Namelist driven, tailor-made on-line diagnostic and output control for COSMO-CLM using the Modular Earth System (MESSy)

Astrid Kerkweg[1], Mariano Mertens[2], Patrick Jöckel[2] and Christiane Hofmann[1]

[1] Meteorological Institute, University Bonn
[2] Institute for Atmospheric Physics DLR, Oberpfaffenhofen

CLM-Assembly 2018, Karlsruhe

Summary

On-line diagnostics are important as amount of model data increases strongly with increasing resolution

MESSy allows:

1) on-line statistics of all model variables
2) additional diagnostic
   - on-line sampling of model data
     - along specific paths or satellite orbits
     - at specific iso-surfaces
   - on-line diagnostics (e.g. tropopause height)

These diagnostic MESSy features are now available to the CLM-Community

Outline

- On-line diagnostic in MESSy ?
- Example: CHANNEL
  - on-line statistic, object attributes
  - tailor-made output files (redirection of objects)
  - aveout
- Examples: Diagnostic Submodels
  - diagnostic of tropopause and boundary layer height
  - output at specified locations
  - output on iso-surfaces
  - algebraic combination of fields

- How to get, use and extend COSMO-CLM/MESSy?

Outline

- **On-line diagnostic in MESSy ?**
- Example: CHANNEL
  - on-line statistic, object attributes
  - tailor-made output files (redirection of objects)
  - aveout
- Examples: Diagnostic Submodels
  - diagnostic of tropopause and boundary layer height
  - output at specified locations
  - output on iso-surfaces
  - algebraic combination of fields

- How to get, use and extend COSMO-CLM/MESSy?

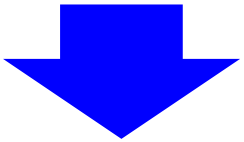
The Modular Earth Submodel System (MESSy)

… is a software providing a framework for standardised bottom-up implementation of Earth System Models (or parts of those) with flexible complexity…

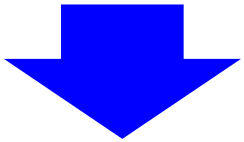
(i.e., using MESSy does not necessarily include “chemistry” or “aerosols”)

What are the design concepts of MESSy?

strict separation of
model-infrastructure[1)] from process-description



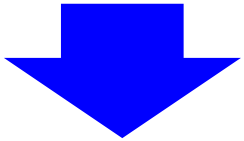
„generic“
submodels



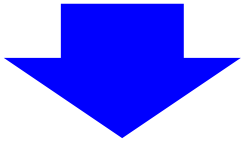
process- /
diagnostic
submodels

What are the design concepts of MESSy?

strict separation of
model-infrastructure[1)] from process-description



„generic“
submodels



process- /
diagnostic
submodels

[1)] **model infrastructure** =
memory management,
input / output,
flow control,
grid-transformations, ...

What are the design concepts of MESSy?

strict separation of
model-infrastructure[1)] from process-description

„generic“
submodels

process- /
diagnostic
submodels

[1)] **model infrastructure** =
memory management,
input / output,
flow control,
grid-transformations, ...

CHANNEL

CHANNEL: Internal statistics

For all channel objects various statistics are available

Instantaneous (i.e. value at output time step)
Average over output interval (different intervals possible)
Standard deviation
Minimum
Maximum
Event counts
Event averages

For every object multiple statistics (with different integration periods) are possible (namelist controlled)

All statistics can either be written in one or multiple different files

What are the design concepts of MESSy?

strict separation of
model-infrastructure[1)] from process-description

„generic“
submodels

process- /
diagnostic
submodels

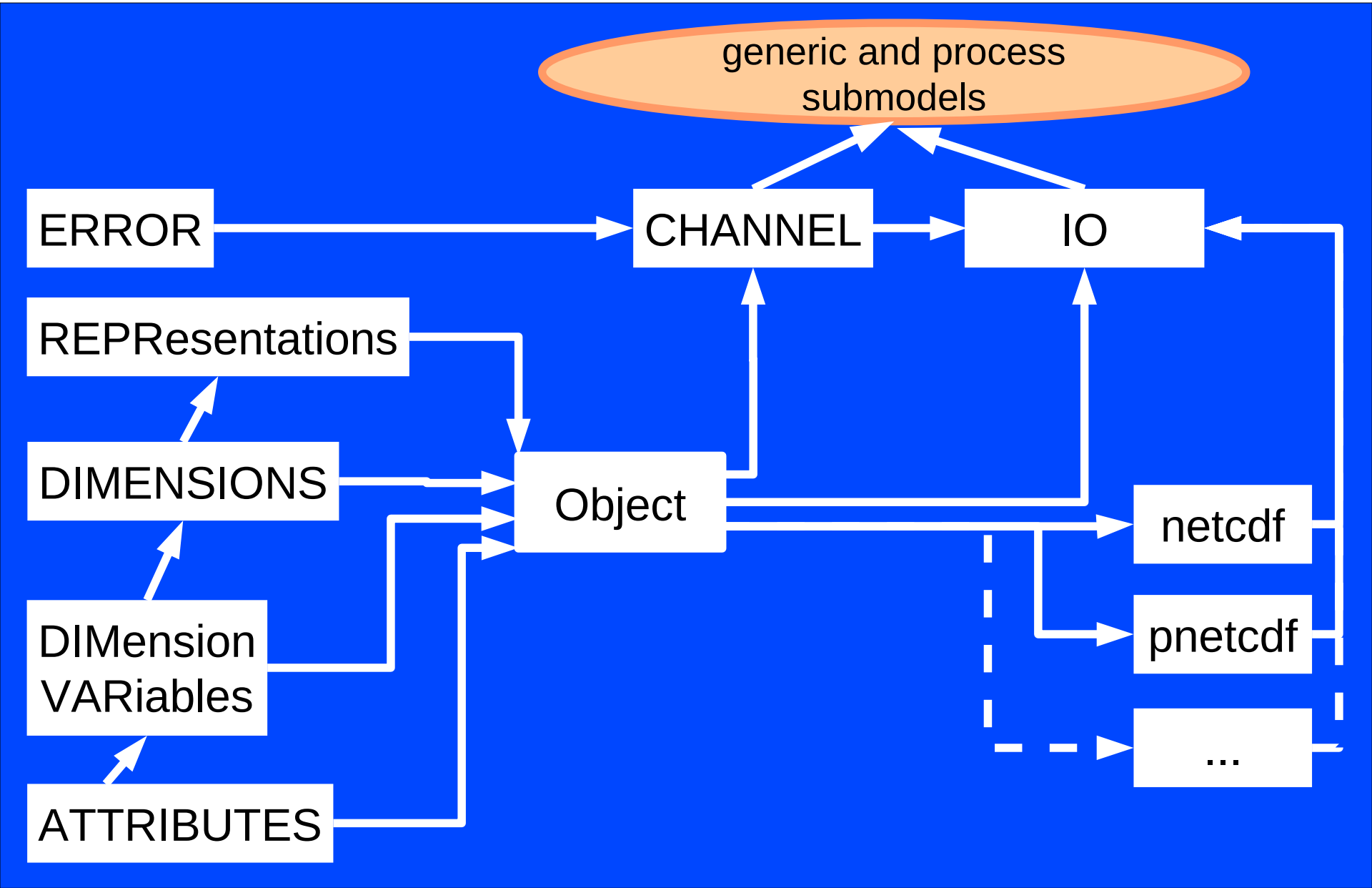
[1)] **model infrastructure** =
memory management,
input / output,
flow control,
grid-transformations, ...

diagnostic submodels:
TROPOP,
S4D, SCOUT, SORBIT,
PTRAC, TREXP,
VAXTRA, VISO

Outline

- On-line diagnostic in MESSy ?
- **Example: CHANNEL**
  - on-line statistic, object attributes,
  - tailor-made output files (redirection of objects)
  - aveout
- Examples: Diagnostic Submodels
  - diagnostic of tropopause and boundary layer height
  - output at specified locations
  - output on iso-surfaces
  - algebraic combination of fields
- How to get, use and extend COSMO-CLM/MESSy?

CHANNEL


generic and process submodels
ERROR
CHANNEL
IO
REPResentations
DIMENSIONS
Object
netcdf
pnetcdf
...
DIMension VARiables
ATTRIBUTES

CHANNEL

CHANNEL …

- … is written in Fortran95 and
- … provides the following basic entities:
  - attributes — time independent, scalar characteristic (e.g., unit)
  - dimension variables — specific coordinate axis (e.g., latitude in degree N)
  - dimensions — basic geometry in one dimension
  - representations — describing multidimensional geometric structures (based on dimensions), e.g. Eulerian (or gridpoint), spectral, Lagrangian
  - channel objects — data fields including meta information (attributes) and underlying geometric structure (respresentation)
  - channels — set of related channel objects

CHANNEL

CHANNEL …

- … is written in Fortran95 and
- … provides the following basic entities:
  - attributes — time independent, scalar characteristic (e.g., unit)
  - dimension variables — specific coordinate axis (e.g., latitude in degree N)
  - dimensions — basic geometry in one dimension
  - representations — describing multidimensional geometric structures (based on dimensions), e.g. Eulerian (or gridpoint), spectral, Lagrangian
  - channel objects — data fields including meta information (attributes) and underlying geometric structure (respresentation)
  - channels — set of related channel objects

Each channel produces an individual output file, if the output of the respective channel is requested in the channel namelist

CHANNEL: namelist control

The output is controlled via namelists

- For output control the channel can be controlled individually and single objects (e.g. T_2M) can be written in multiple channels (e.g. with different integration periods or statistics)

```
&CTRL
...
ADD_CHANNEL(1)= 'test',
ADD_REF(1)    = 'COSMO_ORI', 'fr_land', '*', '',
ADD_REF(2)    = 'COSMO',     'um1', 'test', 'u_wind'
...
/
```

- Determination of output interval individually for each channel

```
&CPL
...
TIMER_DEFAULT     = '',       2, 'hours','first', 0,
TIMER_CHANNEL( 1) = 'test',   1, 'steps','first', 0,
TIMER_CHANNEL( 2) = 'scout', 1, 'hours','first', 0,
...
/
```

CHANNEL: tailor-made output and on-line statistic

```
&CTRL
...
! ------------------------------------------------------------------
! # SET DEFAULT OUTPUT AND RESTART HANDLING
!       - ''
!       - OUTPUT-FILETYPE, RERUN-FILETYPE, NO. OF STEPS PER OUTPUT-FILE,
!       - RERUN, IGNORE, INST, AVE, STD, MIN, MAX, CNT, CAV, RANGE(2)
! # NOTES:
! #     IGNORE = T  FORCES THE MODEL TO IGNORE POTENTIAL lrestreq FLAGS
! #                 FOR PSEUDO-INITIAL RUNS FROM RESTART FILES
! ------------------------------------------------------------------
OUT_DEFAULT     = '', 2, 2, -1, F,F, T,F,F,F,F, F,F, , ,
! ------------------------------------------------------------------
...
OUT_CHANNEL( 11) = 'test',        2, 2, -1, F,F, T,F,F,F,F, F,F, , ,
...

OUT_OBJECT( 10)  = 'COSMO_ORI','fr_land', F,F, F,F,F,F,F, F,F, , ,
...
/
```

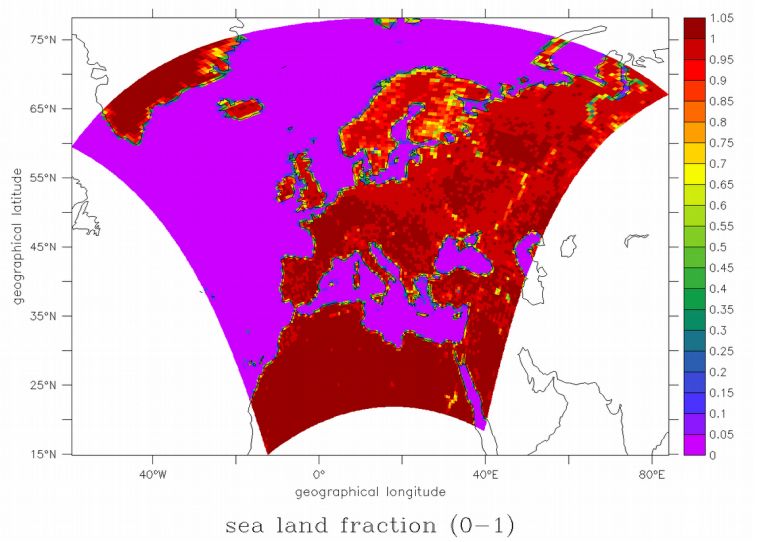
What is the difference between on-line and off-line calculated averages?

On-line statistic is calculated from instantaneous values

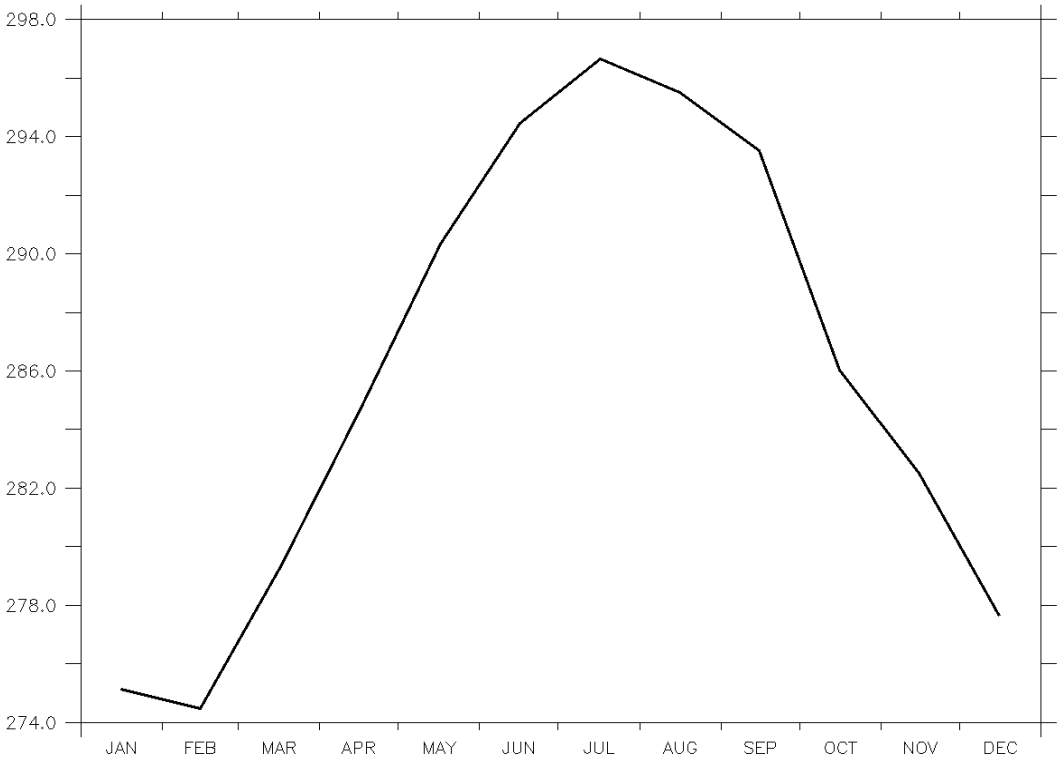
Simulation with EVAL setup (1979-2002)

Temperature averages calculated from instantaneous and x-hourly (x=1,2,3,4,5,6,7,9) values

Example:
temperature in
- lowest model layer,
- area average over land
- average over 1979-2002



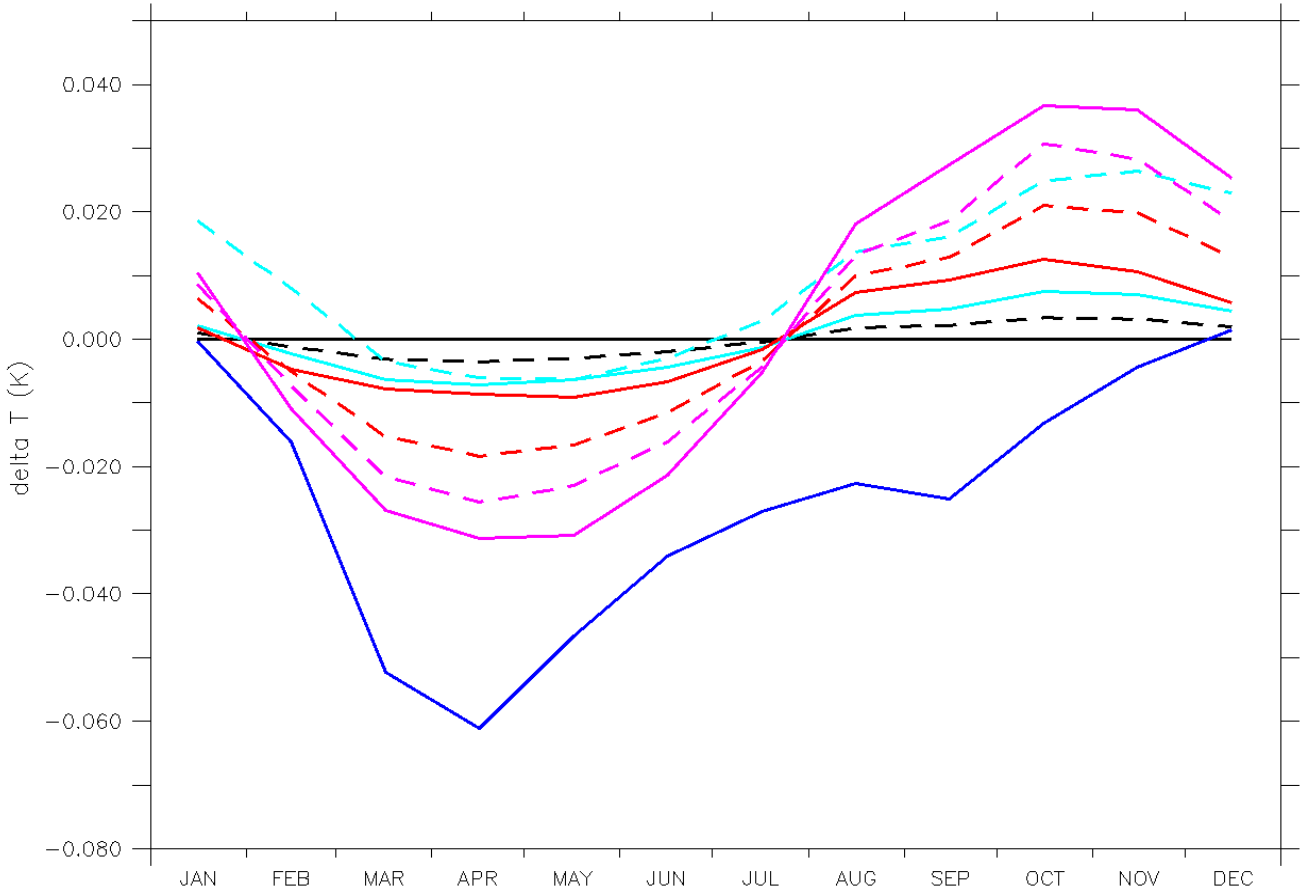
What is the difference between on-line and off-line calculated averages?

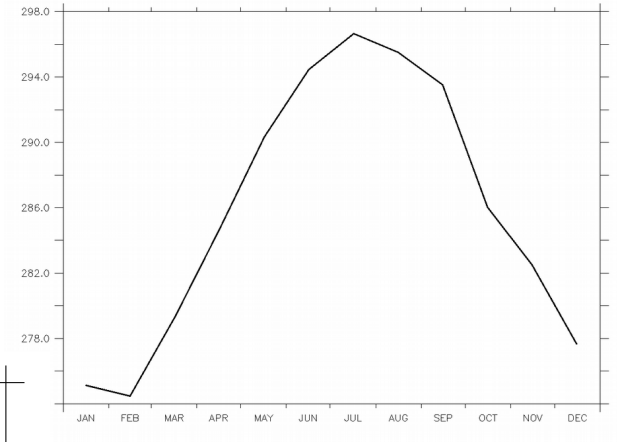


average temperature (K)

What is the difference between on-line and off-line calculated averages?

$\overline{T}_x - \overline{T}_i$ x =1,2,3,4,5,6,7,9 (K)





$\overline{T}_i$ (K)

UNIVERSITÄT BONN

CHANNEL: Other features

- COSMO-CLM/MESSy is equipped with check-pointing facility restarting the simulation automatically if a given wallclock time is (almost) reached
    - Restart files are written as double precision netcdf-files

- All important COSMO fields are already available as CHANNEL objects (e.g. all types of diagnostics and statistics are possible with these fields)

- Original COSMO output is still available (e.g. for use with eTool)

Outline

- On-line diagnostic in MESSy ?
- Example: CHANNEL
  - on-line statistic, object attributes,
  - tailor-made output files (redirection of objects)
  - aveout
- **Examples: Diagnostic Submodels**
  - diagnostic of tropopause and boundary layer height
  - output at specified locations
  - output on iso-surfaces
  - algebraic combination of fields
- How to get, use and extend COSMO-CLM/MESSy?

Diagnostic MESSy submodels

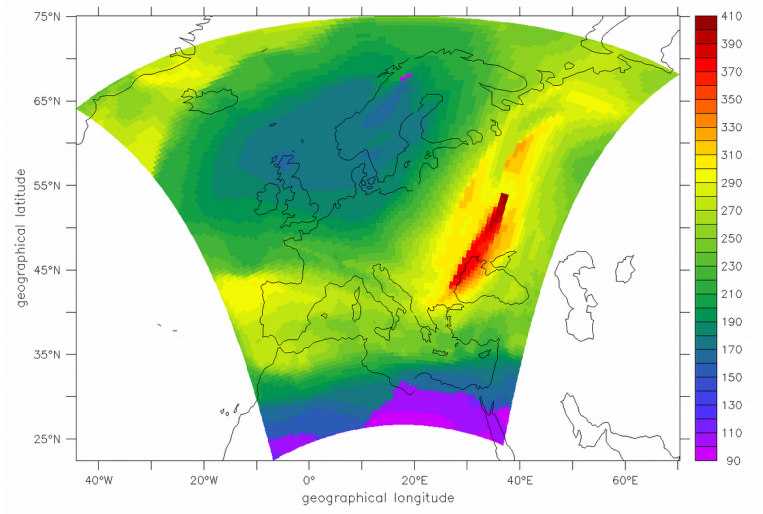
Using additional MESSy submodels various diagnostics are possible

- Interpolation on different vertical grids (e.g. pressure or at special height levels) (VAXTRA)
- output of one variable on the ISO-surface of another variable (VISO)
- Online diagnostics of PBLH/Tropopause height using different definitions (TROPOP)
- Output at given lat/lon points (SCOUT) or along different trajectories, e.g. flight- or satellite tracks (S4D or SORBIT)
- definition of prognostic tracers (PTRAC)
- initialisation of prognostic tracers (PTRACINI)
- emission at a specific location and exponential decay of a prognostic tracer (TREXP)

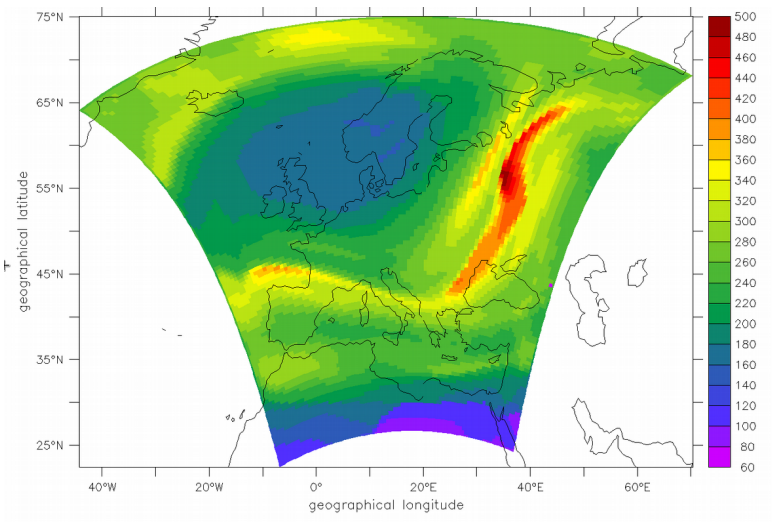
Taylor made additional diagnostics can easily be implemented using own MESSy submoduls

Diagnostic submodels: TROPOP

Different diagnostics for sea level pressure, tropopause height (pressure) or boundary layer height available



WMO definition

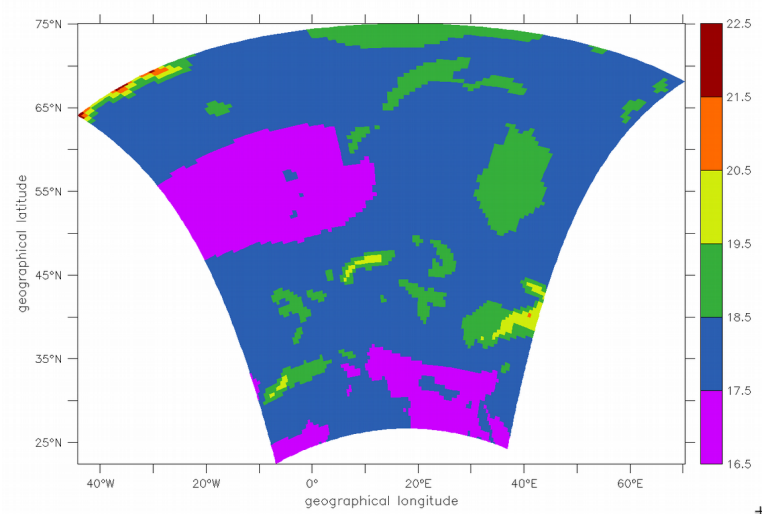


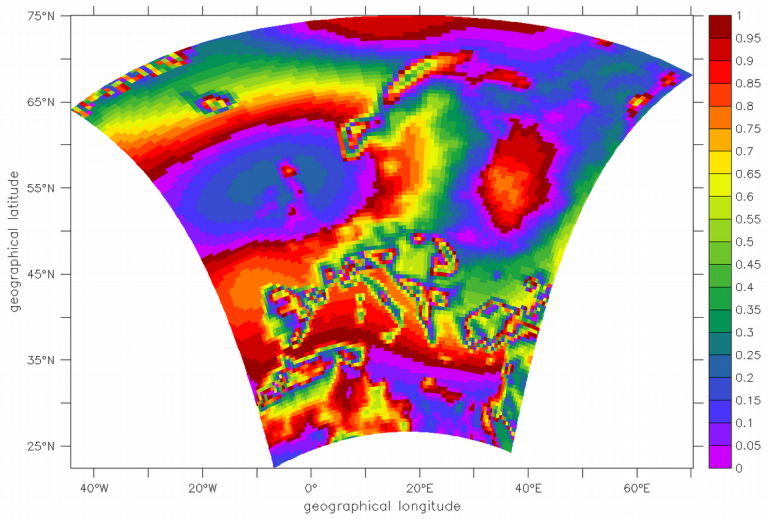
PV definition

Tropopause Pressure (hPa)

Diagnostic submodels: VISO (1)

Diagnostic of vertically layered, 2-D iso-surfaces in 3-D scalar fields





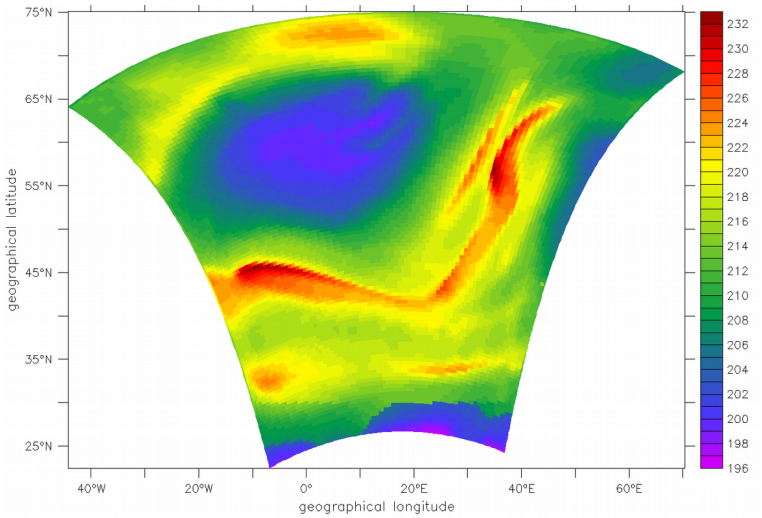
Level index ... *Level fraction...*

... for 500 hPa pressure iso-surface

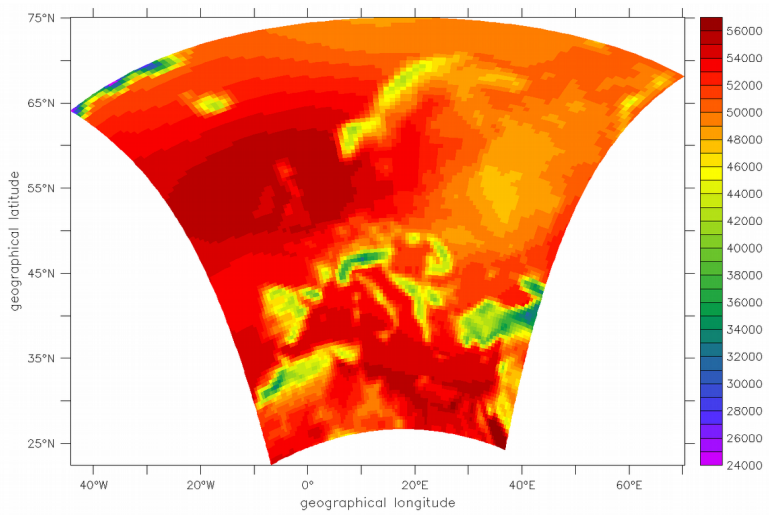
```
&CPL
ISO(5)  = 'p500','COSMO','press', 50000.0, T, T,  ,  ,
/
```

Diagnostic submodels: VISO (2)

Mapping of 3-D scalar fields in grid-point representation on iso-surfaces



Temperature at tropopause (K)



Geopotential (m^2s^{-2}) in 500 hPa

```
&CPL
MAP(16) = 'ttp',     'tropop', 'tp',    'COSMO', 'tm1',
MAP(33) = 'g500',    'viso',   'p500',  'COSMO', 'geopot',
/
```

Diagnostic submodels: VAXTRA

VAXTRA (Vertical AXes TRAnsformations) can be used to transform 3D grid-point variables from the native vertical axis on-line to arbitrary (specified via namelist) pressure, height or potential temperature vertical axes for output via CHANNEL.

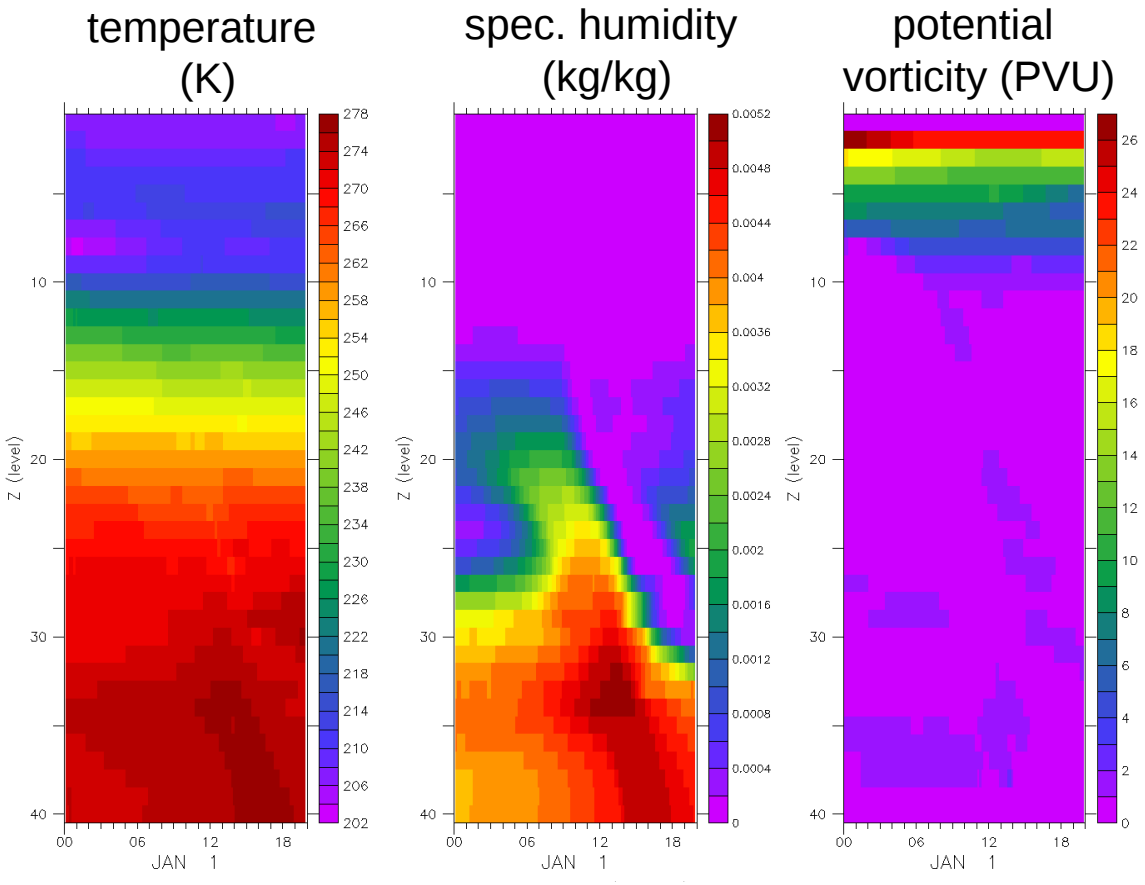
Diagnostic submodels: VAXTRA

VAXTRA (Vertical AXes TRAnsformations) can be used to transform 3D grid-point variables from the native vertical axis on-line to arbitrary (specified via namelist) pressure, height or potential temperature vertical axes for output via CHANNEL.

```
! -*- f90 -*-
&CPL
!#####################################
!# A. DEFINITION OF VERTICAL AXES
!#####################################
!# SYNTAX:
!# - axis(dimension) name, channel, object, scaling (default=1.0), unit,
!#   positive_down (T,F), logarithmic (T) default: F,
!#   dimension length, axes values (max. 100)
!
! PRESSURE
VAX(1) = 'pax','COSMO','press',0.01,'hPa',T, F, 9, 1000.0, 800.0, 700.0, 600.0, 500.0, 400.0, 300.0, 200.0, 100.0, 91*0.0,
!
VAX(2) = 'logpax','COSMO','press',0.01,'hPa',T, T,8, 1000.0, 900.0, 800.0, 700.0, 600.0, 500.0, 400.0, 300.0, 92*0.0,
!
! ALTITUDE (geopot / g=9.81 m/s)
VAX(3) = 'zax','COSMO_ORI','hhl',1.0,'m', F, F, 8, 0.0, 2.0, 10.0, 100.0, 500.0, 1000.0, 2000.0, 3000.0, 92*0.0,
!
! POTENTIAL TEMPERATURE
VAX(4) = 'thax', 'COSMO','tpot', ,'K', F, F, 9, 340.0, 360.0, 380.0, 390.0, 400.0, 420.0, 440.0, 500.0, 550.0, 91*0.0,
!
!#
!########################################
!# B. FIELDS TRANSFORMED TO VERTICAL AXES
!########################################
!# SYNTAX:
!# - name, channel, object, axis, missing value (default: -1.0E+34)
!
! TEST SUITE: TRANSFORM THE AXIS DEFINING FIELDS:
!
TRA(1) = 'press',    'COSMO',     'press',   'pax',    ,
TRA(2) = 'lgpress',  'COSMO',     'press',   'logpax', ,
TRA(3) = 'geopot',   'COSMO_ORI', 'hhl',     'zax',    ,
TRA(4) = 'tpot',     'COSMO',     'tpot',    'thax',   ,
!
! A MORE REALISTIC CASE
TRA(5) = 'geopot_p', 'COSMO', 'geopot',  'pax',    ,
!
/
```

Diagnostic submodels: SCOUT

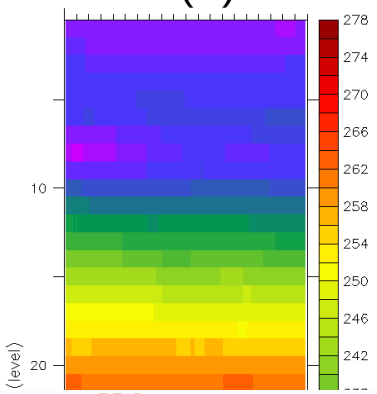
Stationary Column Output: enables high frequency output of model data at position of observation stations



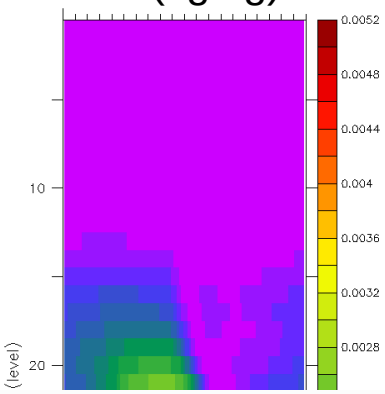
Diagnostic submodels: SCOUT

Stationary Column Output: enables high frequency output of model data at position of observation stations

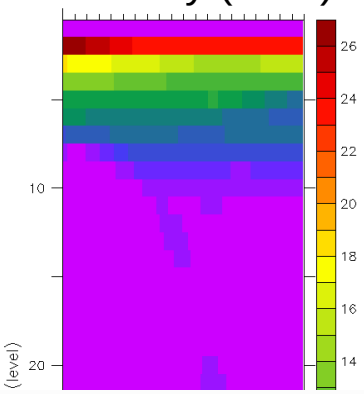
temperature (K)



spec. humidity (kg/kg)



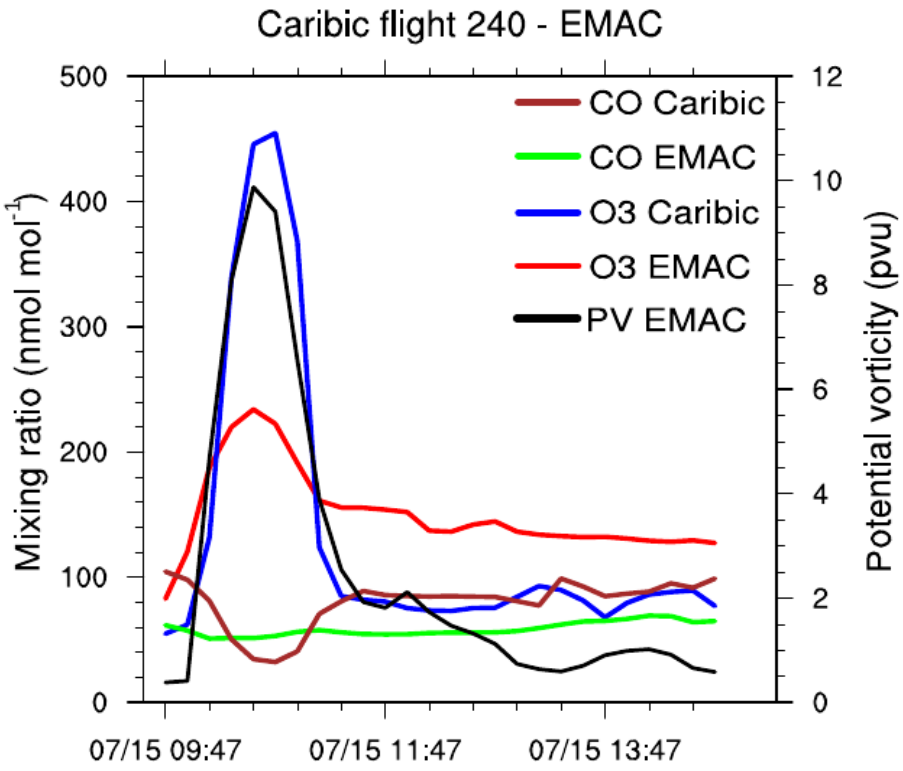
potential vorticity (PVU)

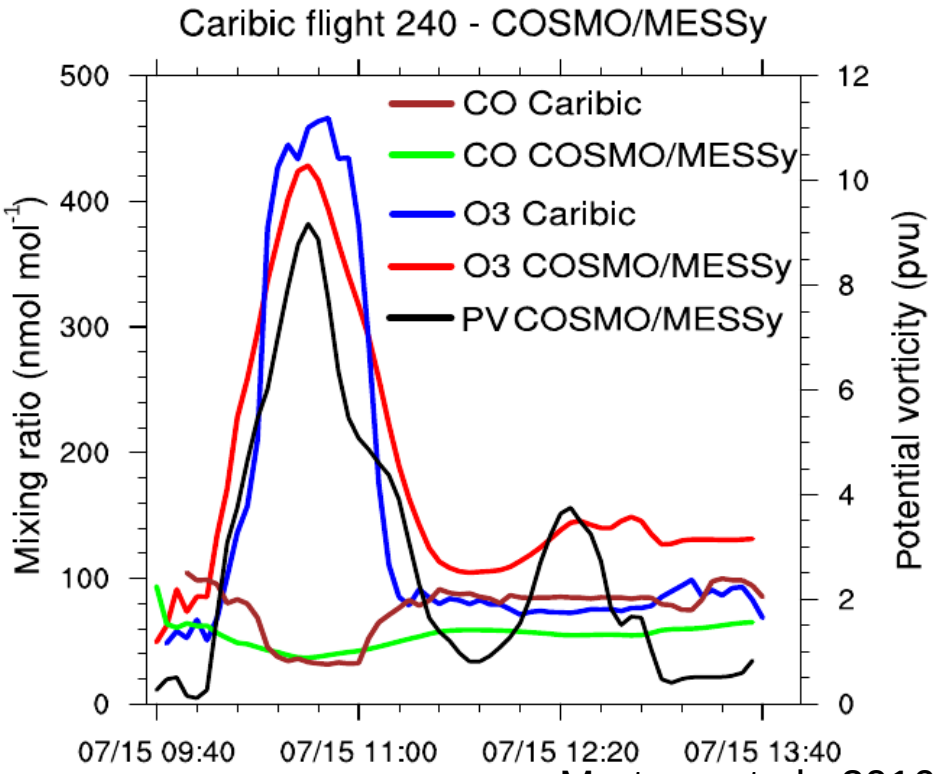



```
! -*- f90 -*-
&CPL
!# NAME, LATITUDE, LONGITUDE, LIST OF CHANNEL OBJECTS
!# NOTES:
!#    - NAME <= 5 CHARACTERS
!#    - SYNTAX FOR CHANNEL OBJECT LIST:
!#         "channel:object,object,object;channel:object;"
!#         (in object-names wildcards (*,?) can be used)
!
LOC(20)   = 'Bonn',   50.7,   7.1, "tracer_gp:*14C*,CO,N2O,H2;tropop
p:tp,PV,pblh;COSMO:press,tm1,qm1;",
!
/
```

Diagnostic submodels: S4D

Interpolation of data to tracks of moving observation platforms (aircraft, ship, train, …)

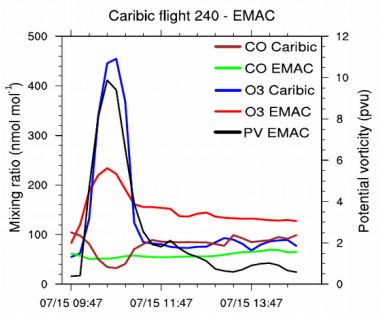




Mertens et al., 2016

Diagnostic submodels: S4D

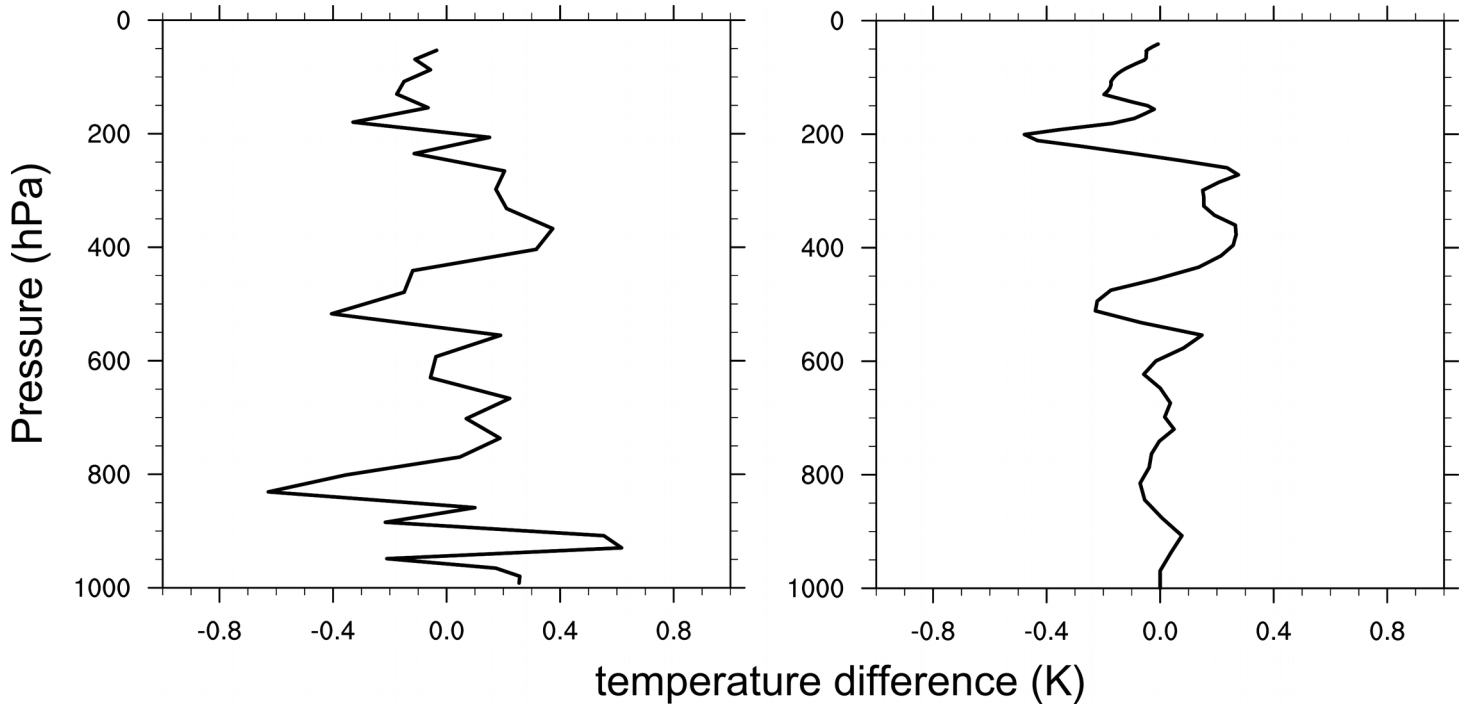
Interpolation of data to tracks of moving observation platforms (aircraft, ship, train, …)




```
! -*- f90 -*-
&CPL
!#############################################################################
!# SYNTAX: NAME, TRACK-DATA FILE BASE, UPDATE-SWITCH, COLUMN OUTPUT ?,
!#         OUTPUT ALL MODEL TIME STEPS ALONG TRACK, FILL VALUE
!#         LIST OF CHANNEL OBJECTS
!#
!# NOTES:
!#    - NAME <= 8 CHARACTERS
!#    - UPDATE SWITCH: -1: NEVER (SWITCHED OFF)
!#                      0: DAILY
!#                      1: MONTHLY
!#    - TRACK-DATA FILE NAMES
!#                <path>/<prefix><YYYY><MM><DD>.pos  (daily)   -> 0 !
!#                <path>/<prefix><YYYY><MM>.pos      (monthly) -> 1 !
!#    - SYNTAX FOR CHANNEL OBJECT LIST:
!#       "channel:object,object,object;channel:object;"
!#       (in object-names wildcards (*,?) can be used)
!#
!#    - THE TRACK-DATA FILES MUST CONTAIN:
!#      year month day hour minute second longitude latitude pressure [hPa]
!#
!#############################################################################
!
!# CARIBIC-1: daily position files
TRACK(1) = 'CARIBIC1', '$INPUTDIR/s4d/misc/CARIBIC-1/C1F_', 0, T, T, -1.E+34,
"tracer_gp:*;tropop:tp,PV;COSMO:geopot,tm1,qm1",
/
```

Diagnostic submodels: S4D

... or along radiosonde flight paths
(as long as GPS data are available)



offline sampled from hourly 3D field
minus
on-line sampled along flight path

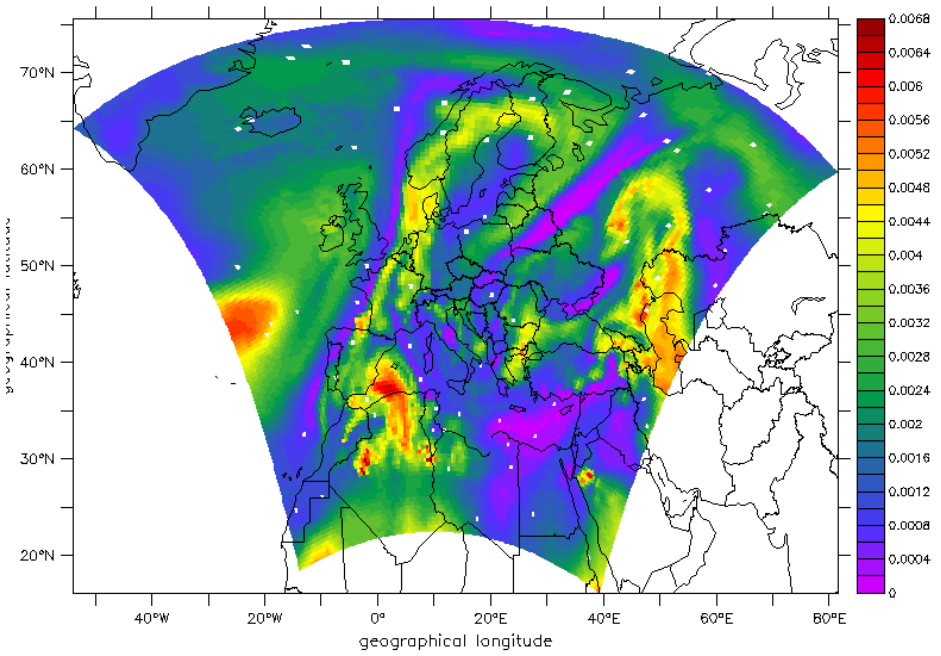
on-line sampled at fixed location
minus
on-line sampled along flight path

Example for station De Bilt @ 13.03.2017, 7 km model resolution

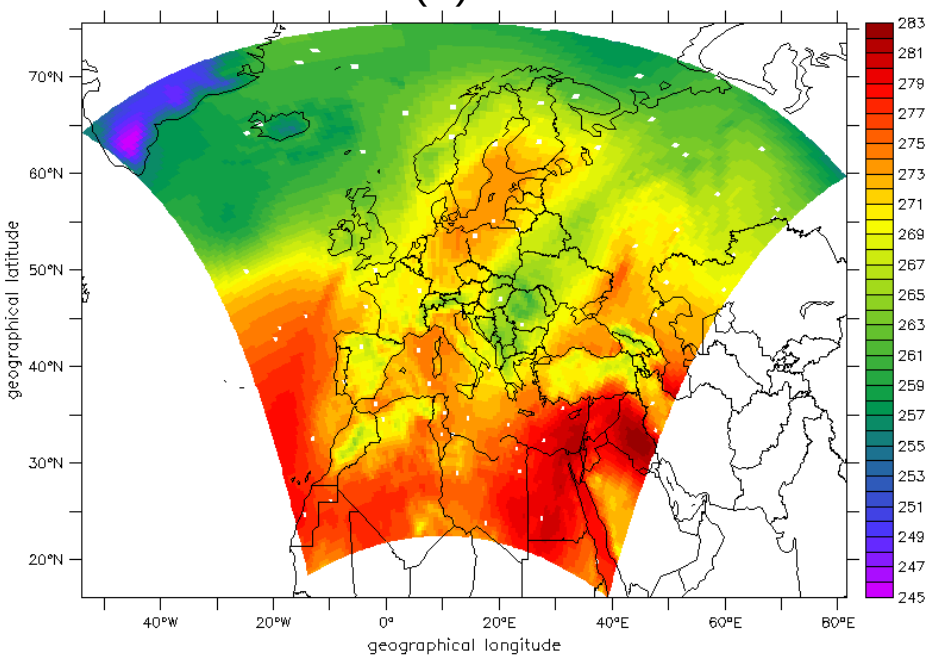
Diagnostic submodels: SORBIT

Sampling model data along sun-synchronous satellite orbits

spec. humidity
(kg/kg)



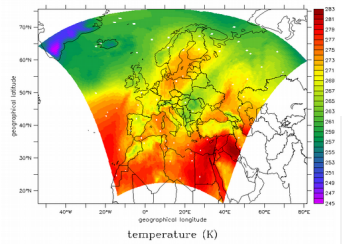
temperature
(K)



Envisat-A, 04.07.2010

Diagnostic submodels: SORBIT

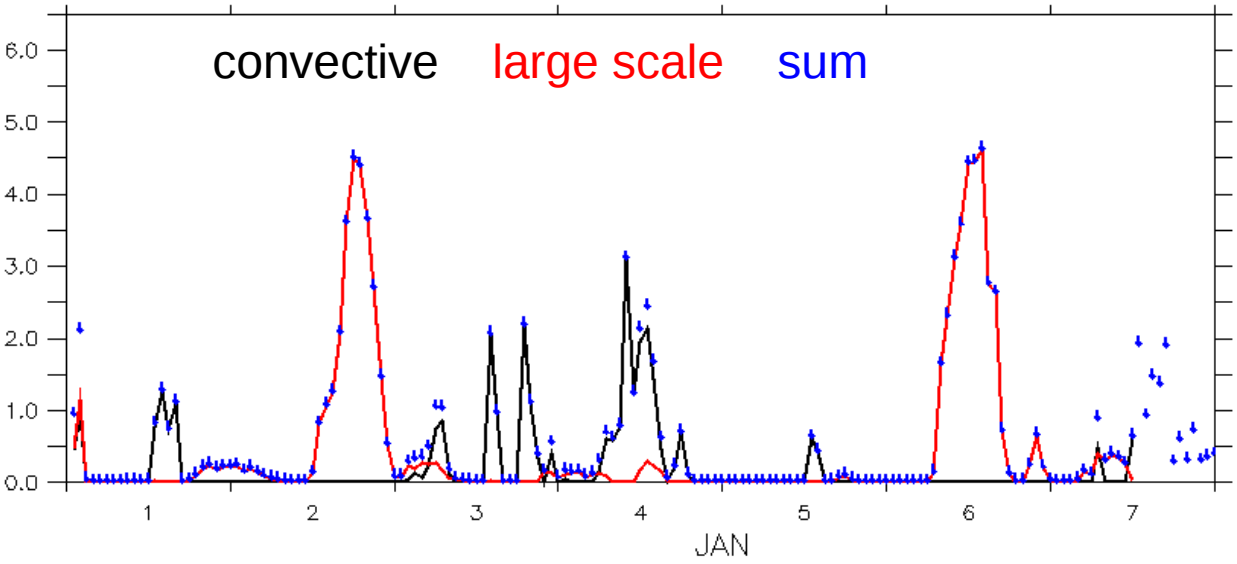
Sampling model data along sun-synchronous satellite orbits




```
! -*- f90 -*-
&CPL
! T: automatic daily output (highly recommended)
! F: output according to channel.nml (for testing)
lout_auto = T,
! value for re-initialisation at first time step of every day
r_init    = -1.0E+34,
!
!# SYNTAX: - NAME,
!#         - LATITUDE DEPENDENT LOCAL TIME (T,F), ORBIT INCLINATION (deg),
!#           ASCENDING(+1) (T) OR DESCENDING(-1) (F)
!#         - LOCAL TIME HOUR, LOCAL TIME (MINUTE) [EQUATOR CROSSING TIME],
!#           LIMIT DT TO LOCAL TIME DISTANCE ? (T,F)
!#         - LIST OF CHANNEL OBJECTS (max 600 characters)
!#
!# NOTES:
!#   - NAME <= 8 CHARACTERS
!#   - SYNTAX FOR CHANNEL OBJECT LIST:
!#       "channel:object,object,object;channel:object;"
!#       (in object-names wildcards (*,?) can be used)
!#
!# Nots:
!#    A:ascending, D:descending
!#    N:nadir, L:limb [f: forward (in flight direction), b: back]
!
! # ENVISAT: SCIAMACHI(N,Lf), MIPAS(Lb)
!
ORB(1)  = 'ENVI-A',  T, 98.5451, T, 22,00, F, 'COSMO:ps,geopot,qv,grvol;tracer_
gp:*;tropop:*;orbit:cossz*;'
!
/
```

Diagnostic submodels: SCALC

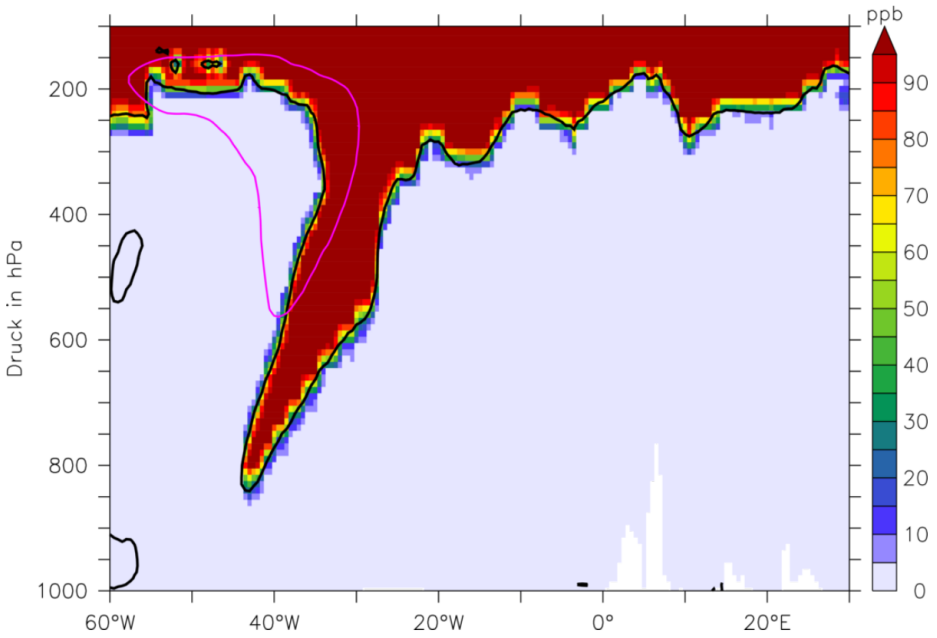
Simple calculation on channel objects




```
&CPL
CALC( 2)='PRC','COSMO ORI:PRR CON,PRS CON', 'SUM', 2
CALC( 3)='PRR','COSMO ORI:PRR GSP,PRS GSP', 'SUM', 2
CALC( 4)='TPR','COSMO ORI:PRR CON,PRS CON,PRR GSP,PRS GSP','SUM',2
/
```

Diagnostic tracer : Tracer initialisation (PTRACINI)

initialisation of tracer

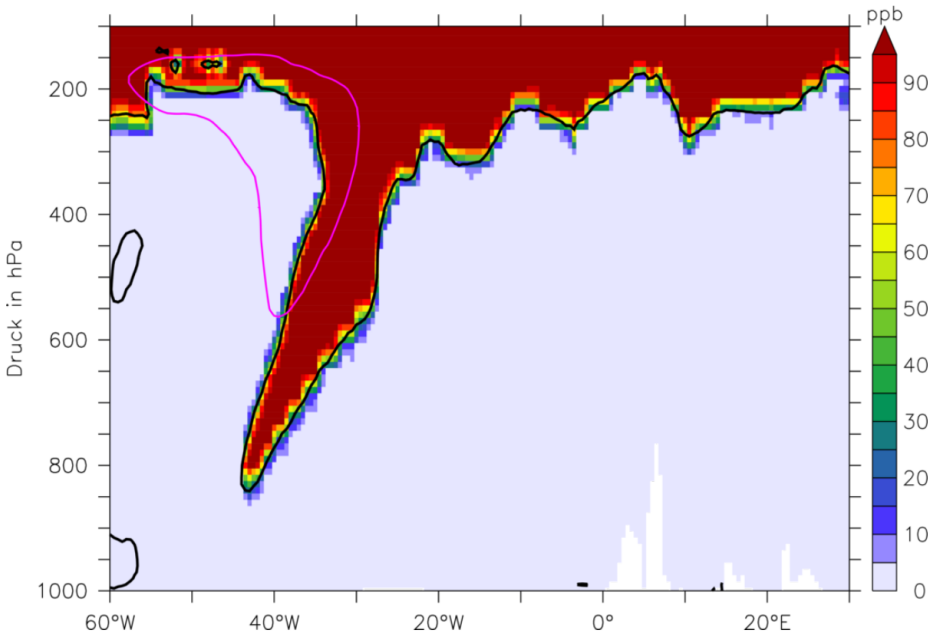


pressure [hPa]	PV [pvu]	QV [g/kg]
< 150		
150 – 900	≥ 2	≤ 1

Hofmann, 2014

Diagnostic tracer : Tracer initialisation (PTRACINI)

initialisation of tracer



pressure [hPa]	PV [pvu]	QV [g/kg]
< 150		
150 – 900	≥ 2	≤ 1

Hofmann, 2014

```
TRINI(1)%TRACER = 'strato','','','','','','','','','',
TRINI(1)%CRIT(1)='COSMO','press','<=',90000,'',''
TRINI(1)%CRIT(2)='COSMO','press','>=',15000,'',''
TRINI(1)%CRIT(3)='tropop','PV','>=',2,'',''
TRINI(1)%CRIT(4)='tracer_qp','QV','<=',0.001,'',''
```

Outline

- On-line diagnostic in MESSy ?
- Example: CHANNEL
  - on-line statistic, object attributes,
  - tailor-made output files (redirection of objects)
  - aveout
- Examples: Diagnostic Submodels
  - diagnostic of tropopause and boundary layer height
  - output at specified locations
  - output on iso-surfaces
  - algebraic combination of fields
- **How to get, use and extend COSMO-CLM/MESSy?**

How to get COSMO-CLM/MESSy ?

- So far, institutions become member of the MESSy consortium and individual users write a Letter of Intent.

- New: CLM-Members
  - Sign Letter of Intent (Template provided)
  - Acknowledge the MESSy Memorandum of Understanding

- After agreement to the licence, COSMO-CLM/MESSy is provided to the user.

- More details: see RedC link to MESSy on WG SUPTECH webpage

How to use / extend COSMO-CLM/MESSy ?

- The MESSy Wiki pages (open to MESSy users) provide an receipe on using COSMO/MESSy (also on RedC)

- 1 ½ day Workshop on “How to use COSMO-CLM/MESSy” in November
  (foodle link will be send shortly to clm-user mailing list)

  https://terminplaner2.dfn.de/foodle/COSMO-CLM-MESSy-Introduction-workshop-5b97dc09289c6

  - Possible dates in November 2018:
    5-6, 19-20, 26-27, 27-28, 28-29 or
    12.4.2019
  - Place: TBD (possible at DLR near Munich)

Backup

CHANNEL

CHANNEL …

- … is written in Fortran95 and
- … provides the following basic entities:
  - attributes — time independent, scalar characteristic (e.g., unit)
  - dimension variables — specific coordinate axis (e.g., latitude in degree N)
  - dimensions — basic geometry in one dimension
  - representations — describing multidimensional geometric structures (based on dimensions), e.g. Eulerian (or gridpoint), spectral, Lagrangian
  - channel objects — data fields including meta information (attributes) and underlying geometric structure (respresentation)
  - channels — set of related channel objects

Each channel produces an individual output file, if the output of the respective channel is requested in the channel namelist

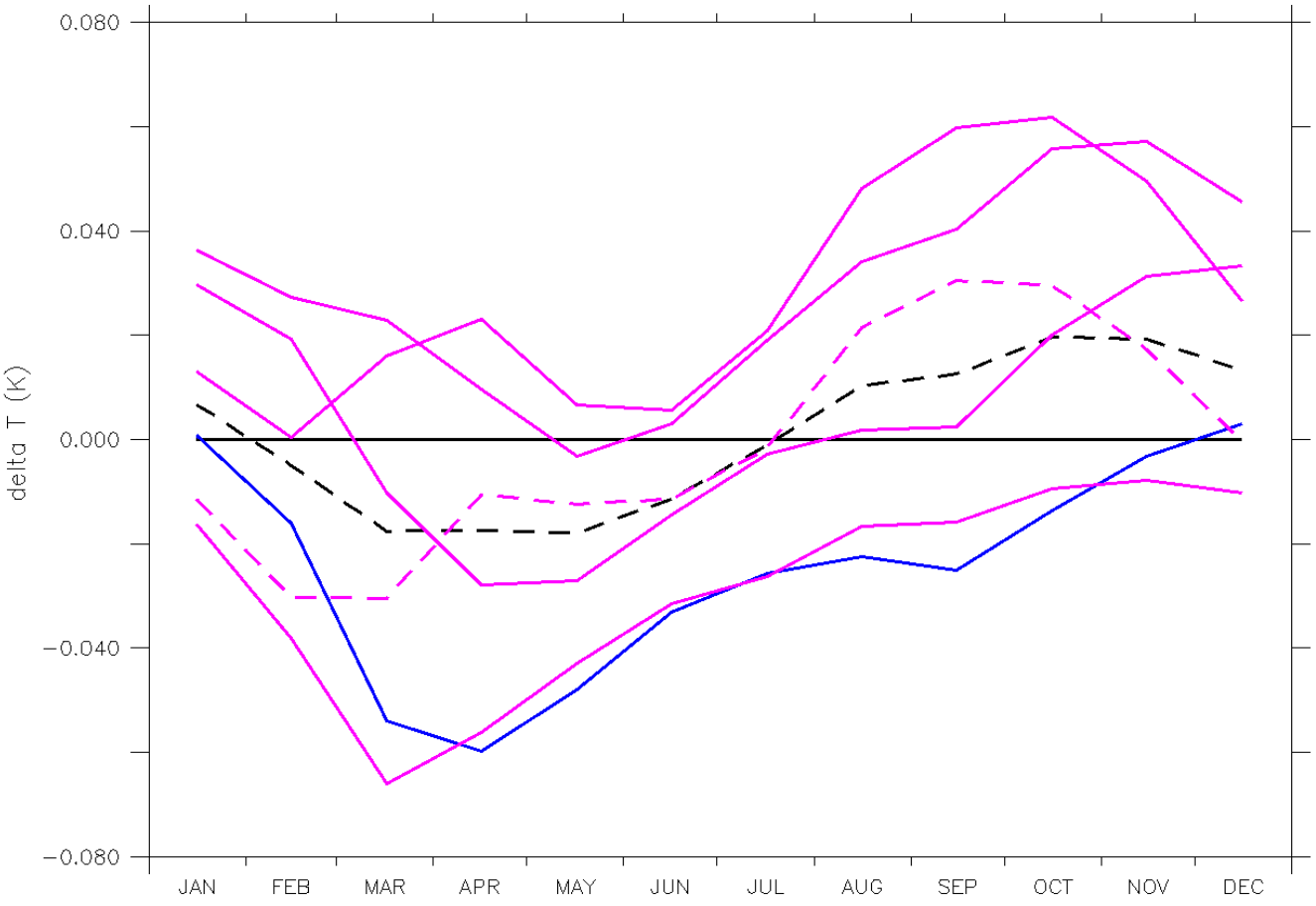
Adding global attributes to netcdf-files

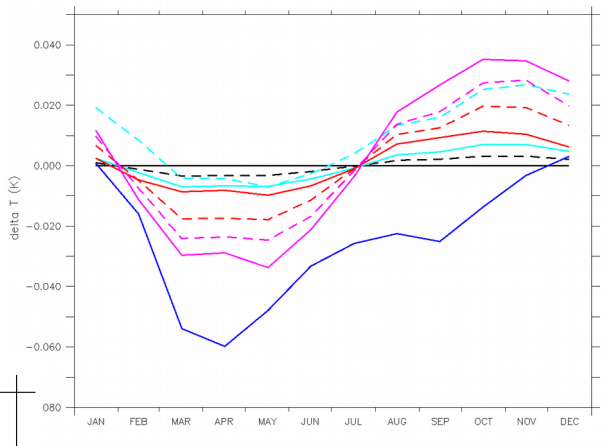
CHANNEL automatically adds global attributes to every netcdf-Files including important information like Compilers, Executable-Checksum etc…

```
// global attributes:
                :MESSy = "MESSy version d2.52, http://www.messy-interface.org" ;
                :MESSy_switch = "version 1.0" ;
                :MESSy_channel = "version 2.2" ;
                :MESSy_tracer = "version 2.3" ;
                :MESSy_timer = "version 0.1" ;
                :MESSy_qtimer = "version 2.0" ;
                :MESSy_import = "version 1.0" ;
                :MESSy_rnd = "version 1.1" ;
                :MESSy_cvtrans = "version 2.3" ;
                :MESSy_dradon = "version 2.1" ;
                :MESSy_tropop = "version 2.1" ;
                :MESSy_mmdclnt = "version 1.0" ;
                :MESSy_experiment = "cordex_eu_emt0" ;
                :EXEC_CHECKSUM = "7a5351f145066f3be0364c816613550f  bin/cosmo.exe (md5sum)" ;
                :RM = "COSMO version 5.0, Deutscher Wetterdienst, Offenbach" ;
                :rotated_pole = "coordinates of the rotated North Pole" ;
                :grid_mapping_name = "rotated_latitude_longitude" ;
                :grid_north_pole_latitude = 39.25 ;
                :grid_north_pole_longitud = -162. ;
                :northpole_grid_longitude = 0. ;
                :RM_start_date_time = "19790101 000000" ;
                :RM_timestep = 300. ;
                :F95_COMPILER_VERSION = "ifort (IFORT) 16.0.4 20160811" ;
                :F95_COMPILER_CALL = "mpif90" ;
                :F95_COMPILER_FLAGS = "-fpp -O2 -xCORE-AVX2 -fp-model precise -fp-model source
```

What is the difference between on-line and off-line calculated averages?

$\overline{T}_x - \overline{T}_i$ x =1,6,6^{+1},6^{+2},6^{+3},6^{+4},6^{+5} (K)



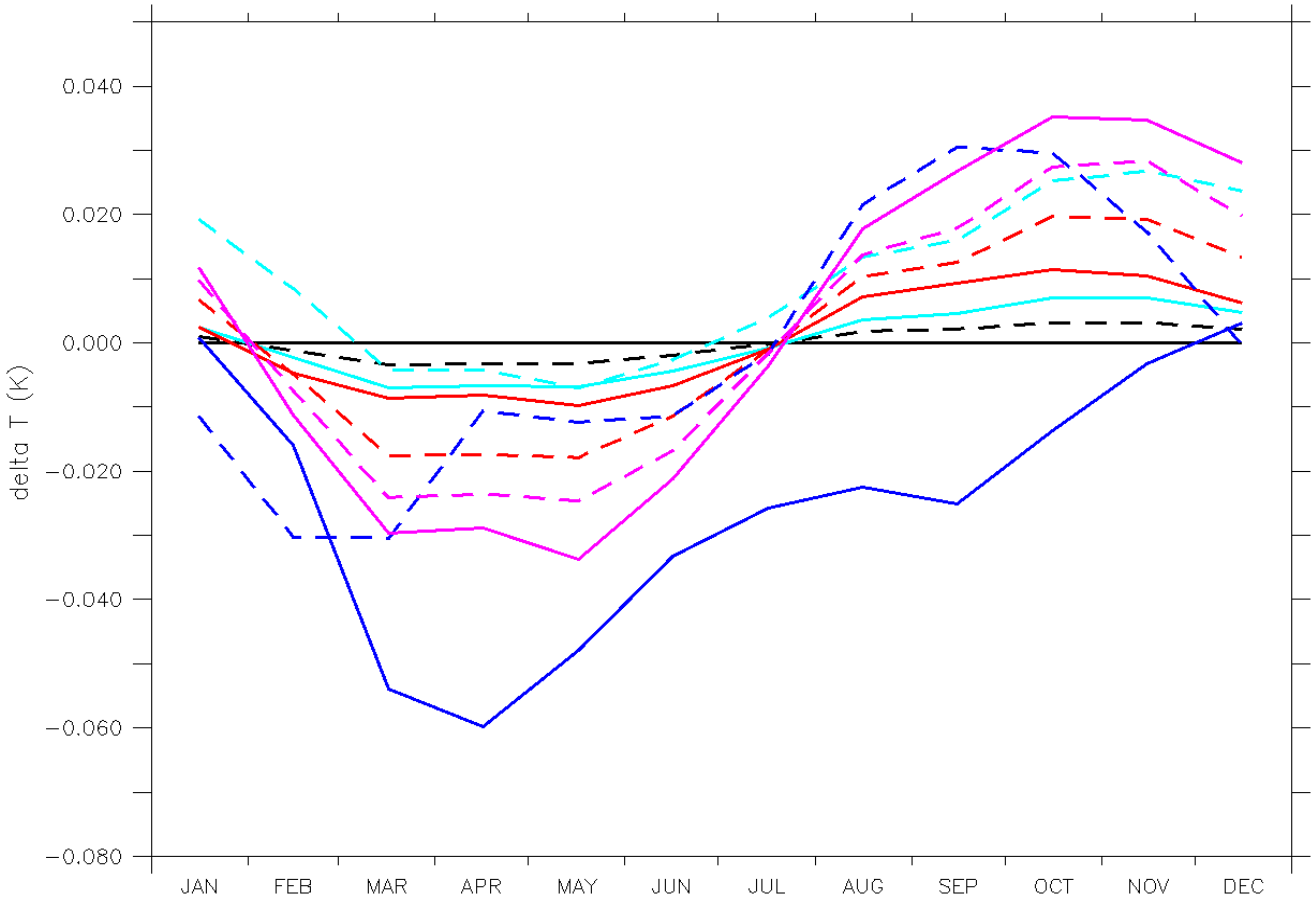


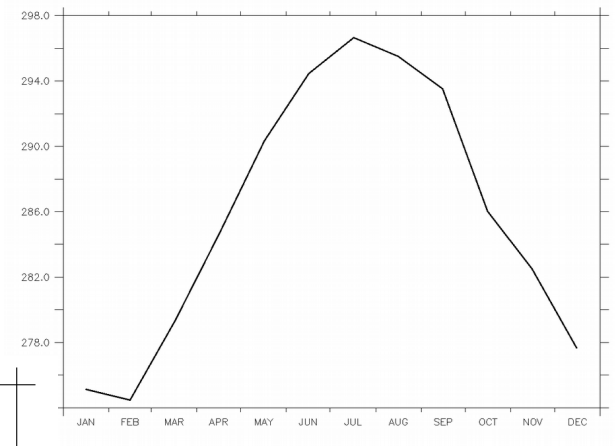
$\overline{T}_x - \overline{T}_i$ (K)

x =1,2,3,4,5,6,7,9

What is the difference between on-line and off-line calculated averages?

$\overline{T}_x - \overline{T}_i$ x =1,2,3,4,5,6,7,9,6^{+2} (K)





$\overline{T}_i$ (K)

Diagnostic tracer (1): Tracer definition (PTRAC)

definition of prognostic tracer

```
&CPL
!#############################################################################
!### SYNTAX FOR BASIC TRACER DEFINITION:
!###
!### TRAC(.) = 'list of tracer sets', 'list of tracer names', unit, \
!###             medium, quantity, type, \
!###             [aerosol model], [mode], [radius], [sigma], [density],
!###
TRAC(1) = 'gp;lg;s1;', 'strato;', 'mol/mol', 1, 1, 0, , , , , ,
!#############################################################################
!### SYNTAX FOR TRACER PROPERTIES DEFINITION:
!###
!### TPROP(.) = 'list of tracer sets', 'list of tracer names', \
!###             'container name', 'container contents',
!###
!###      - lists are separated by semicolon
!###      - container names are case sensitive
!###      - container content is case sensitive
!###      - ON = 1; OFF = 0
!#############################################################################
TPROP(1) = 'gp;lg;s1;', 'strato', 'molarmass ', '1.0',
/
```

Diagnostic tracer (2): Tracer initialisation (PTRACINI)

initialisation of tracer

```
&CPL
! SYNTAX for SET:
!INI1STEP:      - for onetime initialisation (if T, EMIS_IOEVENT is ignored,
!                 if F, EVENT_START is ignored )
!EVENT_START:   - set for onetime initialisation, is ignored for INI1STEP=F
!                - exact date to start the event (has to be a multiple of the
!                 timestep, because exact time is prompted)
!EMIS_IOEVENT: - set for continous initialisation, is ignored for INI1STEP=T
!                - use step-intervall (otherwise oszillation for leapfrog scheme)
!RELAX:         - if T, tracers are relaxed to coarser instance
!                - only usefull for COSMO, ignored for EMAC
!IFNOT:         - if T, field is initialised when crits are NOT fullfilled
!DOMAIN:        - if T, field is only initialised in coupeld domain
!
!TRACER: 'name_tr1', 'subname_tr1', 'name_tr2', 'subname_tr2', ...
!CRIT:   'channel', 'object', 'crit', const(_dp!) , 'compfld_cha','compfld_obj'

TRINI(1)%INI1STEP = T
TRINI(1)%EVENT_START = 2014,06,30,00,06,0
TRINI(1)%EMIS_IOEVENT  = 1,'steps','first',360
TRINI(1)%RELAX = T !only for COSMO
TRINI(1)%IFNOT = F
TRINI(1)%DOMAIN = F
TRINI(1)%TRACER = 'strato','','','','','','','','','',
TRINI(1)%CRIT(1)='COSMO','press','<=',90000,'',''
TRINI(1)%CRIT(2)='COSMO','press','>=',15000,'',''
TRINI(1)%CRIT(3)='tropop','PV','>=',2,'',''
TRINI(1)%CRIT(4)='tracer_gp','QV','<=',0.001,'',''
/
```

Diagnostic tracer (3): Point sources (TREXP)

Point sources and exponential decay

```
&CPL
! SYNTAX for SET:
!INI1STEP:      - for onetime initialisation (if T, EMIS_IOEVENT is ignored,
!                 if F, EVENT_START is ignored )
!EVENT_START:   - set for onetime initialisation, is ignored for INI1STEP=F
!                - exact date to start the event (has to be a multiple of the
!                 timestep, because exact time is prompted)
!EMIS_IOEVENT: - set for continous initialisation, is ignored for INI1STEP=T
!                - use step-intervall (otherwise oszillation for leapfrog scheme)
!RELAX:         - if T, tracers are relaxed to coarser instance
!                - only usefull for COSMO, ignored for EMAC
!IFNOT:         - if T, field is initialised when crits are NOT fullfilled
!DOMAIN:        - if T, field is only initialised in coupeld domain
!
!TRACER: 'name_tr1', 'subname_tr1', 'name_tr2', 'subname_tr2', ...
!CRIT:   'channel', 'object', 'crit', const(_dp!) , 'compfld_cha','compfld_obj'

TRINI(1)%INI1STEP = T
TRINI(1)%EVENT_START = 2014,06,30,00,06,0
TRINI(1)%EMIS_IOEVENT  = 1,'steps','first',360
TRINI(1)%RELAX = T !only for COSMO
TRINI(1)%IFNOT = F
TRINI(1)%DOMAIN = F
TRINI(1)%TRACER = 'strato','','','','','','','','','',
TRINI(1)%CRIT(1)='COSMO','press','<=',90000,'',''
TRINI(1)%CRIT(2)='COSMO','press','>=',15000,'',''
TRINI(1)%CRIT(3)='tropop','PV','>=',2,'',''
TRINI(1)%CRIT(4)='tracer_gp','QV','<=',0.001,'',''
/
```

Diagnostic tracer (3): Point sources (TREXP)

Point sources and exponential decay

```
&CPL
! SYNTAX for SET:
!INI1STEP:      - for onetime initialisation (if T, EMIS_IOEVENT is ignored,
!                 if F, EVENT_START is ignored )
!EVENT_START:   - set for onetime initialisation, is ignored for INI1STEP=F
!                - exact date to start the event (has to be a multiple of the
!                 timestep, because exact time is prompted)
!EMIS_IOEVENT: - set for continous initialisation, is ignored for INI1STEP=T
!                - use step-intervall (otherwise oszillation for leapfrog scheme)
!RELAX:         - if T, tracers are relaxed to coarser instance
!                - only usefull for COSMO, ignored for EMAC
!IFNOT:         - if T, field is initialised when crits are NOT fullfilled
!DOMAIN:        - if T, field is only initialised in coupeld domain
!
!TRACER: 'name_tr1', 'subname_tr1', 'name_tr2', 'subname_tr2', ...
!CRIT:   'channel', 'object', 'crit', const(_dp!) , 'compfld_cha','compfld_obj'

TRINI(1)%INI1STEP = T
TRINI(1)%EVENT_START = 2014,06,30,00,06,0
TRINI(1)%EMIS_IOEVENT  = 1,'steps','first',360
TRINI(1)%RELAX = T !only for COSMO
TRINI(1)%IFNOT = F
TRINI(1)%DOMAIN = F
TRINI(1)%TRACER = 'strato','','','','','','','','','',
TRINI(1)%CRIT(1)='COSMO','press','<=',90000,'',''
TRINI(1)%CRIT(2)='COSMO','press','>=',15000,'',''
TRINI(1)%CRIT(3)='tropop','PV','>=',2,'',''
TRINI(1)%CRIT(4)='tracer_gp','QV','<=',0.001,'',''
/
```