TM5 : Versioning, Coding Standards, & Documentation

(extra: Iceland's Eyjafjallajokull volcano ash dispersion with TM)

P. Le Sager (KNMI)

with W. Peters (svn), A. Segers (svn),
J. Williams (Eyja), M. van Weele (Eyja)

SVN – repos status

- merged Maunaloa and KNMI repositories (beginning of 2010)
- merged TM5-chem of Huijnen et al. (2010) with the latest trunk
- release 3.0 (june 2010)
  - following TM5-chem releases will be benchmark against prior release and observations => documented history of code/results changes (work just started)

User-Friendly SVN

New Tools Directory

available thru install script

Files list

root / tools /

Checkout URL : https://svn.knmi.nl/svn/TM5/tools/

Apply rights on /tools/

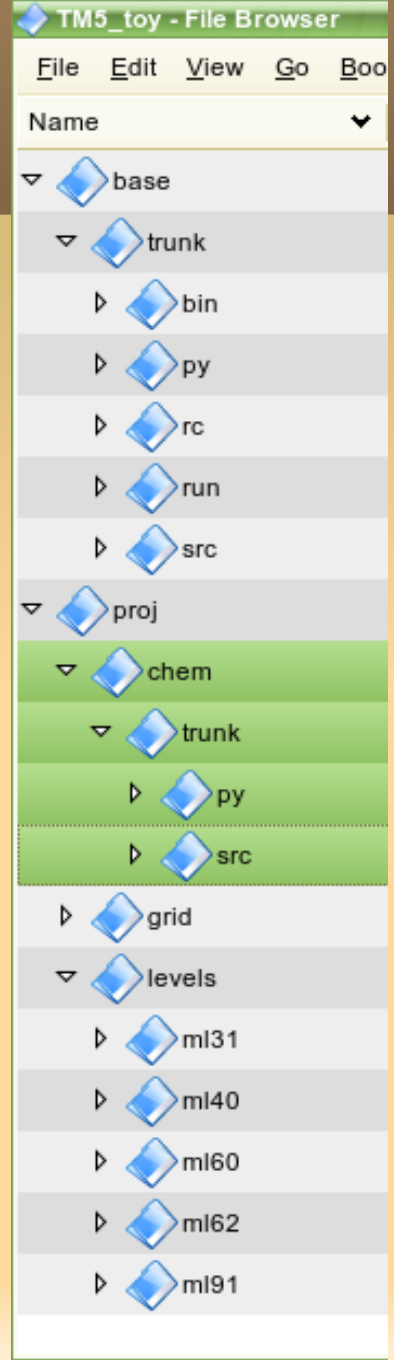
Name	File revision	Action
..		
diadem		
mdf		
protex		
svntools		
TIPP		

TM5 SVN project - a story of many (in)dependent sub-projects

- add complexity to:
  - Versioning (some has no trunk, or no release)
  - Installation script (many options to sort, 75 projects!)
  - Working with your own copy :
    - “svn update / commit / status” applies to only one subset of your installation
    - several svn command may require repository URL : “svn mkdir / add”

- no commit / update of install script

```
svn export https://svn.knmi.nl/svn/TM5/install_tm5
```



Strategies for incomplete checkout

- **Alternative** installation of the install script

  svn co -N https://svn.knmi.nl/svn/TM5/

  pro : can modify / commit / update (**svn -N up**)

  con : "-N" required (but shouldn't... bug?)

- New install script **option**

  "Install .svn directories (yes|no) [no] ?"

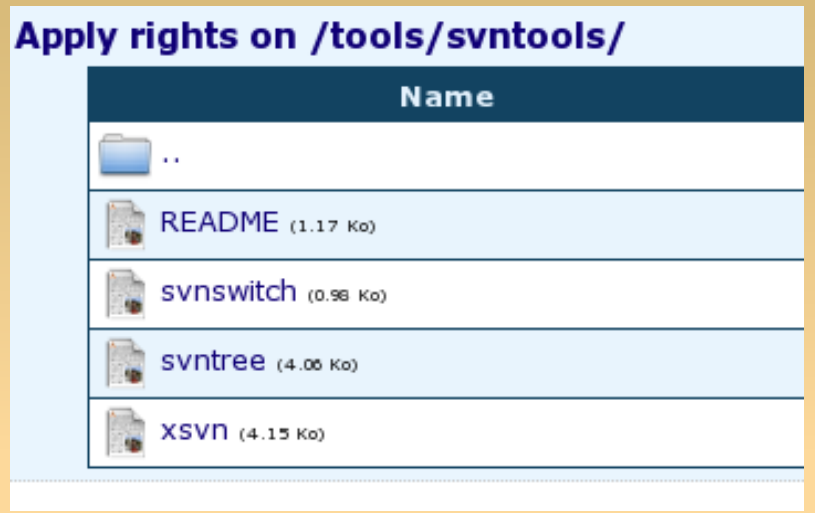
  pro : “svn mkdir MyNewDir” (no need for URL)

  (could use an alias too)

  con : does not link sub-projects

  ... and prevent svntools to do so!

Strategies (cont'd)



Svntools

=> recursive scripts

svntree -c status

svntree -c update

"*svn mkdir / add*" in versioned dir only or thru URL

See README for more advanced usage (xsvn)

Future / ongoing strategies

- improve install script
- additional scripts :

  "TM5-chem package" (demand)

- cheat sheet for TM5 svn
- update documentation on the wiki

Documentation Issues and Priorities

External doc

- **Scientific :** scientific problem and solution adopted - no reference to the code → TM website: tech doc, papers it refers
- **Technical :** include a calling tree, description of all modules, info for testing / modifying / maintaining

  → TM website, technical doc

  (2 yo w/ dead links)
- **User guide** : described user interface, switches, variables -access, default values and range-

External doc

- **Scientific :** scientific problem and solution adopted - no reference to the code → TM website, tech doc
- **Technical :** include a calling tree, description of all modules, info for testing / modifying / maintaining

  → TM website, technical doc

  (2 yo w/ dead links)
- **User guide** : described user interfac… variables -access, default values and …

priority #1 : need a new wiki (before 2011)

Internal doc = Comments

- Tools (ProTex, Sphinx, Doxygen,...)
  - can generate external doc
  - can generate calling tree

- **Header comments**

  are part of coding standards

Coding standards (CS)

- **NOT enough** to "just code in the same language"
- to improve science productivity, CS :
  - reduce common bugs
  - reduce causes of inscrutable software,
  - reduce differences in coding styles [limits legibility]
  - ...

improve maintenance,
portability, readability
-
Exchangeable

Coding standards (CS)

MODELS

- NASA GISS **ModelE** : 14p. "coding conventions" (02/2010)
- **PRISM** "system specification handbook" (02/2003) 20p. Chapter
- ECMWF **IFS** CY28r1 has a 40 pages appendix in its doc (03/2004)
- **GEOS-Chem** : recommendations on website & wiki

ORG.

- Andrews P. (ukmo), G. Cats (knmi/hirlam), D. Dent (ecmwf), M. Gertz, (DWD), J.-L. Ricard (Meteo-France): **European standards** for writing and documenting exchangeable fortran 90 code, version 1.1, 1995.
  http://metoffice.gov.uk/research/nwp/numerical/fortran90/f90_standards.html

LANG.

- PEP 8 → style guide for **Python** (by Guido, see python.org)

ModelE CS

-> use abbrev. only consistent throughout the code
-> use English w/ correct spelling
-> avoid blank or generic terms in variable names (string, ...)
M -> names no longer than 31 characters (F2003)
-> mix-cased for variables
-> file name coincide with content
M -> ext for files: F90 for free format, f or F for fixed format
-> derived type ends with _type, modules with _mod
M -> subset of F2003 (ie works with both intel and gfortran)
M -> discouraged use of obsolete features (entry, arithmetic if,...
-> use private : avoid default public for module data and types
M -> use implicit none
-> limit to 80 columns (not the 132 available)
-> precisely 2 spaces shall be used for each level of indentation

M = mandatory
-
F90 features

ModelE CS (2)

M -> 2 word fortran keywords (except goto): end do, end if, ...
-> spaces around =, +, -
-> no space around * and **
-> 1 space after "," in arguments to proc and functions
-> but no space after "," in arrays indices
-> use lower case for FORTRAN keywords
-> use mixed case for multiword names (numTracers, potentialTemperature)
-> start names with lower case except for derived types and modules

M -> module must have a top-level summary (TAG)
M -> public module entities must be documented (TAG)
M -> procedure dummy variable must be documented (TAG)
M -> public procedure (subroutine or function) must have a top-level summary (TAG)
-> Important/nontrivial local variables should be also be documented

COMMON to ALL MODELS CS !

Internal Doc : ProTeX

set of standard documentation headers (tags, keywords)

!BOP begin of prologue

!MODULE:

!DESCRIPTION:

!INTERFACE:

!REVISION HISTORY:

!EOP end of prologue

Mandatory

Replace !MODULE: with

!ROUTINE:
!IROUTINE: internal routine
!CROUTINE: contained routine
!FUNCTION:

Useful Tags

!USES:
!PUBLIC TYPES:
!PUBLIC MEMBER FUNCTIONS:
!PUBLIC DATA MEMBERS:

!ARGUMENTS:

!INPUT PARAMETERS:
!INPUT/OUTPUT PARAMETERS:
!OUTPUT PARAMETERS:
!RETURN VALUE:

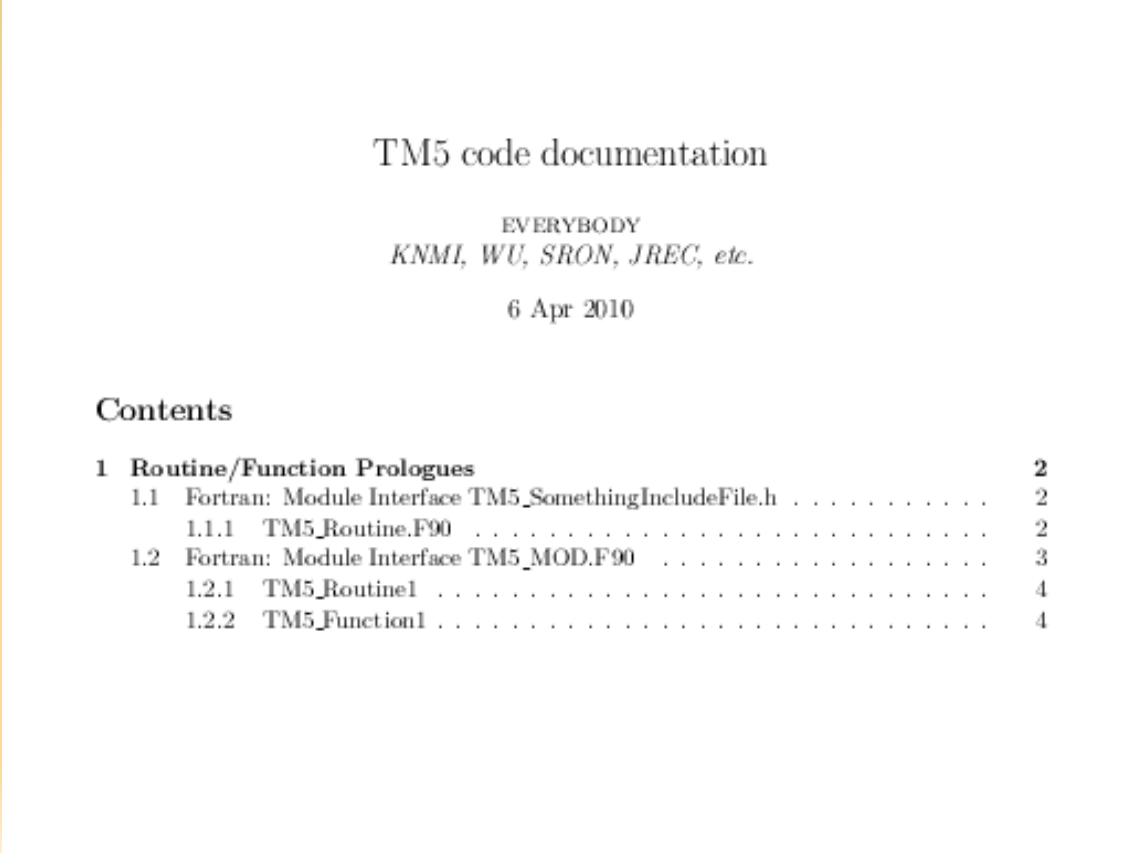
!BUGS:
!SEE ALSO:
!FILES USED:
!REMARKS:
!TO DO:
!CALLING SEQUENCE:
!AUTHOR:
!CALLED FROM:
!LOCAL VARIABLES:

```
!------------------------------------------------------------------------------
!BOP
!
! !IROUTINE:  TM5_Routine1
!
! !DESCRIPTION: This routine does something to the input variable and returns
! the result in the output variable.
!\\
!\\
! !INTERFACE:
!
  SUBROUTINE TM5_Routine1( input, inpout, output, status )
!
! !INPUT PARAMETERS:
!
    INTEGER(ESMF_KIND_I4), INTENT(IN)    :: input   ! Input variable
!
! !INPUT/OUTPUT PARAMETERS:
!
    INTEGER(ESMF_KIND_I4), INTENT(INOUT) :: inpout  ! In/out variable
!
! !OUTPUT PARAMETERS:
!
    INTEGER(ESMF_KIND_I4), INTENT(IN)    :: output  ! Output variable
!
! !REVISION HISTORY:
!   17 Mar 2010 - You - added something
!
! !REMARKS:
! Protex is great!
!
!EOP
!------------------------------------------------------------------------------
!BOC

    !%%% Your code goes here %%%

  END SUBROUTINE TM5_Routine1
!EOC
```

Template routine

Generating Doc from Tag

- Perl script that creates LaTeX source from TAGS (comments)
- <svn>/tools/protex/examples/sample.pdf



TM5 code documentation

EVERYBODY
KNMI, WU, SRON, JREC, etc.

6 Apr 2010

Contents

Extern. Doc

Worth for tools with clean API (self-contained) : GO, Binas, MDF

Can be combined with current tech doc

Low priority

1.2.1 TM5_Routine1

This routine does something to the input variable and returns the result in the output variable.

INTERFACE:

```
SUBROUTINE TM5_Routine1( input, inpout, output, status )
```

INPUT PARAMETERS:

```
INTEGER(ESMF_KIND_I4), INTENT(IN) :: input   ! Input variable
```

INPUT/OUTPUT PARAMETERS:

```
INTEGER(ESMF_KIND_I4), INTENT(INOUT) :: inpout  ! In/out variable
```

OUTPUT PARAMETERS:

```
INTEGER(ESMF_KIND_I4), INTENT(IN) :: output  ! Output variable
```

REVISION HISTORY:

```
17 Mar 2010 - You - added something
```

REMARKS:

```
Protex is great!
```

1.2.2 TM5_Function1

This function does something to the input variable and returns the result in the value variable.

INTERFACE:

```
FUNCTION TM5_Function1( input ) RESULT( value )
```

INPUT PARAMETERS:

```
INTEGER(ESMF_KIND_I4), INTENT(IN) :: input   ! Input variable
```

RETURN VALUE:

```
INTEGER(ESMF_KIND_I4)             :: value   ! Function value
```

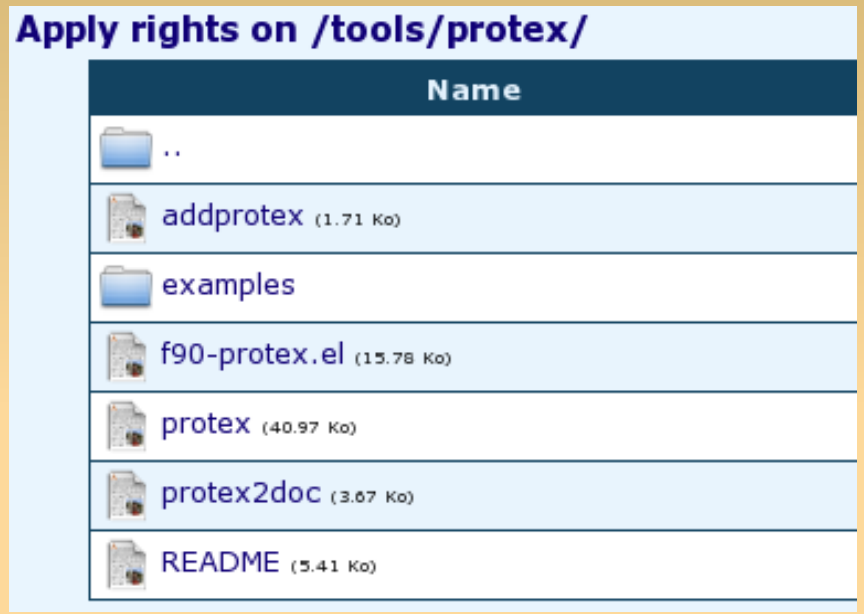
REVISION HISTORY:

```
12 Mar 2010 - You - fixed bug whatever
```

REMARKS:

```
Protex is great!
```

Tools & templates



To **insert tags**, use:

→ templates (examples)
→ addprotex script

addprotex file.F90
(w/ backup)

→ f90-protex.el (emacs)

To **produce doc**:

→ protex2doc (ps, pdf,html)
→ protex (latex)

```
!-------------------------------------------------------------------------
!                    TM5                                                  !
!-------------------------------------------------------------------------
!BOP
!
! !MODULE:       EMISSION_NH3
!
! !DESCRIPTION: performs NH3 emissions needed for TM5 CBM4 version
!\\
!\\
! !INTERFACE:
!
module emission_nh3
!
! !USES:
!
  use dims          , only : nregions
  use global_types  , only : emis_data
  use emission_data, only : bmbcycle, emis2D
  use emission_data, only : msg_emis, do_add_2d, do_add_3d

  implicit none
  private
!
! !PUBLIC MEMBER FUNCTIONS:
!
  public :: declare_emission_nh3   ! initialize
  public :: emission_apply_nh3     ! get emissions
  public :: free_emission_nh3      ! deallocate
!
! !PRIVATE DATA MEMBERS:
!
  type(emis_data),dimension(nregions),target  :: nat_nh3,ant_nh3,bb_nh3
!
! !REVISION HISTORY:
!   18 Jun 2010 - P. Le Sager - added protex tags
!
! !REMARKS:
!
!EOP
!-------------------------------------------------------------------------

contains
```

Module Prologue automatically generated

TM5 Coding Standards ??

Code conception/control → model specific (catch error, tools : GO / mdf, code partition : proj / branches, ...)

F90 language constructs → minimum we could agree on

→ F95 or subset of F2003 (ifort/gfortran)

→ banned "obsolete" features:
entry, arithmetic if, goto, continue, common,...
(use cycle, exit, plain old if, select case, module)

→ use implicit none / private

Style : naming conventions → no need yet

Header doc (LARGE agreement → ProTeX, Doxygen,..)

formatting conventions (indentation, case, spaces,..). Just one:

max line length = 79 (a la PEP 8)

```
                # Walk over the files
                #
                for file in files:
                    #
                    if '__' in file:  # strip project names from F90 files
                        name,ext            = os.path.splitext(file)
                        outfile             = name.split('__')[0]+ext
                    else:
                        outfile             = file

                    if os.path.isfile(os.path.join(sourcedir,file)): # if
listing is a file, add it
                        allfiles[outfile]   = os.path.join(sourcedir,file)
                #
        #
        # Done creating the dictionary of all necessary files, proceed to copy
them
        #
        for target,source in allfiles.iteritems():
            #
            if 'src/' in source: targetdir =
os.path.join(runinfo['compdir'],'src')
            if 'bin/' in source: targetdir =
os.path.join(runinfo['compdir'],'bin')
            if 'py/' in source: targetdir =
os.path.join(runinfo['compdir'],'py')

            targetfile                      = os.path.join(targetdir,target)

            #
            if os.path.exists(targetfile): # if file already exists in rundir,
check time stamp and write to logfile
                #
                srctime  = int(os.path.getmtime(source) )
                tartime  = int(os.path.getmtime(targetfile) )
```

Volcano Ash Simulation

- using TM4 driven by ECMWF winds.
- one uniform source : 21 days lifetime
- Days 1-4 : Continuous release from the surface to ~8 km
- Days 5-10 : " " from the surface to ~5 km

- Model analysis per 3 hours (Period: 14 - 23 April 2010)

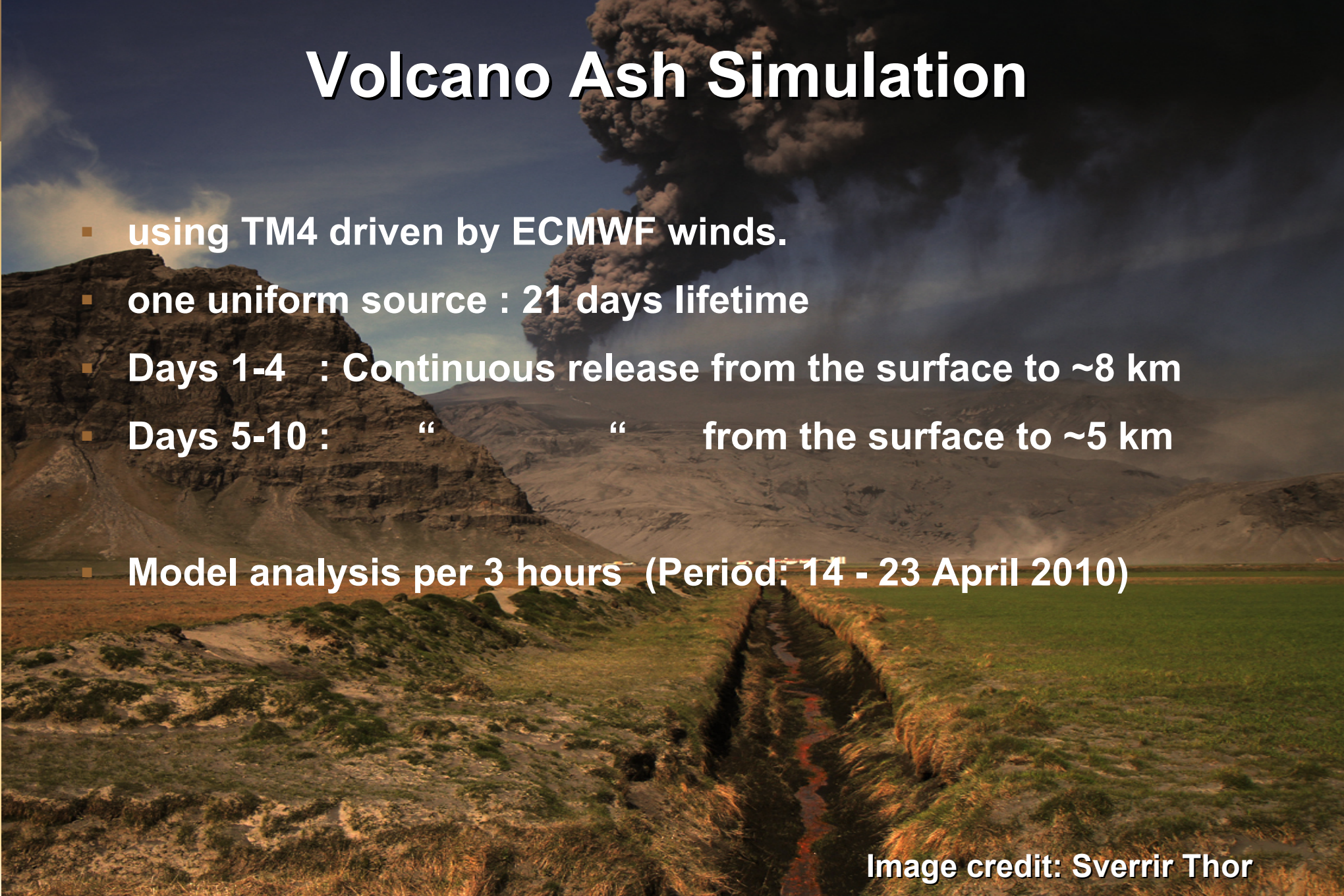


Image credit: Sverrir Thor

Extra Slides

SVN – who's responsible

	In charge	Branches/Trunk/Release
BASE	Arjo	B R T
LEVELS	Arjo	B R T
GRID	Arjo	B R T
USER_OUTPUT		B T
BUDGET		T

	In charge	Branches/Trunk/Release
SIBCASA	Ivar	R T
CHEM_BASE	Philippe	B R T
CHEM_TC	Wouter	B R T
CHEM_M7	Joost	B T
CHEM_ONE	?	T
CHEM_VAR4D	Marteen	B R
C13	Ivar	B T
ENKF	Wouter/Andy	B R T
CHEM_AERO	--obsolete	R T
CHEM_IPCC	--obsolete	R

A simple case

- install:

  svn checkout https://svn.knmi.nl/svn/TM5

- get release 3.0:

  svn checkout https://svn.knmi.nl/svn/TM5/release/3.0

  svn checkout https://svn.knmi.nl/svn/TM5/ -r 3304

  svn checkout https://svn.knmi.nl/svn/TM5/ -r {2010-06-07}

- simplified (no need for URL, recursive)

  svn up / st / add / ci / mkdir

One liners

CS = mandatory or encouraged rules... almost against free-format

- **ECMWF CS = 128 rules into four categories:**

(1) control of the code (2) conception of the code

- standardize error handling
- convention on output unit messages
- unique file to store universal constants
- code partition : e.g. project/branches in TM5

(3) presentation of the code (4) respect of the norm

PRES(01):

"Executable lines should be written using upper case characters"

CTRL(12)

"Conventional prefixes are recommended for names."

POPULAR !

Prefix	Entities	Suffix
TYPE_	Types names	
TYPE_	Types definitions modules	S
PAR	Parameters modules	
YOE	Data modules specific to ECMWF physics	
QA	Data modules specific to CANARI	
YEM	Data modules specific to ALADIN	
TPM_	Data modules specific to spectral transforms packages	
MPL_	Data modules specific to MPL (message passing) package	
YOM	Data modules not specific to ECMWF physics, CANARI, ALADIN, spectral transforms or MPL package	
	Procedure modules	_MOD
NAE	Namelists specific to ECMWF physics	
NEM	Namelists specific to ALADIN	
NAC	Namelists specific to CANARI	
NAM	Namelists not specific to ECMWF physics, ALADIN or CANARI	
SUEC	Setup procedures specific to ECMWF physics	
SUE, not SUEC	Setup procedures specific to ALADIN	
SU, not SUE	Setup procedures not specific to ECMWF physics or ALADIN	
SL	Calculation procedures for any horizontal interpolations system	
LA	Calculation procedures specific to the semi-lagrangian scheme	
AC	Calculation procedures specific to ARPEGE/ALADIN physics ("Arpege Calcul")	
PP	Calculation operators for the post-processing or the analysis	
FP	Procedures specific to *FullPos*	
CA	Procedures specific to CANARI	
FA	Procedures specific to the Files Arpege package (FA)	
LFI	Procedures specific to the Indexed Files Library (LFI)	
MPL_	Procedures specific to the Message Passing Library (MPL)	
SI	Procedures specific to the semi-implicit scheme	
GNH	Procedures specific to non-hydrostatic gridpoint calculations	
CP or GP	Non-specific gridpoint calculation procedures	
SP	Spectral calculation procedures	
COMM, GATH, ISND, IRCV, OSND, ORCV, BR, DI or TR	Procedures dealing with inter-nodes communications ("COMMunicate", "GATHer", "Input SeND", "Input ReCeiVe", "Output SeND", "Output ReCeiVe", "BRoadcast", "DIstribute", "TRanspose")	
RE or RD	Procedures to read data	
WR	Procedures to write data	
E	Procedures specific to ALADIN ("Elliptic")	
	Tangent linear of a procedure	TL
	Adjoint of a procedure	AD
	Inverse of a procedure	IN