C2SM Center for Climate Systems Modeling

CONSORTIUM FOR SMALL SCALE MODELING COSMO

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology
MeteoSwiss

CLM

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Tracer module in COSMO

Anne Roches, C2SM
Oliver Fuhrer, MeteoSwiss

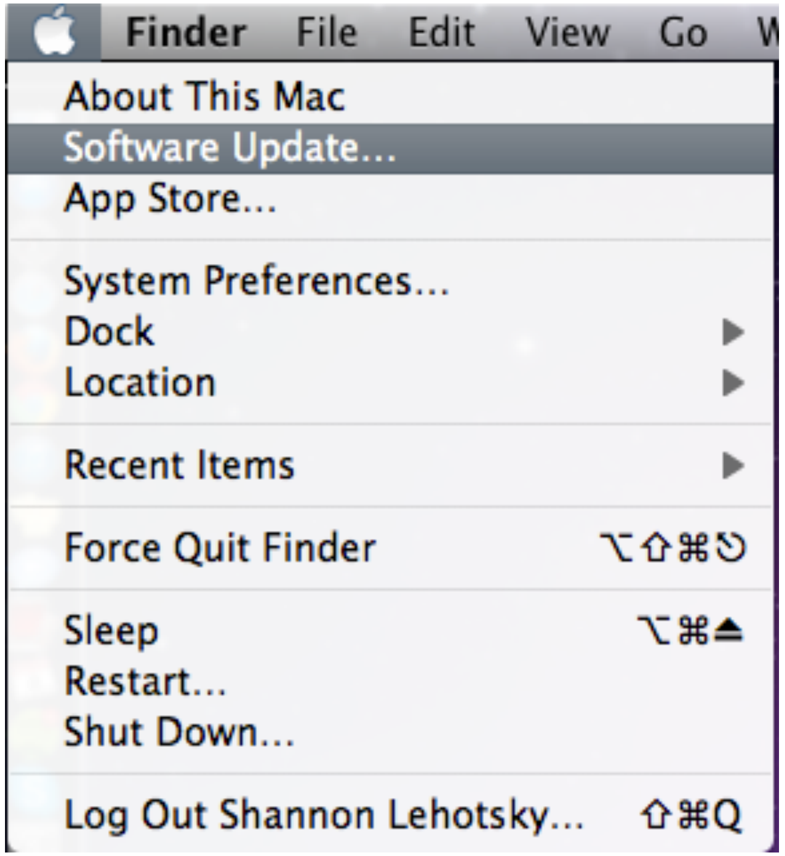
COSMO User Seminar, Offenbach

March 5, 2013

A new feature

Why?

Just one more update?



Do care! (everybody)

qv has disappeared!

~~No water vapor in COSMO anymore~~
~~An unparalleled case of dry bias?~~

At least a point for the modelers!

Do care! (tracer users)

Introduce a new tracer easily

```
CALL trcr_new(                                  &
              yshort_name = 'QV',               &
              igribparam  =  51 ,               &
              igribtable  =   1 ,               &
              …
              itype_adv   = T_ADV_ON,           &
              …
              itype_lbc   = T_LBC_FILE,         &
              …
              )
```

Do care! (tracer users)

Introduce a new tracer easily

```
CALL trcr_new(                                &
              yshort_name = 'QV',             &
              igribparam  =  51 ,             &
              igribtable  =   1 ,             &
              …
              itype_adv   = T_ADV_ON,         &
              …
              itype_lbc   = T_LBC_FILE,       &
              …
              )
```

Guarantees a coherent treatment

Functionality

$$\frac{\partial \psi}{\partial t} = ADV + DIFF + TURB + CONV + RELAX + DAMP + \cancel{SRC/SINKS}$$

Functionality

$$\frac{\partial \psi}{\partial t} = ADV + DIFF + TURB + CONV + RELAX + DAMP + SRC\,\cancel{/\,SINKS}$$

• Advection	✓ (all dycores & adv. schemes)
• Horizontal hyperdiffusion	✓
• Turbulent mixing	✓ (impl. TKE scheme only)
• Passive transport by convection	✓ (Tiedtke only)
• Boundary relaxation	✓ (full or at inflow)
• Rayleigh damping	✓ (both types)

Functionality

C2SM Center for Climate Systems Modeling

$$\frac{\partial \psi}{\partial t} = ADV + DIFF + TURB + CONV + RELAX + DAMP + \cancel{SRC / SINKS}$$

• Advection	✓ (all dycores & adv. schemes)
• Horizontal hyperdiffusion	✓
• Turbulent mixing	✓ (impl. TKE scheme only)
• Passive transport by convection	✓ (Tiedtke only)
• Boundary relaxation	✓ (full or at inflow)
• Rayleigh damping	✓ (both types)
• Memory management	✓
• I/O (incl. restart)	✓
• Lateral and initial boundary conditions	✓ (all existing options + new)
• Clipping	✓ (“sd method” only)

The tracer module

4 new files:

- `data_tracer.f90`

  Constants for the tracer module

- `data_tracer_metadata.f90`

  Internal information for the metadata module

- `src_tracer.f90`

  Tracer routines for the users

- `src_tracer_metadata.f90`

  Low-level metadata routines

The tracer API

User subroutines (I):

- **trcr_new** : definition of a new tracer
  - Provide its name, Grib n°/table, units, and optionally decide which operations it should undergo
  - Get optionally its index
- **trcr_get** : access to a tracer
  - Provide the name/index of the tracer and optionally a time level
  - Get optionally its data at the specified time level, its boundary data, or its tendency
- **trcr_get_ntrcr** : access to the total number of tracers
- **trcr_get_index** : retrieve the tracer index (identifier)
- **trcr_errorstr** : retrieve an error message

The tracer API

User subroutines (II):

- **trcr_meta_define** : definition of a new metadata
  - Provide name of metadata, default value for it, and optionally decide if it is protected
  - Get optionally its index

- **trcr_meta_set** : setting of a metadata
  - Provide name/index of the metadata, optionally name/index of the tracer and value for the metadata for this tracer

- **trcr_meta_get** : access to a metadata
  - Provide name/index of the metadata, name/index of the tracer
  - Get the value for the metadata for this tracer

Remark: these interfaces are valid for int, real*4, real*8, string, logical and for arrays of these types as well as for pointers

The tracer API

Infrastructure subroutines:

• **trcr_init**	: initialization of the tracer structure
• **trcr_alloc**	: allocation of the memory for all tracers (data, boundaries, tendencies)
• **trcr_setup_vartab**	: I/O (mimic src_setup_vartab)
• **trcr_print**	: print of the list of tracers and associated metadata in sd out
• **trcr_cleanup**	: deallocation of the memory

$\left.\right\}$ **1 x**

Current use

1) Standard (1M) microphysics: qx

2) 2-moment scheme (U. Blahak): 7 additional species

3) CarboCount project (talk D. Brunner): CO_2 & CH_4

4) COSMO-ART (H. & B. Vogel): dozens of gases, pollens

5) Tracing of Stratospheric air (B. Skerlak, D. Kunkel)

6) Snow water content forecast (C. Frick)

Changes for existing variables

- qx changes in ~ all modules; bit-identical results

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?

Old world

New world

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?
    - Access: USE -> CALL trcr_get

Old world

```
USE data_fields: qi
```

New world

```
 USE src_tracer: trcr_get
REAL, POINTER::qi(:,:,:)=> NULL()

 CALL trcr_get( ‘QI’, nnew, qi )
```

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?
  - Access: USE -> CALL trcr_get
  - Pointers instead of allocatable variables (! allocated -> associated)

Old world

```
USE data_fields: qi




IF (ALLOCATED(qi)) THEN


ENDIF
```

New world

```
USE src_tracer: trcr_get
REAL, POINTER::qi(:,:,:)=> NULL()


CALL trcr_get( ‘QI’, nnew, qi )
IF (ASSOCIATED(qi)) THEN


ENDIF
```

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?
  - Access: USE -> CALL trcr_get
  - Pointers instead of allocatable variables (! allocated -> associated)
  - Shape: (ie, je, ke, nztlev) -> (ie, je, ke)

Old world

```
USE data_fields: qi




IF (ALLOCATED(qi)) THEN
 qi(:,:,:,nnew) = zeta
ENDIF
```

New world

```
USE src_tracer: trcr_get
REAL, POINTER::qi(:,:,:)=> NULL()

CALL trcr_get( 'QI', nnew, qi )
IF (ASSOCIATED(qi)) THEN
  qi(:,:,:) = zeta    I
ENDIF
```

Changes for existing variables

- qx changes in ~ all modules; bit-identical results
- What changes for you?
  - Access: USE -> CALL trcr_get
  - Pointers instead of allocatable variables (! allocated -> associated)
  - Shape: (ie, je, ke, nztlev) -> (ie, je, ke)
  - One pointer/time level; only valid for one time step
  - Incoherencies appear clearly ("hacks" in organize_physics.f90)

Old world

```
USE data_fields: qi




IF (ALLOCATED(qi)) THEN
 qi(:,:,:,nnew) = zeta
ENDIF
```

New world

```
 USE src_tracer: trcr_get
REAL, POINTER::qi(:,:,:)=> NULL()

 CALL trcr_get( 'QI', nnew, qi )
 IF (ASSOCIATED(qi)) THEN
   qi(:,:,:) = zeta    I
 ENDIF
```

Metadata: CO_2 emissions example (I)

- Emissions needed for CO_2 modeling
- Not part of the tracer module but can be used for this

Metadata: CO_2 emissions example (I)

- Emissions needed for CO_2 modeling
- Not part of the tracer module but can be used for this

```
CALL trcr_meta_define( 'ITYPE_EMISS', 0 )
CALL trcr_meta_set    ( 'CO2', 'ITYPE_EMISS', 1 )
 !0: no emissions, 1: emissions from file
```

Metadata: CO_2 emissions example (I)

- Emissions needed for CO_2 modeling
- Not part of the tracer module but can be used for this

Define if the tracer has an emission

```
CALL trcr_meta_define( 'ITYPE_EMISS', 0 )
CALL trcr_meta_set    ( 'CO2', 'ITYPE_EMISS', 1 )
 !0: no emissions, 1: emissions from file
```

Metadata: CO_2 emissions example (I)

- Emissions needed for CO_2 modeling
- Not part of the tracer module but can be used for this

Define if the tracer has an emission

```
CALL trcr_meta_define( 'ITYPE_EMISS', 0 )
CALL trcr_meta_set   ( 'CO2', 'ITYPE_EMISS', 1 )
 !0: no emissions, 1: emissions from file

…

CALL trcr_meta_define( 'EMISS_FIELD', NULL() )
IF ( itype_emiss /= 0 ) THEN
  CALL trcr_meta_set( 'CO2', 'EMISS_FIELD', co2_e )
ENDIF
```

Metadata: CO_2 emissions example (I)

- Emissions needed for CO_2 modeling
- Not part of the tracer module but can be used for this

Define if the tracer has an emission

```
CALL trcr_meta_define( 'ITYPE_EMISS', 0 )
CALL trcr_meta_set    ( 'CO2', 'ITYPE_EMISS', 1 )
 !0: no emissions, 1: emissions from file
```

…

If yes, define a pointer to the emission

```
CALL trcr_meta_define( 'EMISS_FIELD', NULL() )
IF ( itype_emiss /= 0 ) THEN
  CALL trcr_meta_set( 'CO2', 'EMISS_FIELD', co2_e )
ENDIF
```

Metadata: CO2 emissions example (II)

```
DO i = 1, ntrcr

  CALL trcr_meta_get(i,'ITYPE_EMISS', itype_emiss)

  IF (itype_emiss == 1 ) THEN

    CALL trcr_get      (i, trcr_tens)
    CALL trcr_meta_get(i,'EMISS_FIELD', trcr_emiss)

    trcr_tens(:,:,:) = trcr_tens(:,:,:) + trcr_emiss(:,:,:)

  ENDIF

ENDDO
```

Metadata: CO2 emissions example (II)

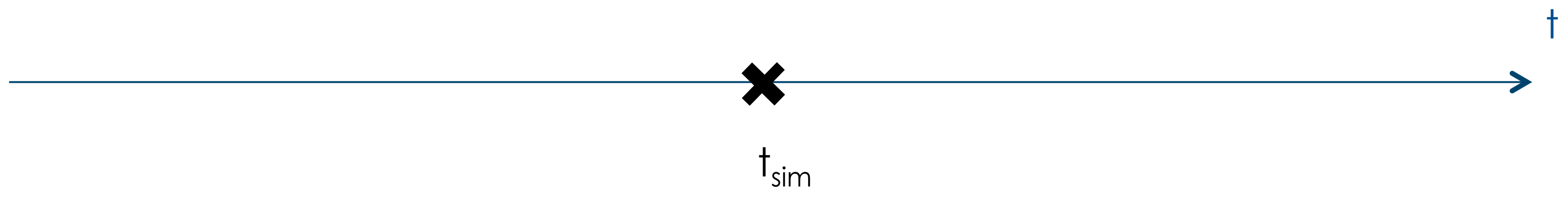
```
DO i = 1, ntrcr

  CALL trcr_meta_get(i,'ITYPE_EMISS', itype_emiss)

  IF (itype_emiss == 1 ) THEN

    CALL trcr_get      (i, trcr_tens)
    CALL trcr_meta_get(i,'EMISS_FIELD', trcr_emiss)

    trcr_tens(:,:,:) = trcr_tens(:,:,:) + trcr_emiss(:,:,:)

  ENDIF

ENDDO
```

Loop over all tracers

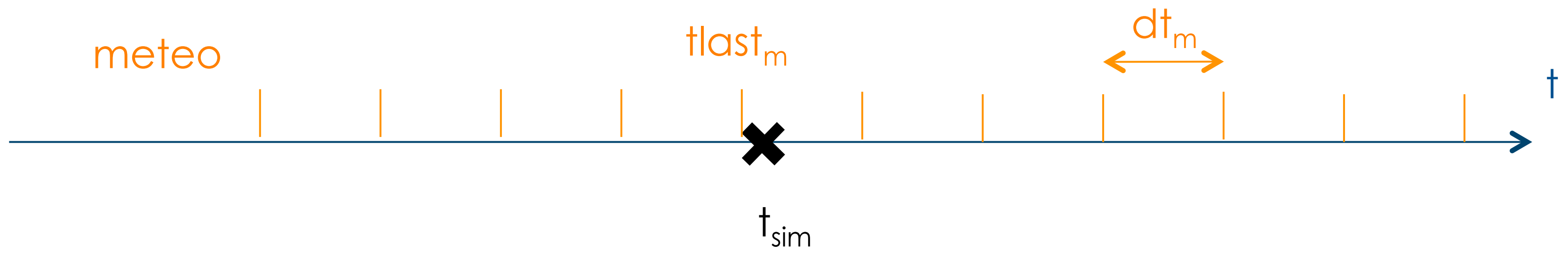
If the tracer has an emission ….

Then, update the tendency using the emission

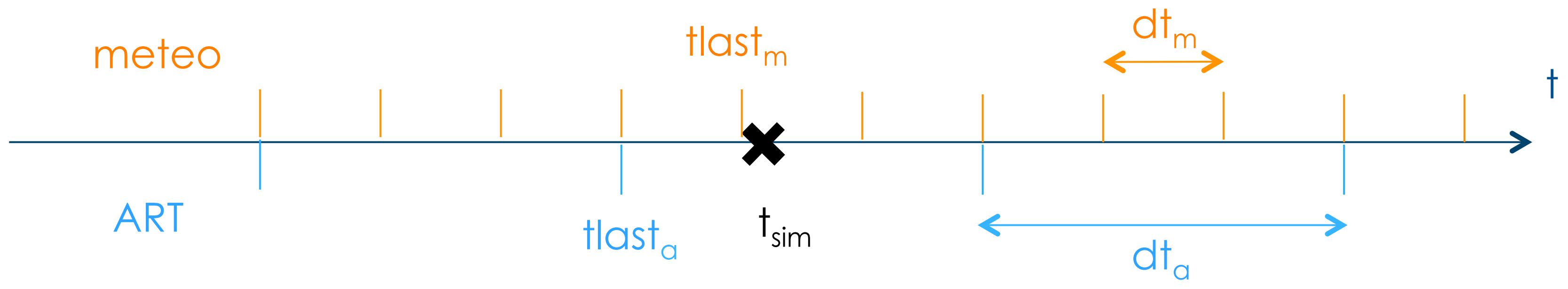
Missing feature: an ART example



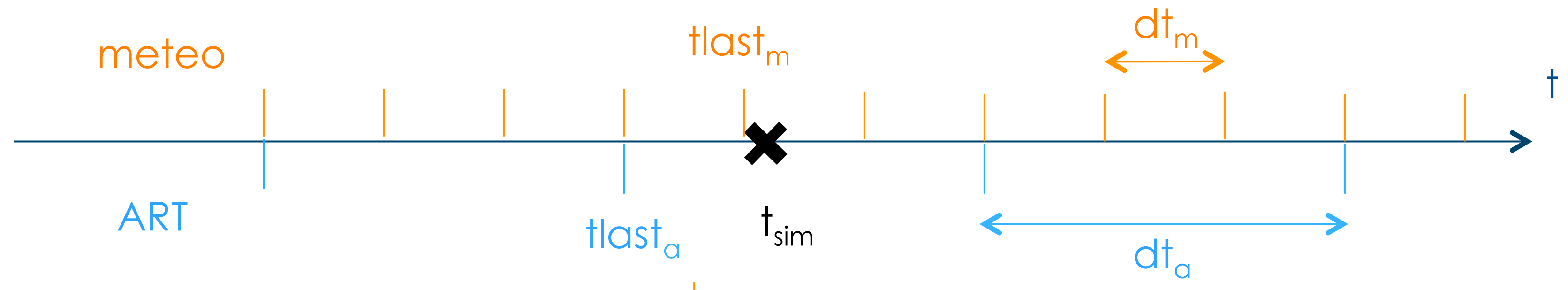
Missing feature: an ART example



Missing feature: an ART example



Missing feature: an ART example



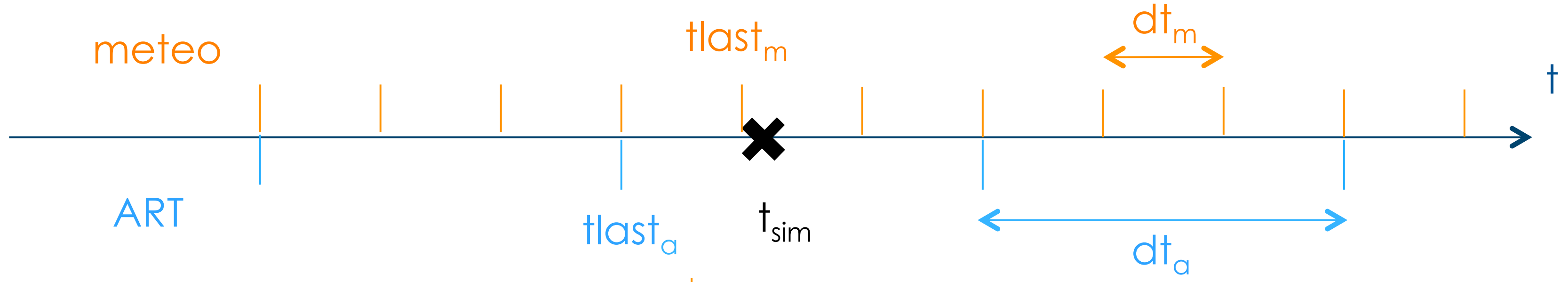
COSMO(tracer) world

```
z2m = (tsim+1-tlastm)/dtm
z1m = 1 - z2m

DO i=1, ntrcr
 trcr_new(:,:,:) =
   z1m*trcr_bd(:,:,:,nbd1m)
  +z2m*trcr_bd(:,:,:,nbd2m)
ENDDO
```

COSMO-ART world

```
z2a = (tsim+1-tlasta)/dta
 z1a = 1 - z2a

DO i=1, ngas
 cgas(:,:,:,nnew) =
    z1a*cgas_bd(:,:,:,nbd1a)
   +z2a*cgas_bd(:,:,:,nbd2a)
ENDDO
```

Missing feature: an ART example



COSMO(tracer) world

```
z2m = (tsim+1-tlastm)/dtm
z1m = 1 - z2m

DO i=1, ntrcr
 trcr_new(:,:,:) =
   z1m*trcr_bd(:,:,:,nbd1m)
  +z2m*trcr_bd(:,:,:,nbd2m)
ENDDO
```

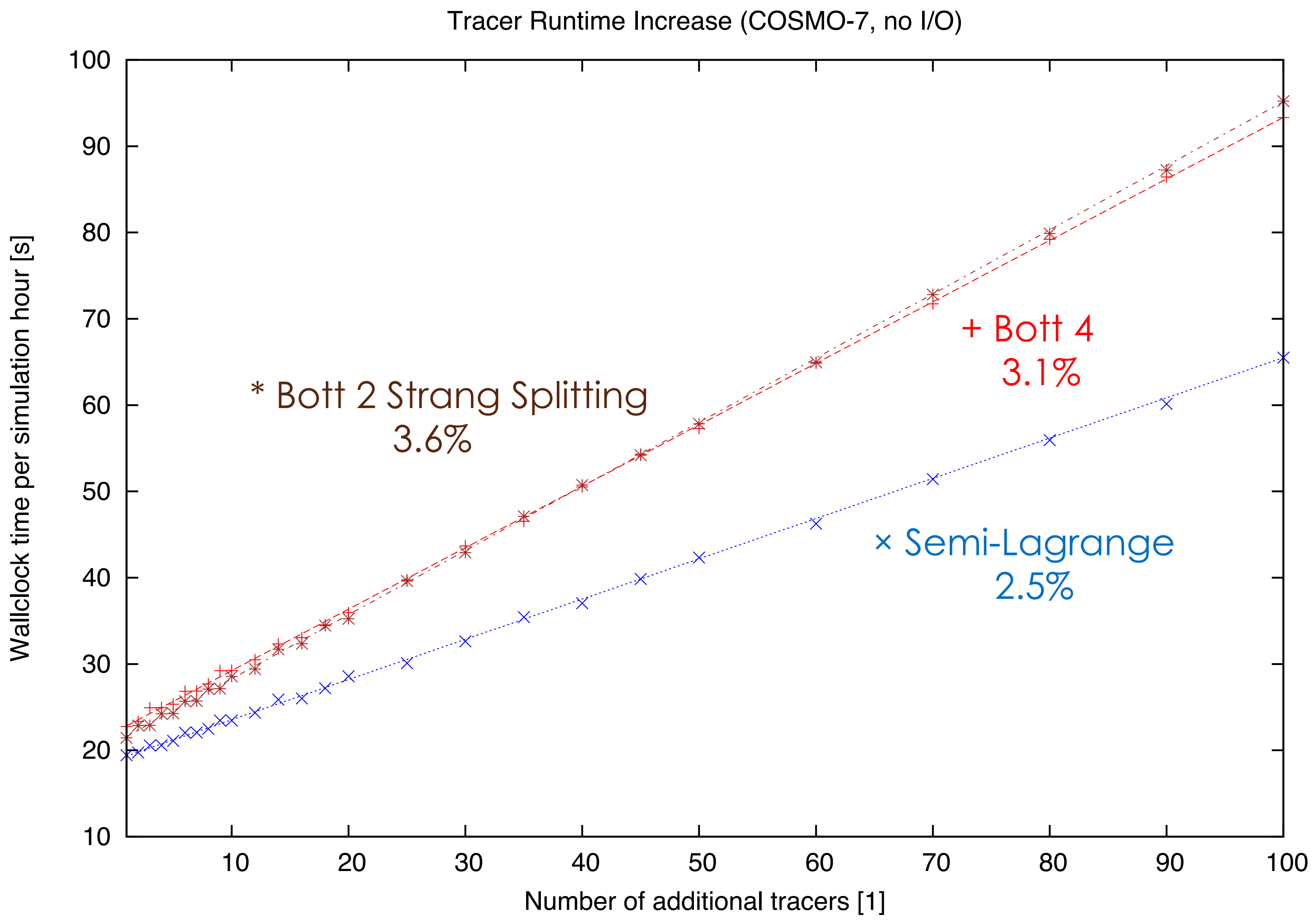
COSMO-ART world

```
z2a = (tsim+1-tlasta)/dta
 z1a = 1 - z2a

DO i=1, ngas
 cgas(:,:,:,nnew) =
    z1a*cgas_bd(:,:,:,nbd1a)
   +z2a*cgas_bd(:,:,:,nbd2a)
ENDDO
```

JUST TELL US!

How much does a tracer cost?

C2SM Center for Climate Systems Modeling



Main message

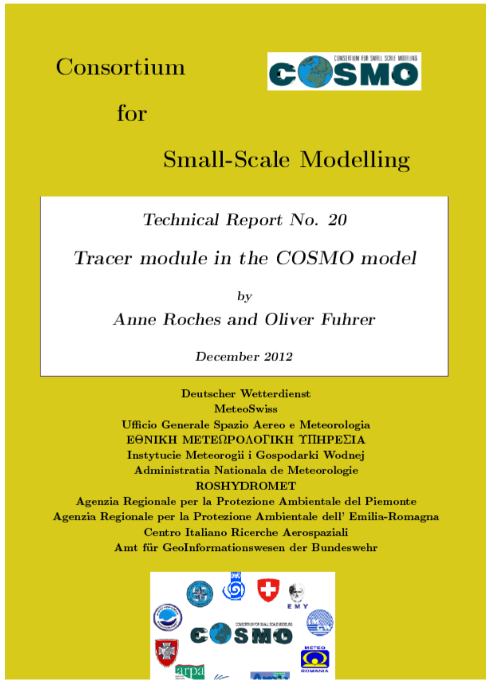
Use this module to include new prognostic variables

Main message

Use this module to include new prognostic variables

Read the documentation

http://cosmo-model.org/content/model/documentation/techReports


Consortium

for

Small-Scale Modelling

Technical Report No. 20

Tracer module in the COSMO model

by

Anne Roches and Oliver Fuhrer

December 2012

Deutscher Wetterdienst
MeteoSwiss
Ufficio Generale Spazio Aereo e Meteorologia
ΕΘΝΙΚΗ ΜΕΤΕΩΡΟΛΟΓΙΚΗ ΥΠΗΡΕΣΙΑ
Instytucie Meteorogii i Gospodarki Wodnej
Administratia Nationala de Meteorologie
ROSHYDROMET
Agenzia Regionale per la Protezione Ambientale del Piemonte
Agenzia Regionale per la Protezione Ambientale dell' Emilia-Romagna
Centro Italiano Ricerche Aerospaziali
Amt für GeoInformationswesen der Bundeswehr

Main message

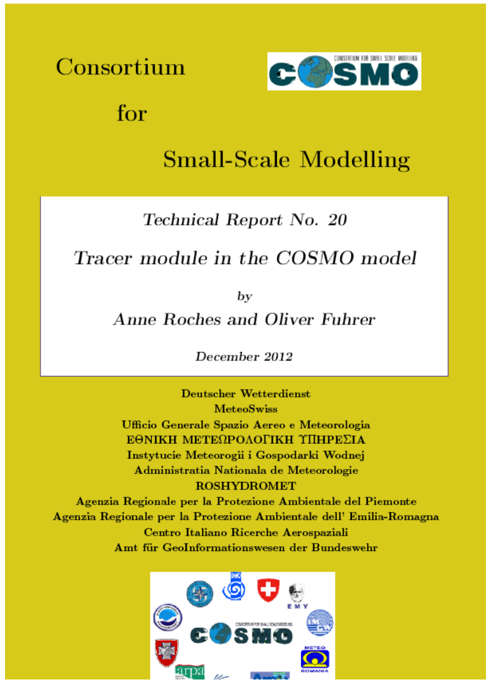
Use this module to include new prognostic variables

Read the documentation

http://cosmo-model.org/content/model/documentation/techReports

If a feature is missing, contact me

anne.roches@env.ethz.ch



Consortium for Small-Scale Modelling

Technical Report No. 20

Tracer module in the COSMO model

by

Anne Roches and Oliver Fuhrer

December 2012

Deutscher Wetterdienst
MeteoSwiss
Ufficio Generale Spazio Aereo e Meteorologia
ΕΘΝΙΚΗ ΜΕΤΕΩΡΟΛΟΓΙΚΗ ΥΠΗΡΕΣΙΑ
Instytucie Meteorogii i Gospodarki Wodnej
Administratia Nationala de Meteorologie
ROSHYDROMET
Agenzia Regionale per la Protezione Ambientale del Piemonte
Agenzia Regionale per la Protezione Ambientale dell' Emilia-Romagna
Centro Italiano Ricerche Aerospaziali
Amt für GeoInformationswesen der Bundeswehr

Main message

Use this module to include new prognostic variables

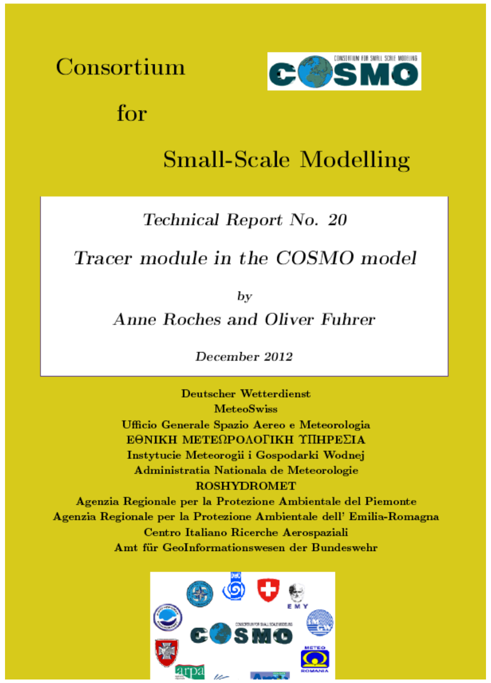
Read the documentation

http://cosmo-model.org/content/model/documentation/techReports

If a feature is missing, contact me

anne.roches@env.ethz.ch

Applying the COSMO standards procedure is feasible ☺



Consortium for Small-Scale Modelling

Technical Report No. 20

Tracer module in the COSMO model

by

Anne Roches and Oliver Fuhrer

December 2012

Deutscher Wetterdienst
MeteoSwiss
Ufficio Generale Spazio Aereo e Meteorologia
ΕΘΝΙΚΗ ΜΕΤΕΩΡΟΛΟΓΙΚΗ ΥΠΗΡΕΣΙΑ
Instytucie Meteorogii i Gospodarki Wodnej
Administratia Nationala de Meteorologie
ROSHYDROMET
Agenzia Regionale per la Protezione Ambientale del Piemonte
Agenzia Regionale per la Protezione Ambientale dell' Emilia-Romagna
Centro Italiano Ricerche Aerospaziali
Amt für GeoInformationswesen der Bundeswehr

Would be nice to have

Better field management in COSMO:

• **field_new**	: definition of a new field
• **field_get**	: access to a field

Would be nice to have

Better field management in COSMO:

- **field_new** : definition of a new field
- **field_get** : access to a field

Convenient I/O for the COSMO variables

Would be nice to have

Better field management in COSMO:

- **field_new** : definition of a new field
- **field_get** : access to a field

Convenient I/O for the COSMO variables

Flexible handling of associated fields

C2SM
Center for Climate
Systems Modeling

Thank you!!!

And happy tracing!