Supplement of Clim. Past, 21, 1699–1724, 2025
https://doi.org/10.5194/cp-21-1699-2025-supplement

Climate of the Past Open Access EGU

Supplement of

On the implementation of external forcings in a regional climate model – a sensitivity study around the Samalas volcanic eruption in the Eastern Mediterranean/Middle East

Eva Hartmann et al.

Correspondence to: Eva Hartmann (eva.hartmann@geogr.uni-giessen.de)

Code for the implementation of the forcings

To use the transient forcings several changes need to be done in the source code of CCLM. The changes are explained in the following:

- In *src_radiation.f90* between line 1384 and 1396: add integer *mm*
- In *src_radiation.f90* between line 206 and 675: add variables for forcings as shown in Figure S1
- In *src_radiation.f90* between line 1539 and 1546: add reals *aod (28812)*, *solc_tbl (28224)* and *solc*
- Create a new file in *src* which contains the values for the greenhouse gas, the orbital, the solar and the volcanic forcing. It is named *data_forcing.f90* and has the structure shown in Figure S2 and Figure S2 where the values according to the used dataset needs to be filled
- In *ObjDependencies* add the file *data_forcing.f90* to the section *Dependencies of the Data Modules*
- In *ObjFiles* add the file *data_forcing.o* to the section *DATAOBJ*

S1 Greenhouse Gases

The greenhouse gas forcing is added as a namelist option. Originally one can choose for the namelist parameter *ico2_rad* different CO_2 or effective CO_2 scenarios with the numbers 1 to 10. Here we add option 11. To use the transient greenhouse gas forcing the following changes need to be done in the namelist and the source code:

- Set CCLM namelist parameter *ico2_rad* to 11

```
USE data_forcing, ONLY: &
  aod_1,           & ! stratospheric aerosol optical depth
  aod_2,           & ! stratospheric aerosol optical depth
  aod_3,           & ! stratospheric aerosol optical depth
  solc_1,          & ! solar constant
  solc_2,          & ! solar constant
  solc_3,          & ! solar constant
  recc,            & ! eccentricity
  robld,           & ! obliquity
  rlonp,           & ! perihelion
  tco2,            & ! CO2
  tch4,            & ! CH4
  tn o               ! N2O
```

Figure S1. Code to be implemented in *src_radiation.f90* for the transient greenhouse gas forcing.

```
!+ Data module for all parametric data for the climate forcings
!-------------------------------------------------------------------------------

MODULE data_forcing

!-------------------------------------------------------------------------------
!
! Description:
!  This module declares and initializes all parametric scalar and array
!  data which are used for the climate forcings
!
! Current Code Owner: Eva Hartmann
!  email:  Eva.Hartmann@geogr.uni-giessen.de
!!
! Code Description:
! Language: Fortran 90.
! Software Standards: "European Standards for Writing and
! Documenting Exchangeable Fortran 90 Code".
!===============================================================================
!
! Modules used:
USE data_parameters, ONLY :   &
    ireals,    & ! KIND-type parameter for real variables
    iintegers    ! KIND-type parameter for standard integer variables

!===============================================================================

IMPLICIT NONE

!===============================================================================

! Global (i.e. public) Declarations:

! 1. Data arrays for properties of different forcings
! ---------------------------------------------------

  REAL  (KIND=ireals) ::  &
!   a) parameters for the volcanic forcing
    aod_1 (6000),   & !  stratospheric aerosol optical depth
    aod_2 (12000),  & !  stratospheric aerosol optical depth
    aod_3 (10812),  & !  stratospheric aerosol optical depth

!   b) parameters for the orbital forcing
    recc (3451),    & ! eccentricity
    robld (3451),   & ! obliquity
    rlonp (3451),   & ! longitude of perihelion

!   c) parameters for the solar forcing
    solc_1 (6000),   & ! solar constant
    solc_2 (12000),  & ! solar constant
    solc_3 (10224),  & ! solar constant

!   d) parameters for the greenhouse gases
    tco2 (2351),  & ! CO2 concentration
    tch  (2351),  & ! CH4 concentration
    tn o (2351)     ! N2O concentration

  ! Initialization of forcing values

! Volcanic Forcing data on monthly resolution for chosen Latitudes
DATA aod_1 / 0.0049, 0.0047, 0.0045, ... / ! monthly_data from 500 to 1 BCE

DATA aod_2 / ... / ! monthly_data from 0 to 999 CE

DATA aod_3 / ... / ! monthly_data from 1000 to 1900 CE

! Solar Forcing data on monthly resolution
DATA solc_1 / 1360.7570, 1360.7861, ... / ! monthly_data from 500 to 1 BCE

DATA solc_2 / ... / ! monthly_data from 0 to 999 CE

DATA solc_3 / ... / ! monthly_data from 1000 to 1851 CE

```

Figure S2. Code of the new created file *data_forcing.f90* for the transient forcing values - part I.

```
! Orbital Forcing data on yearly resolution
! (Eccentricity, Obliquity, Longitude of Perihelion)
DATA recc / 0.017640, 0.017640, ... / ! yearly data from 500 BCE to 2950 CE

DATA robld / 23.754, 23.754, ... / ! yearly data from 500 BCE to 2950 CE

DATA rlonp / 240.46, 240.48, ... / ! yearly data from 500 BCE to 2950 CE

! Greenhouse Gas Concentration on yearly resolution
! (CO2, CH4, N2O)
DATA tco2 / 2.76197e-04, 2.76199e-04, ... / ! yearly data from 500 BCE to 1850 CE

DATA tch4 / 6.1082e-07, 6.1085e-07, ... / ! yearly data from 500 BCE to 1850 CE

DATA tn2o / 2.6911e-07, 2.6910e-07, .../ ! yearly data from 500 BCE to 1850 CE

!=======================================================================

END MODULE data_forcing

```

Figure S3. Code of the new created file *data_forcing.f90* for the transient forcing values - part II.

```
      CASE (11)
        ! specific transient effective CO2
        ! 25 * ch4 and 298 * n2o
        zqco2  = ( tco2(jj-6999+500) &
                   + tch4(jj-6999+500)*25 &
                   + tn2o(jj-6999+500)*298 ) * 1.519_ireals
```

Figure S4. Code to be implemented in *src_radiation.f90* for the transient greenhouse gas forcing.

- In *organize_physics.f90* line 2673: change 10 to 11
- In *src_radiation.f90* after line 1870: add case 11 as shown in Figure S4

S2 Orbital Forcing

The orbital forcing is added as three values on yearly resolution. They are the eccentricity, the obliquity and the longitude of perihelion of the earth's orbit. The original code is owned by Patrick Ludwig and was modified for the transient simulation. To use the transient orbital forcing the following changes need to be done in the source code:

- In *src_radiation.f90* from line 2042 to 2058: comment out and replace by:
- In *src_radiation.f90* after line 2041: add orbital forcing as shown in Figure S5 and Figure S6

S3 Solar Forcing

The solar forcing is added as monthly value for the solar constant. To use the transient solar forcing the following changes need to be done in the source code:

```
if (itype_calendar==0) then
    yearl = 365 + IABS( MOD(jj,4) - 4) / 4
 else
    yearl = 360
 end if

 api     = 2.*ASIN(1.)                       ! Pi
 zve 0 = 80.5                                ! Day of the vernal equinox in 1900
 zvebas = yearl-zve00                        ! Remainder of the year
 zyearve = float(itaja) - 1. + zvebas        ! itaja: actual day of the year
                                             ! zyearve: time of the year from
                                             ! last years vernal equinox in degrees
                                             ! zvetim = MOD(zyearve/yearl,1.)*2.*api
                                             ! Time of the year from the vernal equinox
                                             ! in radians
 zyeardif=zyearve/yearl
 zone=1.
 zvetim = MOD(zyeardif,zone)*2.*api

 zecc=recc(jj-6999+500)
 zobld=robld(jj-6999+500)
 zlonp=rlonp(jj-6999+500)

 zoblr=zobld*api/180.
 zlonpr=zlonp*api/180.
 zsqecc=SQRT((1+zecc)/(1-zecc))

 zeps=1.E-6

 ! CALCULATION OF ECCENTRIC ANOMALY OF VERNAL EQUINOX!
 zeve=2.*ATAN(TAN(0.5*zlonpr)/zsqecc)

 ! CALCULATION OF TIME ANGLE IN RADIANS OF LONGITUDE OF PERIHELION
 ! FROM VERNAL EQUINOX
 ztlonpr= zeve - zecc*SIN(zeve)

 ! CALCULATE ECCENTRIC ANOMALY:
 ! USE FIRST DERIVATIVE (NEWTON) TO CALCULATE SOLUTION FOR
 ! EQUATION OF ECCENTRIC ANOMALY *ZENEW*
 zztime=zvetim-ztlonpr

 zeold=zztime/(1.-zecc)
 zenew=zztime
 iter=0

 DO
    zzeps=zeold-zenew
    IF (iter.GE.10) THEN
       yzerrmsg = 'PALEO ORBIT:  - eccentric anomaly not found!'
       CALL model_abort (my_world_id, 2091, yzerrmsg, 'radiation')
    END IF
    IF (ABS(zzeps).LT.zeps) EXIT
    iter=iter+1
    zeold=zenew
    zcose=COS(zenew)
    zenew=(zztime+zecc*(SIN(zenew)-zenew*zcose))/(1.-zecc*zcose)
 END DO

 zsocof=(1./(1.-zecc*COS(zenew)))**2

 ! CALCULATION OF THE DECLINATION.
 ztgean=TAN(zenew*0.5)

 ! *znu*: TRUE ANOMALY
 ! (ACTUAL ANGLE OF EARTH'S POSITION FROM PERIHELION)
 ! *zlambda*: TRUE LONGITUDE OF THE EARTH
 ! (ACTUAL ANGLE FROM VERNAL EQUINOX)
 znu=2.*ATAN(zsqecc*ztgean)
 zlambda=znu+zlonpr
 zsinde=SIN(zoblr)*SIN(zlambda)
 zdek=ASIN(zsinde)

 zdeksin_save = SIN (zdek)
```

Figure S5. Code to be implemented in *src_radiation.f90* for the transient orbital forcing - part I.

```
74    zdekcos_save = COS (zdek)
75
76    zsct_save = zsocof*solc
77
78    zsct_h = zsct_h + zsct_save
79    nz_zsct = nz_zsct + 1
80
81    ztwo     = 0.681 + 0.2422*(jj-1949)-(jj-1949)/4
82    ztho     = 2.*pi*( REAL(itaja, ireals) -1.0 + ztwo )/365.2422
83    zdtzgl_save  = 0.000075 + 0.001868*COS(   ztho) - 0.032077*SIN(   ztho) &
84         - 0.014615*COS(2.*ztho) - 0.040849*SIN(2.*ztho)
```

Figure S6. Code to be implemented in *src_radiation.f90* for the transient orbital forcing - part II.

```
1          solc_tbl(1:6000) = solc_1
2          solc_tbl(6001:18000) = solc_2
3          solc_tbl(18001:28224) = solc_3
```

Figure S7. Code to be implemented in *src_radiation.f90* for the transient solar forcing - part I.

- In *src_radiation.f90* line 289: comment out old solar constant
- In *src_radiation.f90* after line 2033 and after line 3623: add solar forcing values as tables shown in Figure S7
- In *src_radiation.f90* after line 2036 and after line 3628: add how to get the solar forcing values for the actual month as shown in Figure S8

S4 Volcanic Forcing

The volcanic forcing is added as monthly value for the stratospheric aerosol optical depth. The raw data is given in latitudinal bands. As preparation, the latitudes covering the domain of the RCM simulations are selected and we continue with an average of those latitudes. To use the transient volcanic forcing the following changes need to be done in the source code:

- In *src_radiation.f90* after line 1887: add volcanic forcing values as tables shown in Figure S9
- In *src_radiation.f90* line 1890 and line 7991: add variables *aod*, *ntstep* and *ydate_ini* to the function *aerdis*
- In *src_radiation.f90* after line 7884: add additional local variables for the calculation of the volcanic forcing as shown in Figure S10

```
1          READ (yrad1(5:6),'(I2)') mm
2          solc = solc_tbl((jj-7000+500)*12 + mm)
```

Figure S8. Code to be implemented in *src_radiation.f90* for the transient solar forcing - part II.

```
      aod(1:6000) = aod_1
      aod(6001:18000) = aod_2
      aod(18001:28812) = aod_3
```

Figure S9. Code to be implemented in *src_radiation.f90* for the transient volcanic forcing - part I.

```
    REAL  (KIND=ireals), DIMENSION(28812), INTENT(IN) ::  &
      aod                  ! aerosol optical depth of the volcanic forcing file
    INTEGER (KIND=iintegers), INTENT(IN)  ::  &
      ntstep
    INTEGER  (KIND=iintegers)  ::  &
      itaja,                  & !
      ist,                    & ! number of aod values
      mm,                     & ! output from routine
      jj                        ! output from routine
    REAL    (KIND=ireals   ) ::  &
      zstunde                   ! output from routine get_utc_dat
    CHARACTER (LEN=14)   yrad1          ! output from routine get_utc_dat
    CHARACTER (LEN=28)   yrad2          ! output from routine get_utc_dat

    CHARACTER (LEN=14), INTENT(IN) ::  ydate_ini   ! start of the forecast yyyymmddhh
```

Figure S10. Code to be implemented in *src_radiation.f90* for the transient volcanic forcing - part II.

- In *src_radiation.f90* line 7912: comment out the calculation of *pstbga* and replace by:
- In *src_radiation.f90* after line 7912: add additional calculation of the volcanic forcing as shown in Figure S11

S5 Land-Use Change

To make the land-use change transient in the CCLM simulation one has to convert the ESM output to the needs of CCLM with an appropriate converter for external data. It could be created a new external data file for each month, each year or whatever is needed. This can be read in by the INT2LM routine during the simulation run.

```
      CALL get_utc_date ( ntstep, ydate_ini, dt, itype_calendar, yrad1, yrad2,  &
             itaja, zstunde )

      READ (yrad1(1:4),'(I4)') jj
      READ (yrad1(5:6),'(I2)') mm

      ist = ((jj-7000 + 500) * 12) + mm

      pstbga = aod(ist) / 19330.0_ireals
```

Figure S11. Code to be implemented in *src_radiation.f90* for the transient volcanic forcing - part III.