 72 of file condensation.f90.

5.1.2.2 subroutine condensation::getcond_2 (real(dp),dimension(:) *temp*, real(dp),dimension(:) *water*, real(dp),dimension(:) *cond*)

Computation of condensation (continuous, quadratic dependence on humidity).

Definition at line 102 of file condensation.f90.

5.1.2.3 subroutine condensation::setcondpar (real(dp),optional *gravity*, real(dp),optional *timescale*)

Setting of parameters.

Definition at line 48 of file condensation.f90.

5.1.3 Variable Documentation

5.1.3.1 real(dp) condensation::g

Definition at line 40 of file condensation.f90.

5.1.3.2 real(dp) condensation::tau

Definition at line 41 of file condensation.f90.

5.2 evaporation Module Reference

Computation of the evaporation rate (buoyant-diffusion approach by Ingersoll).

Functions/Subroutines

- subroutine setevappar_bd (evap_fact_para, gamma_reg_para, R_univ_para, mol_w_para, mol_c_para, diff_w_c_para, visc_c_para, g_para)

  Setting of parameters.

- subroutine getevap_bd (temp, temp_amp, p, H, evap)

  Computation of evaporation.

Variables

- real(dp) evap_fact
- real(dp) gamma_reg
- real(dp) R_univ
- real(dp) mol_w
- real(dp) mol_c
- real(dp) diff_w_c
- real(dp) visc_c
- real(dp) g

5.2.1 Detailed Description

Computation of the evaporation rate (buoyant-diffusion approach by Ingersoll).

5.2.2 Function/Subroutine Documentation

5.2.2.1 subroutine evaporation::getevap_bd (real(dp),dimension(:) *temp*, real(dp),dimension(:) *temp_amp*, real(dp),dimension(:) *p*, real(dp),dimension(:) *H*, real(dp),dimension(:) *evap*)

Computation of evaporation.

Definition at line 113 of file evaporation.f90.

5.2.2.2 subroutine evaporation::setevappar_bd (real(dp),optional *evap_fact_para*, real(dp),optional *gamma_reg_para*, real(dp),optional *R_univ_para*, real(dp),optional *mol_w_para*, real(dp),optional *mol_c_para*, real(dp),optional *diff_w_c_para*, real(dp),optional *visc_c_para*, real(dp),optional *g_para*)

Setting of parameters.

Definition at line 50 of file evaporation.f90.

5.2.3 Variable Documentation

5.2.3.1 real(dp) evaporation::diff_w_c

Definition at line 42 of file evaporation.f90.

5.2.3.2 real(dp) evaporation::evap_fact

Definition at line 42 of file evaporation.f90.

5.2.3.3 real(dp) evaporation::g

Definition at line 42 of file evaporation.f90.

5.2.3.4 real(dp) evaporation::gamma_reg

Definition at line 42 of file evaporation.f90.

5.2.3.5 real(dp) evaporation::mol_c

Definition at line 42 of file evaporation.f90.

5.2.3.6 real(dp) evaporation::mol_w

Definition at line 42 of file evaporation.f90.

5.2.3.7 real(dp) evaporation::R_univ

Definition at line 42 of file evaporation.f90.

5.2.3.8 real(dp) evaporation::visc_c

Definition at line 42 of file evaporation.f90.

5.3 instemp Module Reference

Computation of the daily mean surface temperature of Mars based on obliquity, eccentricity and the anomaly of vernal equinox (local insolation temperature = LIS scheme).

Data Types

- type ins

  Surface temperatures.

Functions/Subroutines

- subroutine setinstemp (o, ecc, ave, obl, sma, sa, sac, op, ct)

  Main subroutine of module instemp.

- real(dp) instam (o, phi)

  Annual mean temperature at latitude phi.

- real(dp) instmax (o, phi)

  Annual maximum temperature at latitude phi.

- real(dp) inst (o, psi, phi)

  Temperature at orbit position psi and latitude phi.

- real(dp) inst1 (o, ipsi, phi)

  Temperature at orbit position ipsi (integer value) and latitude phi.

5.3.1 Detailed Description

Computation of the daily mean surface temperature of Mars based on obliquity, eccentricity and the anomaly of vernal equinox (local insolation temperature = LIS scheme).

5.3.2 Function/Subroutine Documentation

5.3.2.1 real(dp) instemp::inst (type(ins) *o*, real(dp) *psi*, real(dp) *phi*)

Temperature at orbit position psi and latitude phi.

Definition at line 272 of file instemp.f90.

5.3.2.2 real(dp) instemp::inst1 (type(ins) *o*, integer(i4b) *ipsi*, real(dp) *phi*)

Temperature at orbit position ipsi (integer value) and latitude phi.

Definition at line 294 of file instemp.f90.

5.3.2.3 real(dp) instemp::instam (type(ins) *o*, real(dp) *phi*)

Annual mean temperature at latitude phi.

Definition at line 228 of file instemp.f90.

5.3.2.4 real(dp) instemp::instmax (type(ins) *o*, real(dp) *phi*)

Annual maximum temperature at latitude phi.

Definition at line 250 of file instemp.f90.

5.3.2.5 subroutine instemp::setinstemp (type(ins) *o*, real(dp),optional *ecc*, real(dp),optional *ave*, real(dp),optional *obl*, real(dp),optional *sma*, real(dp),optional *sa*, real(dp),optional *sac*, real(dp),optional *op*, real(dp),optional *ct*)

Main subroutine of module instemp.

Definition at line 57 of file instemp.f90.

5.4 maic2_types Module Reference

Declarations of kind types for MAIC-2.

Variables

- integer, parameter i2b = selected_int_kind(4)

  2-byte integers

- integer, parameter i4b = selected_int_kind(9)

  4-byte integers

- integer, parameter sp = kind(1.0)

  single-precision reals

- integer, parameter dp = kind(1.0d0)

  double-precision reals

5.4.1 Detailed Description

Declarations of kind types for MAIC-2.

5.4.2 Variable Documentation

5.4.2.1 integer,parameter maic2_types::dp = kind(1.0d0)

double-precision reals

Definition at line 40 of file maic2_types.F90.

5.4.2.2 integer,parameter maic2_types::i2b = selected_int_kind(4)

2-byte integers

Definition at line 37 of file maic2_types.F90.

5.4.2.3 integer,parameter maic2_types::i4b = selected_int_kind(9)

4-byte integers

Definition at line 38 of file maic2_types.F90.

5.4.2.4 integer,parameter maic2_types::sp = kind(1.0)

single-precision reals

Definition at line 39 of file maic2_types.F90.

5.5 maic2_variables Module Reference

Declarations of global variables for MAIC-2.

Variables

- real(dp), dimension(0:lmax) phi_node

  phi_node(l): Latitude of grid point

- real(dp), dimension(0:lmax) phi_cb1

  phi_cb1(l): Latitude of lower cell boundary

- real(dp), dimension(0:lmax) phi_cb2

  phi_cb2(l): Latitude of upper cell boundary

- real(dp), dimension(0:lmax) cos_phi_cb1

  cos_phi_cb1(l): Cosine of phi_cb1(l)

- real(dp), dimension(0:lmax) cos_phi_cb2

  cos_phi_cb2(l): Cosine of phi_cb2(l)

- real(dp), dimension(0:lmax) sin_phi_cb1

  sin_phi_cb1(l): Sine of phi_cb1(l)

- real(dp), dimension(0:lmax) sin_phi_cb2

  sin_phi_cb2(l): Sine of phi_cb2(l)

- real(dp), dimension(lmax) dphi

  dphi(l): Grid spacing

- real(dp), dimension(lmax) dphi_inv

  dphi_inv(l): Inverse of the grid spacing

- real(dp), dimension(0:lmax) temp_surf

  temp_surf(l): Daily mean surface temperature

- real(dp), dimension(0:lmax) temp_surf_amp

  temp_surf_amp(l): Amplitude of the daily cycle of the surface temperature

- real(dp), dimension(0:lmax) temp_co2

  temp_co2(l): CO2 condensation temperature

- real(dp), dimension(0:lmax) p_surf

  p_surf(l): Surface pressure

- real(dp), dimension(0:lmax) water

water(l): Water content in the atmosphere

- real(dp), dimension(0:lmax) water_new

  (.)_new: New value of quantity (.) computed during an integration step

- real(dp), dimension(0:lmax) cond

  cond(l): Condensation rate

- real(dp), dimension(0:lmax) evap

  evap(l): Evaporation rate

- real(dp), dimension(0:lmax) a_net

  a_net(l): Net surface mass balance of water ice

- real(dp), dimension(0:lmax) H

  H(l): Ice thickness.

- real(dp), dimension(0:lmax) H_new

  (.)_new: New value of quantity (.) computed during an integration step

- real(dp) RHO

  RHO: Density of ice-dust mixture.

- real(dp) RHO_I

  RHO_I: Density of ice.

- real(dp) RHO_W

  RHO_W: Density of pure water.

- real(dp) G

  G: Acceleration due to gravity.

- real(dp) R

  R: Radius of Mars.

- real(dp) rho_inv

  rho_inv: Inverse of the density of ice-dust mixture

- integer(i4b) insol_time_min

  insol_time_min: Minimum time of the data values for the insolation etc.

- integer(i4b) insol_time_stp

  insol_time_stp: Time step of the data values for the insolation etc.

- integer(i4b) insol_time_max

  insol_time_max: Maximum time of the data values for the insolation etc.

- integer(i4b) itercount

  itercount: Counter for the number of time integration steps

- real(dp), dimension(0:100000) insol_ma_90

  insol_ma_90(n): Data for the mean-annual north- or south-polar insolation

- real(dp), dimension(0:100000) obl_data

  obl_data(n): Data for the obliquity

- real(dp), dimension(0:100000) ecc_data

  ecc_data(n): Data for the eccentricity

- real(dp), dimension(0:100000) ave_data

  ave_data(n): Data for the anomaly of vernal equinox (= 360 deg - Ls of perihelion)

- real(dp), dimension(0:100000) cp_data

  cp_data(n): Data for Laskar's climate parameter = eccentricity ∗sin(Laskar's longitude of perihelion from moving equinox), (where Laskar's longitude of perihelion from moving equinox = Ls of perihelion - 180 deg)

- real(dp), dimension(0:ntime) psi_tab

  psi_tab(n): True anomalies (orbital positions with respect to perihelion) over a Martian year

- real(dp), dimension(0:lmax) diff_aux

  diff_aux(l): Auxiliary quantity needed for the diffusional transport

- integer(i4b) n_output

  n_output: Number of specified times for data output

- real(dp) dtime_out

  dtime_out: Time step for data output

- real(dp), dimension(100) time_output

  time_output(n): Specified times for data output

- integer(i4b) iter_out

  iter_out: Intervall of integration steps for data output

- integer(i4b), dimension(100) iter_output

  iter_output(n): Specified integration steps for data output

- real(dp), parameter pi = 3.141592653589793_dp

  pi: Mathematical constant

- real(dp), parameter pi_inv = 1.0_dp/pi *pi_inv: Inverse of pi*

- real(dp), parameter pi_180 = pi/180.0_dp

  pi_180: pi divided by 180 (-> deg to rad)

- real(dp), parameter pi_180_inv = 180.0_dp/pi

  pi_180_inv: 180 divided by pi (-> rad to deg)

- real(dp), parameter eps = 1.0e-05_dp

  eps: Small number

5.5.1 Detailed Description

Declarations of global variables for MAIC-2.

5.5.2 Variable Documentation

5.5.2.1 real(dp),dimension(0:lmax) maic2_variables::a_net

a_net(l): Net surface mass balance of water ice

Definition at line 76 of file maic2_variables.F90.

5.5.2.2 real(dp),dimension(0:100000) maic2_variables::ave_data

ave_data(n): Data for the anomaly of vernal equinox (= 360 deg - Ls of perihelion)

Definition at line 116 of file maic2_variables.F90.

5.5.2.3 real(dp),dimension(0:lmax) maic2_variables::cond

cond(l): Condensation rate

Definition at line 72 of file maic2_variables.F90.

5.5.2.4 real(dp),dimension(0:lmax) maic2_variables::cos_phi_cb1

cos_phi_cb1(l): Cosine of phi_cb1(l)

Definition at line 48 of file maic2_variables.F90.

5.5.2.5 real(dp),dimension(0:lmax) maic2_variables::cos_phi_cb2

cos_phi_cb2(l): Cosine of phi_cb2(l)

Definition at line 50 of file maic2_variables.F90.

5.5.2.6 real(dp),dimension(0:100000) maic2_variables::cp_data

cp_data(n): Data for Laskar's climate parameter = eccentricity ∗sin(Laskar's longitude of perihelion from moving equinox), (where Laskar's longitude of perihelion from moving equinox = Ls of perihelion - 180 deg)

Definition at line 122 of file maic2_variables.F90.

5.5.2.7 real(dp),dimension(0:lmax) maic2_variables::diff_aux

diff_aux(l): Auxiliary quantity needed for the diffusional transport

Definition at line 129 of file maic2_variables.F90.

5.5.2.8 real(dp),dimension(lmax) maic2_variables::dphi

dphi(l): Grid spacing

Definition at line 56 of file maic2_variables.F90.

5.5.2.9 real(dp),dimension(lmax) maic2_variables::dphi_inv

dphi_inv(l): Inverse of the grid spacing

Definition at line 58 of file maic2_variables.F90.

5.5.2.10 real(dp) maic2_variables::dtime_out

dtime_out: Time step for data output

Definition at line 134 of file maic2_variables.F90.

5.5.2.11 real(dp),dimension(0:100000) maic2_variables::ecc_data

ecc_data(n): Data for the eccentricity

Definition at line 113 of file maic2_variables.F90.

5.5.2.12 real(dp),parameter maic2_variables::eps = 1.0e-05_dp

eps: Small number

Definition at line 151 of file maic2_variables.F90.

5.5.2.13 real(dp),dimension(0:lmax) maic2_variables::evap

evap(l): Evaporation rate

Definition at line 74 of file maic2_variables.F90.

5.5.2.14 real(dp) maic2_variables::G

G: Acceleration due to gravity.

Definition at line 91 of file maic2_variables.F90.

5.5.2.15 real(dp),dimension(0:lmax) maic2_variables::H

H(l): Ice thickness.

Definition at line 78 of file maic2_variables.F90.

5.5.2.16 real(dp),dimension(0:lmax) maic2_variables::H_new

(.)_new: New value of quantity (.) computed during an integration step

Definition at line 80 of file maic2_variables.F90.

5.5.2.17 real(dp),dimension(0:100000) maic2_variables::insol_ma_90

insol_ma_90(n): Data for the mean-annual north- or south-polar insolation

Definition at line 109 of file maic2_variables.F90.

5.5.2.18 integer(i4b) maic2_variables::insol_time_max

insol_time_max: Maximum time of the data values for the insolation etc.

Definition at line 104 of file maic2_variables.F90.

5.5.2.19 integer(i4b) maic2_variables::insol_time_min

insol_time_min: Minimum time of the data values for the insolation etc.

Definition at line 100 of file maic2_variables.F90.

5.5.2.20 integer(i4b) maic2_variables::insol_time_stp

insol_time_stp: Time step of the data values for the insolation etc.

Definition at line 102 of file maic2_variables.F90.

5.5.2.21 integer(i4b) maic2_variables::iter_out

iter_out: Intervall of integration steps for data output

Definition at line 138 of file maic2_variables.F90.

5.5.2.22 integer(i4b),dimension(100) maic2_variables::iter_output

iter_output(n): Specified integration steps for data output

Definition at line 140 of file maic2_variables.F90.

5.5.2.23 integer(i4b) maic2_variables::itercount

itercount: Counter for the number of time integration steps

Definition at line 106 of file maic2_variables.F90.

5.5.2.24 integer(i4b) maic2_variables::n_output

n_output: Number of specified times for data output

Definition at line 132 of file maic2_variables.F90.

5.5.2.25 real(dp),dimension(0:100000) maic2_variables::obl_data

obl_data(n): Data for the obliquity

Definition at line 111 of file maic2_variables.F90.

5.5.2.26 real(dp),dimension(0:lmax) maic2_variables::p_surf

p_surf(l): Surface pressure

Definition at line 66 of file maic2_variables.F90.

5.5.2.27 real(dp),dimension(0:lmax) maic2_variables::phi_cb1

phi_cb1(l): Latitude of lower cell boundary

Definition at line 44 of file maic2_variables.F90.

5.5.2.28 real(dp),dimension(0:lmax) maic2_variables::phi_cb2

phi_cb2(l): Latitude of upper cell boundary

Definition at line 46 of file maic2_variables.F90.

5.5.2.29 real(dp),dimension(0:lmax) maic2_variables::phi_node

phi_node(l): Latitude of grid point

Definition at line 42 of file maic2_variables.F90.

5.5.2.30 real(dp),parameter maic2_variables::pi = 3.141592653589793_dp

pi: Mathematical constant

Definition at line 143 of file maic2_variables.F90.

5.5.2.31 real(dp),parameter maic2_variables::pi_180 = pi/180.0_dp

pi_180: pi divided by 180 (-> deg to rad)

Definition at line 147 of file maic2_variables.F90.

5.5.2.32 real(dp),parameter maic2_variables::pi_180_inv = 180.0_dp/pi

pi_180_inv: 180 divided by pi (-> rad to deg)

Definition at line 149 of file maic2_variables.F90.

5.5.2.33 real(dp),parameter maic2_variables::pi_inv = 1.0_dp/pi

pi_inv: Inverse of pi

Definition at line 145 of file maic2_variables.F90.

5.5.2.34 real(dp),dimension(0:ntime) maic2_variables::psi_tab

psi_tab(n): True anomalies (orbital positions with respect to perihelion) over a Martian year

Definition at line 126 of file maic2_variables.F90.

5.5.2.35 real(dp) maic2_variables::R

R: Radius of Mars.

Definition at line 93 of file maic2_variables.F90.

5.5.2.36 real(dp) maic2_variables::RHO

RHO: Density of ice-dust mixture.

Definition at line 85 of file maic2_variables.F90.

5.5.2.37 real(dp) maic2_variables::RHO_I

RHO_I: Density of ice.

Definition at line 87 of file maic2_variables.F90.

5.5.2.38 real(dp) maic2_variables::rho_inv

rho_inv: Inverse of the density of ice-dust mixture

Definition at line 95 of file maic2_variables.F90.

5.5.2.39 real(dp) maic2_variables::RHO_W

RHO_W: Density of pure water.

Definition at line 89 of file maic2_variables.F90.

5.5.2.40 real(dp),dimension(0:lmax) maic2_variables::sin_phi_cb1

sin_phi_cb1(l): Sine of phi_cb1(l)

Definition at line 52 of file maic2_variables.F90.

5.5.2.41 real(dp),dimension(0:lmax) maic2_variables::sin_phi_cb2

sin_phi_cb2(l): Sine of phi_cb2(l)

Definition at line 54 of file maic2_variables.F90.

5.5.2.42 real(dp),dimension(0:lmax) maic2_variables::temp_co2

temp_co2(l): CO2 condensation temperature

Definition at line 64 of file maic2_variables.F90.

5.5.2.43 real(dp),dimension(0:lmax) maic2_variables::temp_surf

temp_surf(l): Daily mean surface temperature

Definition at line 60 of file maic2_variables.F90.

5.5.2.44 real(dp),dimension(0:lmax) maic2_variables::temp_surf_amp

temp_surf_amp(l): Amplitude of the daily cycle of the surface temperature

Definition at line 62 of file maic2_variables.F90.

5.5.2.45 real(dp),dimension(100) maic2_variables::time_output

time_output(n): Specified times for data output

Definition at line 136 of file maic2_variables.F90.

5.5.2.46 real(dp),dimension(0:lmax) maic2_variables::water

water(l): Water content in the atmosphere

Definition at line 68 of file maic2_variables.F90.

5.5.2.47 real(dp),dimension(0:lmax) maic2_variables::water_new

(.)_new: New value of quantity (.) computed during an integration step

Definition at line 70 of file maic2_variables.F90.

Chapter 6

Data Type Documentation

6.1 instemp::ins Type Reference

Surface temperatures.

Public Attributes

- real(dp), dimension(0:360,-90:90) t
- real(dp), dimension(-90:90) tam
- real(dp), dimension(-90:90) tmax

6.1.1 Detailed Description

Surface temperatures.

Definition at line 46 of file instemp.f90.

6.1.2 Member Data Documentation

6.1.2.1 real(dp),dimension(0:360,-90:90) instemp::ins::t

Definition at line 47 of file instemp.f90.

6.1.2.2 real(dp),dimension(-90:90) instemp::ins::tam

Definition at line 48 of file instemp.f90.

6.1.2.3 real(dp),dimension(-90:90) instemp::ins::tmax

Definition at line 49 of file instemp.f90.

The documentation for this type was generated from the following file:

- subroutines/instemp.f90

Chapter 7

File Documentation

7.1 maic2.F90 File Reference

Main program file of MAIC-2.

```
#include "maic2_specs.h"
#include "subroutines/maic2_types.F90"
```

Functions/Subroutines

- program maic2

  Main program of MAIC-2.

7.1.1 Detailed Description

Main program file of MAIC-2.

7.1.2 Copyright

7.1.3 License

Definition in file maic2.F90.

7.1.4 Function Documentation

7.1.4.1 program maic2 ()

Main program of MAIC-2.

Definition at line 104 of file maic2.F90.

7.2 subroutines/boundary_maic2.F90 File Reference

Determination of the surface temperature and of the net mass balance (accumulation-ablation rate) for the polar caps of Mars.

Functions/Subroutines

- subroutine boundary_maic2 (time, dtime)

  Determination of the surface temperature and of the net mass balance (accumulation-ablation rate) for the polar caps of Mars.

7.2.1 Detailed Description

Determination of the surface temperature and of the net mass balance (accumulation-ablation rate) for the polar caps of Mars.

7.2.2 Copyright

7.2.3 License

Definition in file boundary_maic2.F90.

7.2.4 Function Documentation

7.2.4.1 subroutine boundary_maic2 (real(dp),intent(in) *time*, real(dp),intent(in) *dtime*)

Determination of the surface temperature and of the net mass balance (accumulation-ablation rate) for the polar caps of Mars.

Definition at line 37 of file boundary_maic2.F90.

7.3 subroutines/calc_top_maic2.F90 File Reference

Computation of the ice-cap topography.

Functions/Subroutines

- subroutine calc_top_maic2 (time, dtime)

  Computation of the ice-cap topography.

7.3.1 Detailed Description

Computation of the ice-cap topography.

7.3.2 Copyright

7.3.3 License

Definition in file calc_top_maic2.F90.

7.3.4 Function Documentation

7.3.4.1 subroutine calc_top_maic2 (real(dp),intent(in) *time*, real(dp),intent(in) *dtime*)

Computation of the ice-cap topography.

Definition at line 35 of file calc_top_maic2.F90.

7.4 subroutines/condensation.f90 File Reference

Computation of the condensation rate.

Modules

- module condensation

    Computation of the condensation rate.

Functions/Subroutines

- subroutine condensation::setcondpar (gravity, timescale)

    Setting of parameters.

- subroutine condensation::getcond_1 (temp, water, cond, dtime)

    Computation of condensation (removal of water exceeding the saturation pressure at the surface).

- subroutine condensation::getcond_2 (temp, water, cond)

    Computation of condensation (continuous, quadratic dependence on humidity).

Variables

- real(dp) condensation::g
- real(dp) condensation::tau

7.4.1 Detailed Description

Computation of the condensation rate.

7.4.2 Copyright

7.4.3 License

Definition in file condensation.f90.

7.5 subroutines/diff_trans.F90 File Reference

Diffusive transport of water in the Martian atmosphere.

Functions/Subroutines

- subroutine diff_trans (dtime)

  Diffusive transport of water in the Martian atmosphere.

7.5.1 Detailed Description

Diffusive transport of water in the Martian atmosphere.

7.5.2 Copyright

7.5.3 License

Definition in file diff_trans.F90.

7.5.4 Function Documentation

7.5.4.1 subroutine diff_trans (real(dp),intent(in) *dtime*)

Diffusive transport of water in the Martian atmosphere.

Definition at line 35 of file diff_trans.F90.

7.6 subroutines/evaporation.f90 File Reference

Computation of the evaporation rate (buoyant-diffusion approach by Ingersoll).

Modules

- module evaporation

  Computation of the evaporation rate (buoyant-diffusion approach by Ingersoll).

Functions/Subroutines

- subroutine evaporation::setevappar_bd (evap_fact_para, gamma_reg_para, R_-univ_para, mol_w_para, mol_c_para, diff_w_c_para, visc_c_para, g_para)

  Setting of parameters.

- subroutine evaporation::getevap_bd (temp, temp_amp, p, H, evap)

  Computation of evaporation.

Variables

- real(dp) evaporation::evap_fact
- real(dp) evaporation::gamma_reg
- real(dp) evaporation::R_univ
- real(dp) evaporation::mol_w
- real(dp) evaporation::mol_c
- real(dp) evaporation::diff_w_c
- real(dp) evaporation::visc_c
- real(dp) evaporation::g

7.6.1 Detailed Description

Computation of the evaporation rate (buoyant-diffusion approach by Ingersoll).

7.6.2 Copyright

7.6.3 License

Definition in file evaporation.f90.

7.7 subroutines/get_orb_par.F90 File Reference

Determination of the orbital parameters (eccentricity, obliquity, climate parameter CP, anomaly of vernal equinox, mean-annual north- or south-polar insolation).

Functions/Subroutines

- subroutine get_orb_par (time, ecc, obl, cp, ave, insol_ma_90NS)

  Determination of the orbital parameters (eccentricity, obliquity, climate parameter CP, anomaly of vernal equinox, mean-annual north- or south-polar insolation).

7.7.1 Detailed Description

Determination of the orbital parameters (eccentricity, obliquity, climate parameter CP, anomaly of vernal equinox, mean-annual north- or south-polar insolation).

7.7.2 Copyright

7.7.3 License

Definition in file get_orb_par.F90.

7.7.4 Function Documentation

7.7.4.1 subroutine get_orb_par (real(dp),intent(in) *time*, real(dp),intent(out) *ecc*, real(dp),intent(out) *obl*, real(dp),intent(out) *cp*, real(dp),intent(out) *ave*, real(dp),intent(out) *insol_ma_90NS*)

Determination of the orbital parameters (eccentricity, obliquity, climate parameter CP, anomaly of vernal equinox, mean-annual north- or south-polar insolation).

Definition at line 39 of file get_orb_par.F90.

7.8 subroutines/get_psi_tab.F90 File Reference

Computation of the table of true anomalies (orbital positions with respect to perihelion) over a Martian year.

Functions/Subroutines

- subroutine get_psi_tab (ecc, ave)

  Computation of the table of true anomalies (orbital positions with respect to perihelion) over a Martian year.

7.8.1 Detailed Description

Computation of the table of true anomalies (orbital positions with respect to perihelion) over a Martian year.

7.8.2 Copyright

7.8.3 License

Definition in file get_psi_tab.F90.

7.8.4 Function Documentation

7.8.4.1 subroutine get_psi_tab (real(dp),intent(in) *ecc*, real(dp),intent(in) *ave*)

Computation of the table of true anomalies (orbital positions with respect to perihelion) over a Martian year.

Definition at line 37 of file get_psi_tab.F90.

7.9 subroutines/instemp.f90 File Reference

Computation of the daily mean surface temperature of Mars based on obliquity, eccentricity and the anomaly of vernal equinox (local insolation temperature = LIS scheme).

Data Types

- type instemp::ins

  Surface temperatures.

Modules

- module instemp

  Computation of the daily mean surface temperature of Mars based on obliquity, eccentricity and the anomaly of vernal equinox (local insolation temperature = LIS scheme).

Functions/Subroutines

- subroutine instemp::setinstemp (o, ecc, ave, obl, sma, sa, sac, op, ct)

  Main subroutine of module instemp.

- real(dp) instemp::instam (o, phi)

  Annual mean temperature at latitude phi.

- real(dp) instemp::instmax (o, phi)

  Annual maximum temperature at latitude phi.

- real(dp) instemp::inst (o, psi, phi)

  Temperature at orbit position psi and latitude phi.

- real(dp) instemp::inst1 (o, ipsi, phi)

  Temperature at orbit position ipsi (integer value) and latitude phi.

7.9.1 Detailed Description

Computation of the daily mean surface temperature of Mars based on obliquity, eccentricity and the anomaly of vernal equinox (local insolation temperature = LIS scheme).

7.9.2 Copyright

Copyright 2010, 2011 Ralf Greve, Bjoern Grieger, Oliver J. Stenzel

7.9.3 License

Definition in file instemp.f90.

7.10 subroutines/maic2_types.F90 File Reference

Declarations of kind types for MAIC-2.

Modules

- module maic2_types

    Declarations of kind types for MAIC-2.

Variables

- integer, parameter maic2_types::i2b = selected_int_kind(4)

    2-byte integers

- integer, parameter maic2_types::i4b = selected_int_kind(9)

    4-byte integers

- integer, parameter maic2_types::sp = kind(1.0)

    single-precision reals

- integer, parameter maic2_types::dp = kind(1.0d0)

    double-precision reals

7.10.1 Detailed Description

Declarations of kind types for MAIC-2.

7.10.2 Copyright

7.10.3 License

You should have received a copy of the GNU General Public License along with MAIC-2. If not, see <http://www.gnu.org/licenses/>.

Definition in file maic2_types.F90.

7.11 subroutines/maic2_variables.F90 File Reference

Declarations of global variables for MAIC-2.

Modules

- module maic2_variables

  Declarations of global variables for MAIC-2.

Variables

- real(dp), dimension(0:lmax) maic2_variables::phi_node

  phi_node(l): Latitude of grid point

- real(dp), dimension(0:lmax) maic2_variables::phi_cb1

  phi_cb1(l): Latitude of lower cell boundary

- real(dp), dimension(0:lmax) maic2_variables::phi_cb2

  phi_cb2(l): Latitude of upper cell boundary

- real(dp), dimension(0:lmax) maic2_variables::cos_phi_cb1

  cos_phi_cb1(l): Cosine of phi_cb1(l)

- real(dp), dimension(0:lmax) maic2_variables::cos_phi_cb2

  cos_phi_cb2(l): Cosine of phi_cb2(l)

- real(dp), dimension(0:lmax) maic2_variables::sin_phi_cb1

  sin_phi_cb1(l): Sine of phi_cb1(l)

- real(dp), dimension(0:lmax) maic2_variables::sin_phi_cb2

  sin_phi_cb2(l): Sine of phi_cb2(l)

- real(dp), dimension(lmax) maic2_variables::dphi

  dphi(l): Grid spacing

- real(dp), dimension(lmax) maic2_variables::dphi_inv

  dphi_inv(l): Inverse of the grid spacing

- real(dp), dimension(0:lmax) maic2_variables::temp_surf

  temp_surf(l): Daily mean surface temperature

- real(dp), dimension(0:lmax) maic2_variables::temp_surf_amp

  temp_surf_amp(l): Amplitude of the daily cycle of the surface temperature

- real(dp), dimension(0:lmax) maic2_variables::temp_co2

  temp_co2(l): CO2 condensation temperature

- real(dp), dimension(0:lmax) maic2_variables::p_surf

  p_surf(l): Surface pressure

- real(dp), dimension(0:lmax) maic2_variables::water

  water(l): Water content in the atmosphere

- real(dp), dimension(0:lmax) maic2_variables::water_new

  (.)_new: New value of quantity (.) computed during an integration step

- real(dp), dimension(0:lmax) maic2_variables::cond

  cond(l): Condensation rate

- real(dp), dimension(0:lmax) maic2_variables::evap

  evap(l): Evaporation rate

- real(dp), dimension(0:lmax) maic2_variables::a_net

  a_net(l): Net surface mass balance of water ice

- real(dp), dimension(0:lmax) maic2_variables::H

  H(l): Ice thickness.

- real(dp), dimension(0:lmax) maic2_variables::H_new

  (.)_new: New value of quantity (.) computed during an integration step

- real(dp) maic2_variables::RHO

  RHO: Density of ice-dust mixture.

- real(dp) maic2_variables::RHO_I

  RHO_I: Density of ice.

- real(dp) maic2_variables::RHO_W

  RHO_W: Density of pure water.

- real(dp) maic2_variables::G

  G: Acceleration due to gravity.

- real(dp) maic2_variables::R

  R: Radius of Mars.

- real(dp) maic2_variables::rho_inv

  rho_inv: Inverse of the density of ice-dust mixture

- integer(i4b) maic2_variables::insol_time_min *insol_time_min: Minimum time of the data values for the insolation etc.*

- integer(i4b) maic2_variables::insol_time_stp

  insol_time_stp: Time step of the data values for the insolation etc.

- integer(i4b) maic2_variables::insol_time_max

  insol_time_max: Maximum time of the data values for the insolation etc.

- integer(i4b) maic2_variables::itercount

  itercount: Counter for the number of time integration steps

- real(dp), dimension(0:100000) maic2_variables::insol_ma_90

  insol_ma_90(n): Data for the mean-annual north- or south-polar insolation

- real(dp), dimension(0:100000) maic2_variables::obl_data

  obl_data(n): Data for the obliquity

- real(dp), dimension(0:100000) maic2_variables::ecc_data

  ecc_data(n): Data for the eccentricity

- real(dp), dimension(0:100000) maic2_variables::ave_data

  ave_data(n): Data for the anomaly of vernal equinox (= 360 deg - Ls of perihelion)

- real(dp), dimension(0:100000) maic2_variables::cp_data

  cp_data(n): Data for Laskar's climate parameter = eccentricity ∗sin(Laskar's longitude of perihelion from moving equinox), (where Laskar's longitude of perihelion from moving equinox = Ls of perihelion - 180 deg)

- real(dp), dimension(0:ntime) maic2_variables::psi_tab

  psi_tab(n): True anomalies (orbital positions with respect to perihelion) over a Martian year

- real(dp), dimension(0:lmax) maic2_variables::diff_aux

  diff_aux(l): Auxiliary quantity needed for the diffusional transport

- integer(i4b) maic2_variables::n_output

  n_output: Number of specified times for data output

- real(dp) maic2_variables::dtime_out

  dtime_out: Time step for data output

- real(dp), dimension(100) maic2_variables::time_output

  time_output(n): Specified times for data output

- integer(i4b) maic2_variables::iter_out

  iter_out: Intervall of integration steps for data output

- integer(i4b), dimension(100) maic2_variables::iter_output
  iter_output(n): Specified integration steps for data output
- real(dp), parameter maic2_variables::pi = 3.141592653589793_dp
  pi: Mathematical constant
- real(dp), parameter maic2_variables::pi_inv = 1.0_dp/pi
  pi_inv: Inverse of pi
- real(dp), parameter maic2_variables::pi_180 = pi/180.0_dp
  pi_180: pi divided by 180 (-> deg to rad)
- real(dp), parameter maic2_variables::pi_180_inv = 180.0_dp/pi
  pi_180_inv: 180 divided by pi (-> rad to deg)
- real(dp), parameter maic2_variables::eps = 1.0e-05_dp
  eps: Small number

7.11.1 Detailed Description

Declarations of global variables for MAIC-2.

7.11.2 Copyright

7.11.3 License

Definition in file maic2_variables.F90.

7.12 subroutines/output.F90 File Reference

Data output.

Functions/Subroutines

- subroutine output (time)
  Data output.

7.12.1 Detailed Description

Data output.

7.12.2 Copyright

7.12.3 License

Definition in file output.F90.

7.12.4 Function Documentation

7.12.4.1 subroutine output (real(dp),intent(in) *time*)

Data output.

Definition at line 35 of file output.F90.

7.13 subroutines/p_sat.f90 File Reference

Computation of the water-vapour saturation pressure.

Functions/Subroutines

- real(dp) p_sat (temp_surf)

  Computation of the water-vapour saturation pressure.

7.13.1 Detailed Description

Computation of the water-vapour saturation pressure.

7.13.2 Copyright

7.13.3 License

Definition in file p_sat.f90.

7.13.4 Function Documentation

7.13.4.1 real(dp) p_sat (real(dp),intent(in) *temp_surf*)

Computation of the water-vapour saturation pressure.

Definition at line 35 of file p_sat.f90.

7.14 subroutines/tri_sle.F90 File Reference

Solution of a system of linear equations Ax=b with tridiagonal matrix A.

Functions/Subroutines

- subroutine tri_sle (a0, a1, a2, x, b, n_rows)

  Solution of a system of linear equations Ax=b with tridiagonal matrix A.

7.14.1 Detailed Description

Solution of a system of linear equations Ax=b with tridiagonal matrix A.

7.14.2 Copyright

Copyright 2010, 2011 Ralf Greve, Bjoern Grieger, Oliver J. Stenzel

7.14.3 License

This file is part of MAIC-2.

MAIC-2 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MAIC-2 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with MAIC-2. If not, see <`http://www.gnu.org/licenses/`>.

Definition in file tri_sle.F90.

7.14.4 Function Documentation

7.14.4.1 subroutine tri_sle (real(dp),dimension(0:∗),intent(inout) *a0*, real(dp),dimension(0:∗),intent(inout) *a1*, real(dp),dimension(0:∗),intent(inout) *a2*, real(dp),dimension(0:∗),intent(out) *x*, real(dp),dimension(0:∗),intent(inout) *b*, integer(i4b),intent(in) *n_rows*)

Solution of a system of linear equations Ax=b with tridiagonal matrix A.

Parameters

- ← ***a0*** a0(j) is element A_(j,j-1) of matrix A
- ← ***a1*** a1(j) is element A_(j,j) of matrix A
- ← ***a2*** a2(j) is element A_(j,j+1) of matrix A
- ← ***b*** inhomogeneity vector
- ← ***n_rows*** size of matrix A (indices run from 0 (!!!) to n_rows)
- → ***x*** solution vector.

Definition at line 41 of file tri_sle.F90.

Index