The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2148-6

Fifth ICTP Workshop on the Theory and Use of Regional Climate Models

31 May - 11 June, 2010

New computational aspects of RegCM4

E. Coppola
ESP-ICTP
Trieste
ITALY

S. Cozzini
SISSA
Trieste
ITALY

G. Giuliani and M. Scarcia
CETEMPS, Dipartimento di Fisica
Universita' degli Studi di L'Aquila
L'Aquila

Fifth ICTP Workshop on
Theory and Use of Regional
Climate Models

Unveiling RegCM-4.0-rc1

Erika Coppola, Stefano Cozzini, Graziano Giuliani, Martin Scarcia

Democritos and SISSA/eLAB – Trieste University of L'Aquila, Cetemps -L'Aquila

Outline

- Introduction: why RegCm4 ?
  - Work done on Regcm4
- The New Features of Regcm4
  - Where is Regcm4 ?
  - The package
  - Installing
  - Using
  - Modifying
  - Future Activities

Goals of hands-on sessions of the workshop

- For the audience
  - Learn how to use it
- For us
  - Receive feedback from audience about the work done so far
  - Get suggestions for future activities
  - Find a lot of bugs
    - Bug contest the best (or worst) bug reporter will earn a beer (or coffee)
  - Finalize the officially release of RegCM-4.0 by the end of workshop

Why Regcm4 ?

- Need of simple and more user friendly package
  - For users
  - For developers
- Enhancement in the physics of the model:
  - Insert new features to allow better simulation
- Need of integrated environment for managing the software project
  - Revision system
  - Bug tracking etc..

RegCM4: a software reengineering project

- Refactoring of Model Code in F90 ANSI standard to use new Fortran features (modules, types and dynamic allocation mostly)
  - Get rid of commons.
  - Isolate access to data and use automatic memory allocation.
  - Avoid need of a model recompile from code for just a dimension change.
- Repackaging of the whole project
- Use state of the art Software engineering Tools

How we developed RegCM4

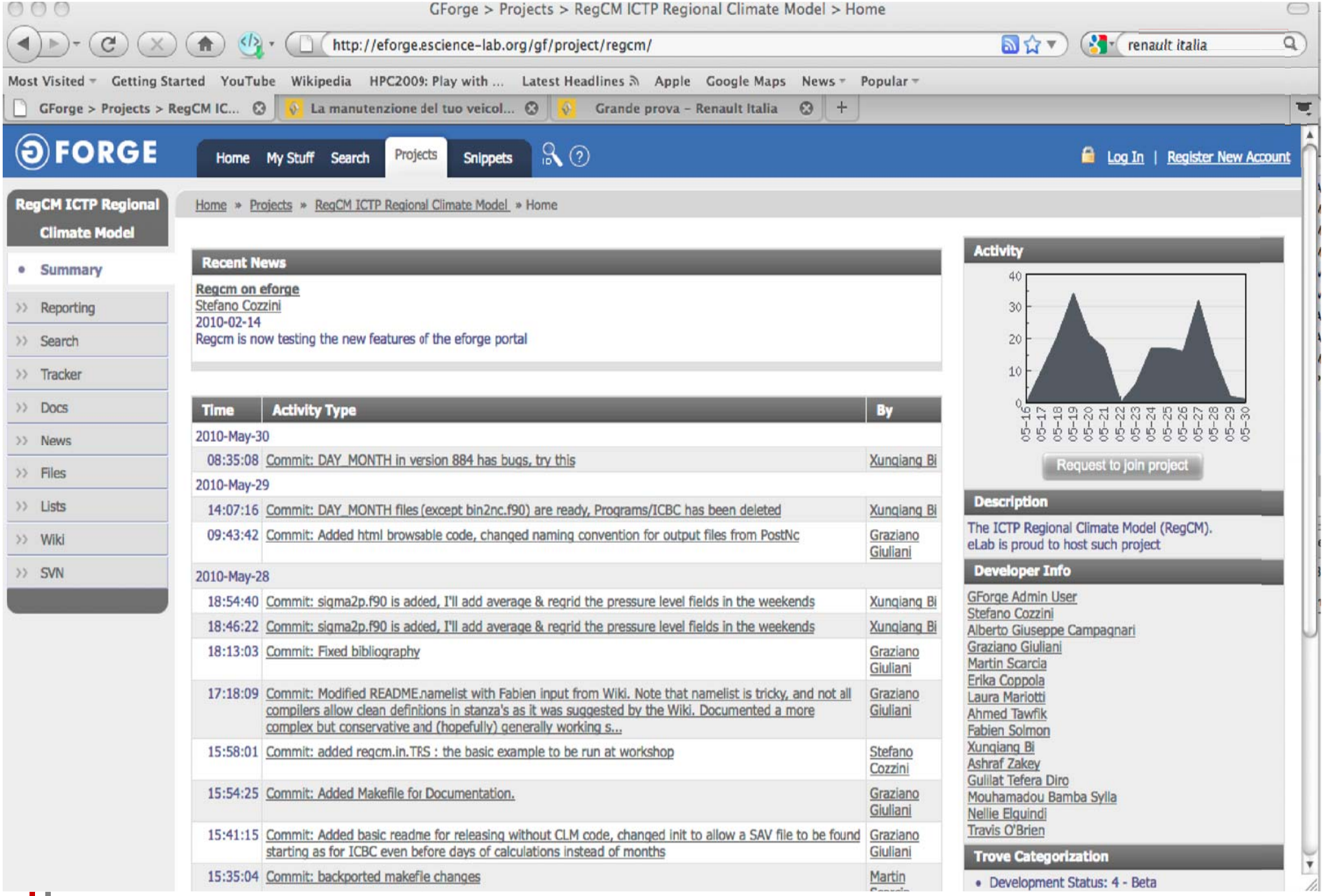
- Branch the code repository at T=0 (jan 2010) and identify regression tests for the new development to be successfully accepted.
- Worked on the new code with scheduled milestones and deadlines for each major identified model component
  - Identified code "maintainers" which review new code
  - Get approval for each major design change in developer meetings
  - Reach as fast as possible a working implementation to test improvements in all the part of the model
- Perform non-regression tests before reaching Time of release
- Backport any new model improvement implemented in the timeframe
- Identify and solve model "instabilities" tracked during development.

THIS IS THE FUTURE DEVELOPING MODEL AS WELL

Where is the RegCm4 package ?

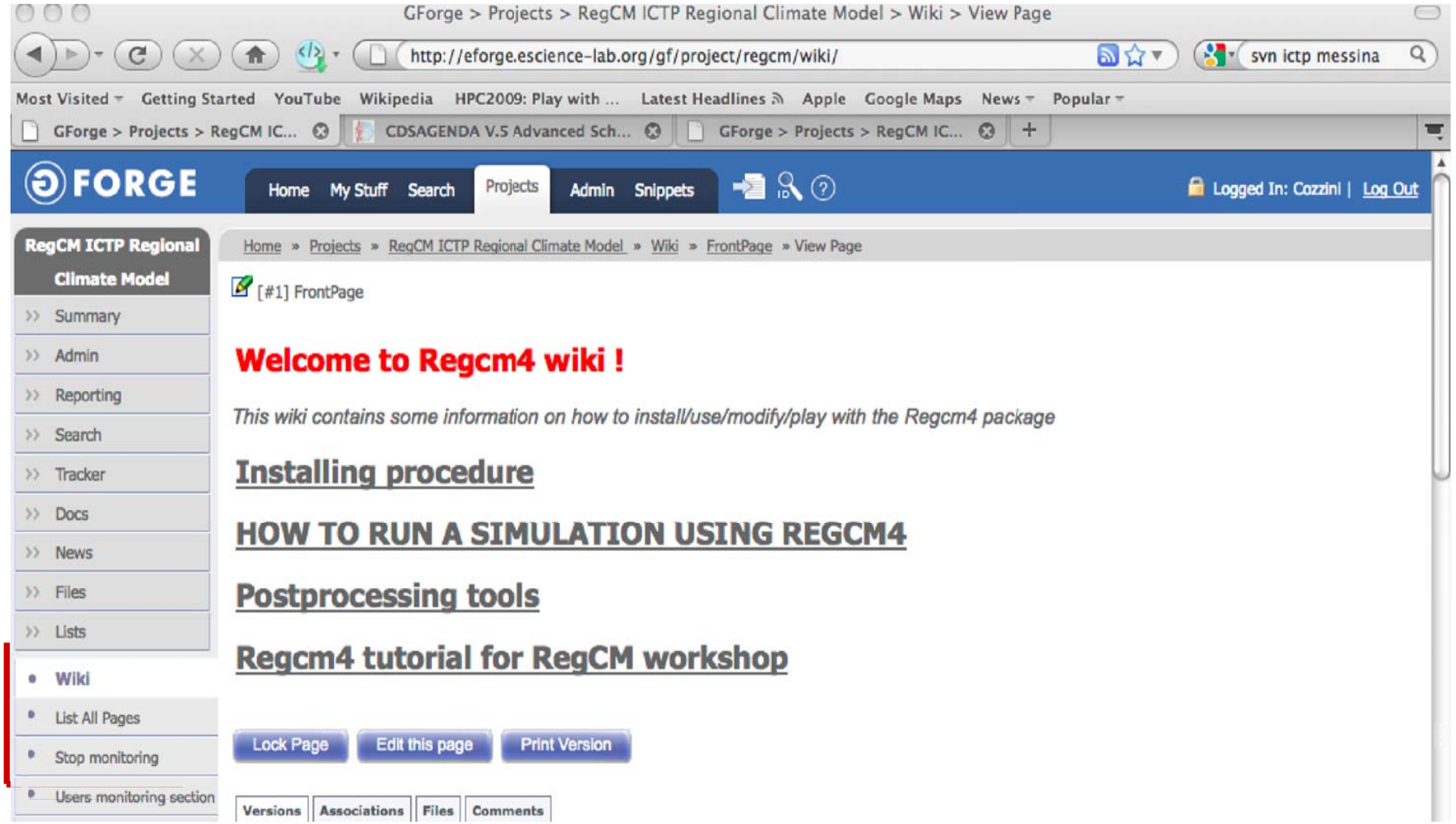
- http://eforge.escience-lab.org/gf/project/regcm/

- eforge portal:
  - This forge is an integrated set of tools or components that facilitates collaboration on sofware project. It provides tools for version control, bug tracking, task management, and tools for communication (newsgroups, web pages, wiki, mailing lists, etc.). A forge makes scientific technical and scientific collaboration easier. The objective is to provide an infrastructure for scientific-technical collaborations on software project

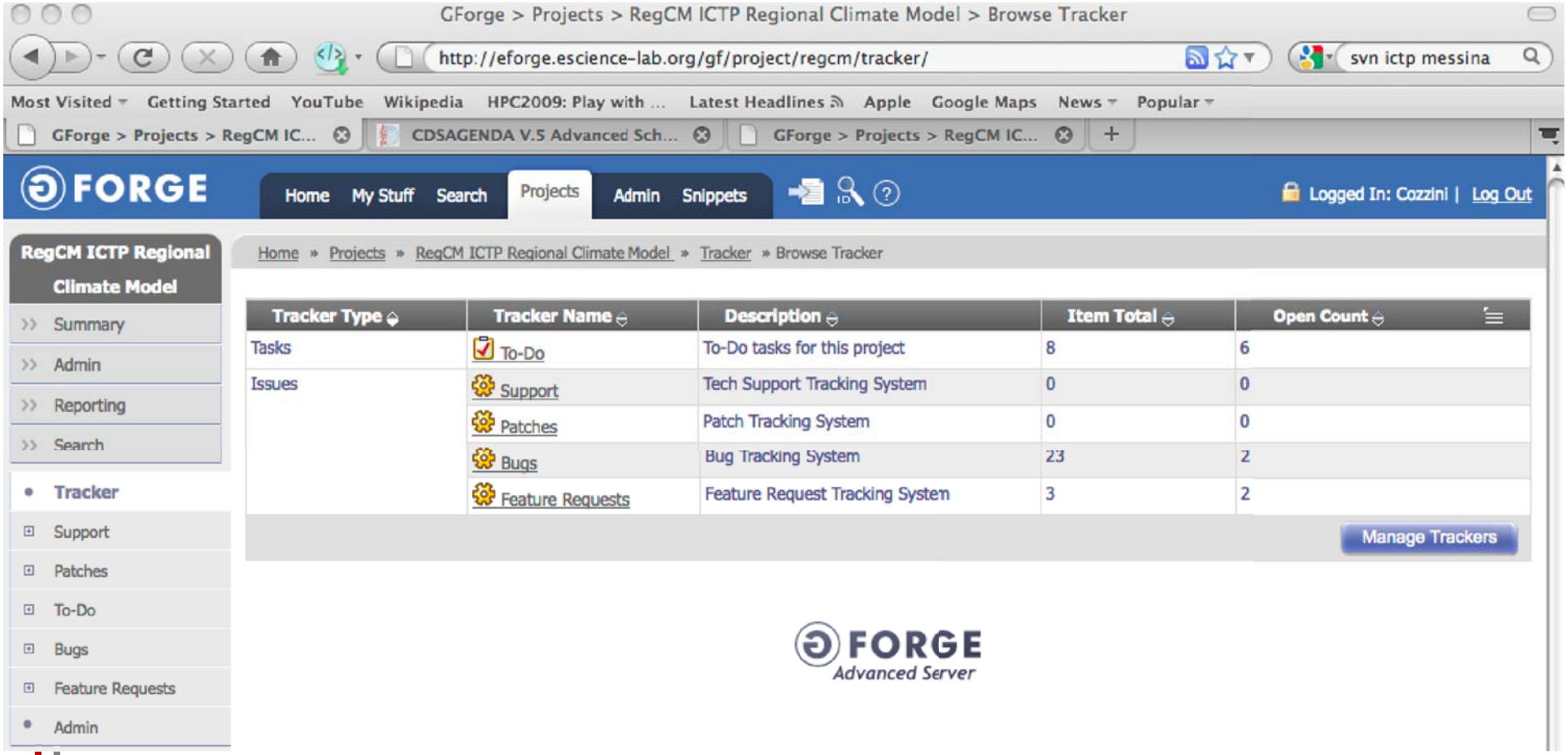

GForge > Projects > RegCM ICTP Regional Climate Model > Home
http://eforge.escience-lab.org/gf/project/regcm/
renault italia
Most Visited
Getting Started
YouTube
Wikipedia
HPC2009: Play with ...
Latest Headlines
Apple
Google Maps
News
Popular
GForge > Projects > RegCM IC...
La manutenzione del tuo veicol...
Grande prova – Renault Italia
FORGE
Home
My Stuff
Search
Projects
Snippets
Log In | Register New Account
RegCM ICTP Regional Climate Model
Summary
Reporting
Search
Tracker
Docs
News
Files
Lists
Wiki
SVN
Home » Projects » RegCM ICTP Regional Climate Model » Home
Recent News
Regcm on eforge
Stefano Cozzini
2010-02-14
Regcm is now testing the new features of the eforge portal
Time
Activity Type
By
2010-May-30
08:35:08 Commit: DAY_MONTH in version 884 has bugs, try this Xunqiang Bi
2010-May-29
14:07:16 Commit: DAY_MONTH files (except bin2nc.f90) are ready, Programs/ICBC has been deleted Xunqiang Bi
09:43:42 Commit: Added html browsable code, changed naming convention for output files from PostNc Graziano Giuliani
2010-May-28
18:54:40 Commit: sigma2p.f90 is added, I'll add average & regrid the pressure level fields in the weekends Xunqiang Bi
18:46:22 Commit: sigma2p.f90 is added, I'll add average & regrid the pressure level fields in the weekends Xunqiang Bi
18:13:03 Commit: Fixed bibliography Graziano Giuliani
17:18:09 Commit: Modified README.namelist with Fabien input from Wiki. Note that namelist is tricky, and not all compilers allow clean definitions in stanza's as it was suggested by the Wiki. Documented a more complex but conservative and (hopefully) generally working s... Graziano Giuliani
15:58:01 Commit: added regcm.in.TRS : the basic example to be run at workshop Stefano Cozzini
15:54:25 Commit: Added Makefile for Documentation. Graziano Giuliani
15:41:15 Commit: Added basic readme for releasing without CLM code, changed init to allow a SAV file to be found starting as for ICBC even before days of calculations instead of months Graziano Giuliani
15:35:04 Commit: backported makefile changes Martin
Activity
40
30
20
10
0
05-16 05-17 05-18 05-19 05-20 05-21 05-22 05-23 05-24 05-25 05-26 05-27 05-28 05-29 05-30
Request to join project
Description
The ICTP Regional Climate Model (RegCM).
eLab is proud to host such project
Developer Info
GForge Admin User
Stefano Cozzini
Alberto Giuseppe Campagnari
Graziano Giuliani
Martin Scarcia
Erika Coppola
Laura Mariotti
Ahmed Tawfik
Fabien Solmon
Xunqiang Bi
Ashraf Zakey
Gulilat Tefera Diro
Mouhamadou Bamba Sylla
Nellie Elguindi
Travis O'Brien
Trove Categorization
Development Status: 4 - Beta

Documentation: WIKI on eforge

- eforge.escience-lab.org/gf/project/regm/wiki



Bug tracking and reporting..



Get the Regcm4 package

- Regcm4 is distributed as source code through access to svn server:
- Everybody can get it but only developers can modify it
- It is now available regCM v. 4.0 release candidate 1 (tagged RegCM-4.0-rc1)
- To get it you need svn client (available on all Linux Boxes by default)
- Info on svn:
  - http://svnbook.red-bean.com/

The svn command to get the code:

```
$ svn checkout --username anonymous \
https://eforge.escience-
lab.org/svn/regcm/tags/RegCM-4.0-rc1
Authentication realm: <https://eforge.escience-
lab.org:443> Document repository
Password for 'anonymous':

A    RegCM-4.0-rc1/Tools
A    RegCM-4.0-rc1/Tools/Scripts
A    RegCM-4.0-rc1/Tools/Scripts/GrADS
...
 U   RegCM-4.0-rc1
Checked out revision 867.
$
```

PASSWORD IS: ANONYMOUS

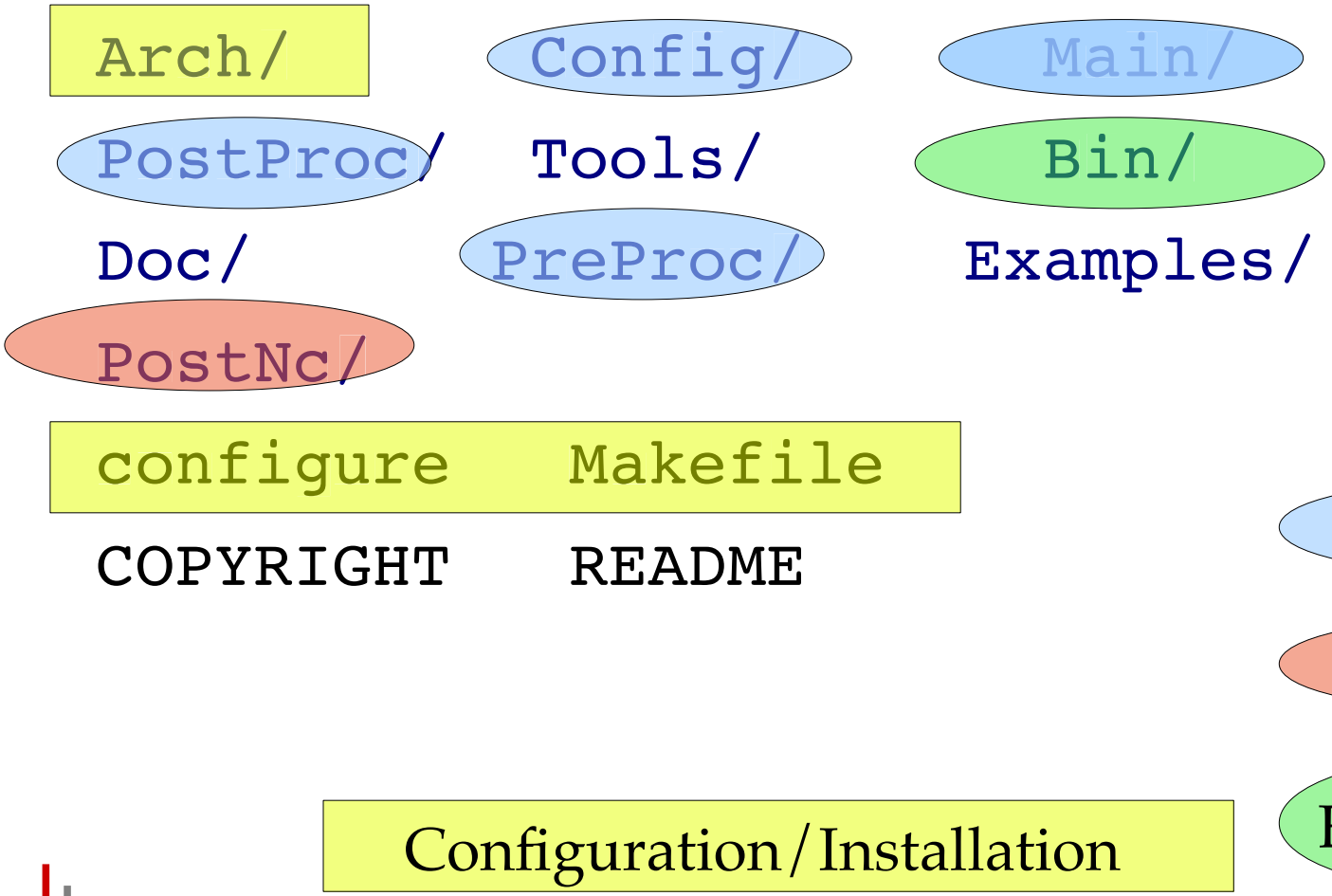
The Regcm4 package

- RegCM-4.0-rc1 >ls

```
Arch/       Config/      Main/
PostProc/   Tools/       Bin/
Doc/        Makefile     PreProc/
configure   COPYRIGHT    Examples/
PostNc/     README
```

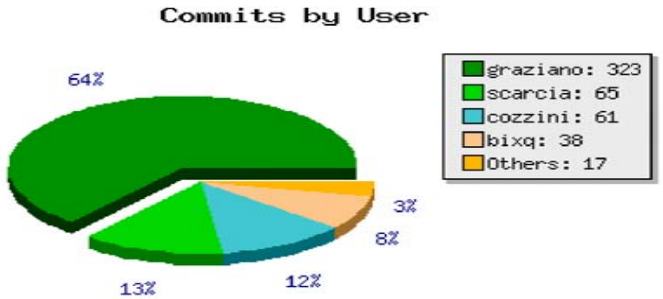
```
>cat README

This is regcm version 4.0 release candidate 1
(RegCM-4.0-rc1)

distributed at the fifth ICTP workshop  (june
2010)

To know more about the project visit:
http://eforge.escience-lab.org/gf/project/regcm
```

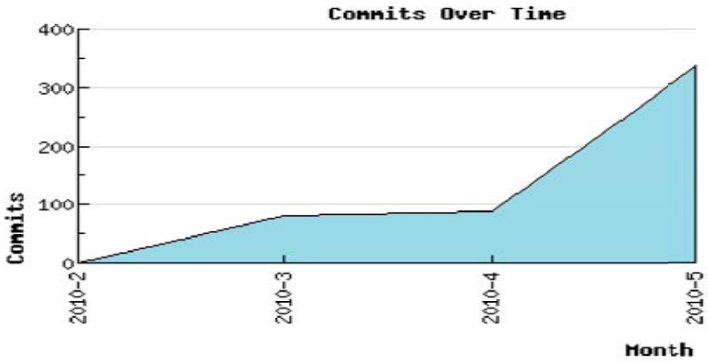
How is the package organized ?



Size&activity of the package:

- RegCMv3 : ~48000 lines of f77 code with some f90 extensions
- RegCMv4-rc1 : ~75000 lines of ANSI f90 code + 8000 lines of ANSI C++ code





FORGE Advanced Server

New features on code side (1)

- No more COMMON blocks - use data section of MODULE
- MODULE encapsulation of subroutines and functions (TBC for Main)
- Smart(er) control of I/O files, no "implicit" fortran units
- ONE place for all physical constants:

  Config/mod_constants.F90

New features on code side (2)

- Dynamic allocation
  - COMPILE ONCE RUN FOR ALL THE DOMAIN
  - SIMULATION ARE INDEPENDENT FROM EXECUTABLES
- ONE configuration namelist to rule all the executables and steps
- Postprocessing to convert to NetCDF CF-1.4 compliant format (PostNc)

What do you need to install Regcm4

- Netcdf library:
  - http://www.unidata.ucar.edu/software/netcdf/
- A compiler (An ansi F90 compiler && C++):
  - Gfortran /g95 +g++ (freely available for Linux box)
  - Intel (free available for scientific personal usage)
  - PGI (not free)
- Python interpreter (available)
- MPI library for parallel version
  - Recommended : openmpi (www.open-mpi.org)

Special Lab session next week to setup a RegCM simulation Box

New Physical Features

- SST : added
- SEA ICE
- CLM coupling
- See tutorial how to use these features starting from Wednesday

SST daily cycle

RegCM4 contains a option which allows for the computation of sea surface skin temperature as a prognostic variable following the scheme of Zeng (2005). This allows for a realistic representation of the diurnal sea surface skin temperature, leading to improvements in the surface fluxes thus air-sea interactions. The scheme is based on a two-layer model which includes warm layer/cool skin effects as described by Fairall (1996). Temperatures in the two layers are calculated using a one-dimensional heat transfer equation and boundary conditions determined by surface to atmosphere fluxes (latent, sensible and radiative) and a 3~m depth sea surface temperature taken from the prescribed SSTs.

Reactivate Seaice scheme in BATS

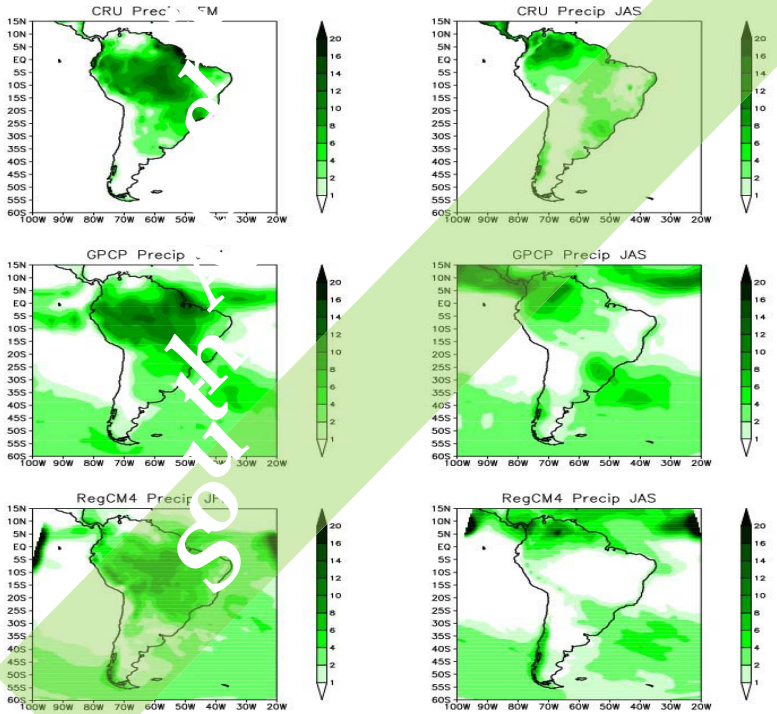
- Read the NOAA Optimum Interpolation (OI) Seaice coverage data at ICBC step (by use SSTTYP='OI_WK' or 'OI_ST');
- Assign the albedo value for those seaice grids (albedov.F90);
- Initialize the seaice thickness as 1 meter (bdyin.F90);
- Following the macro define of SEAICE, you can trace back all the required modifications.

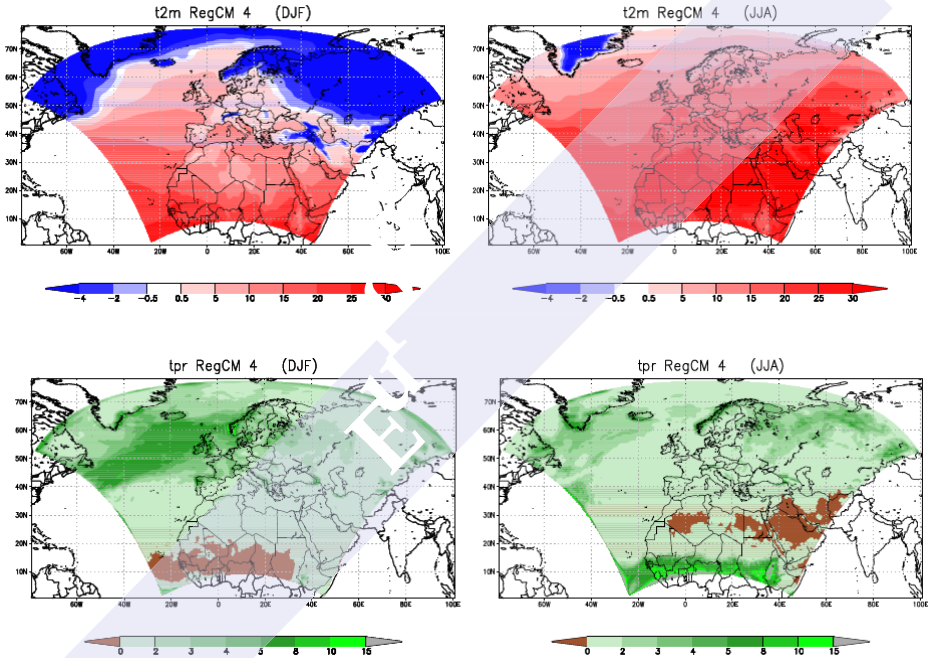
CLM land surface scheme

CLM improvements

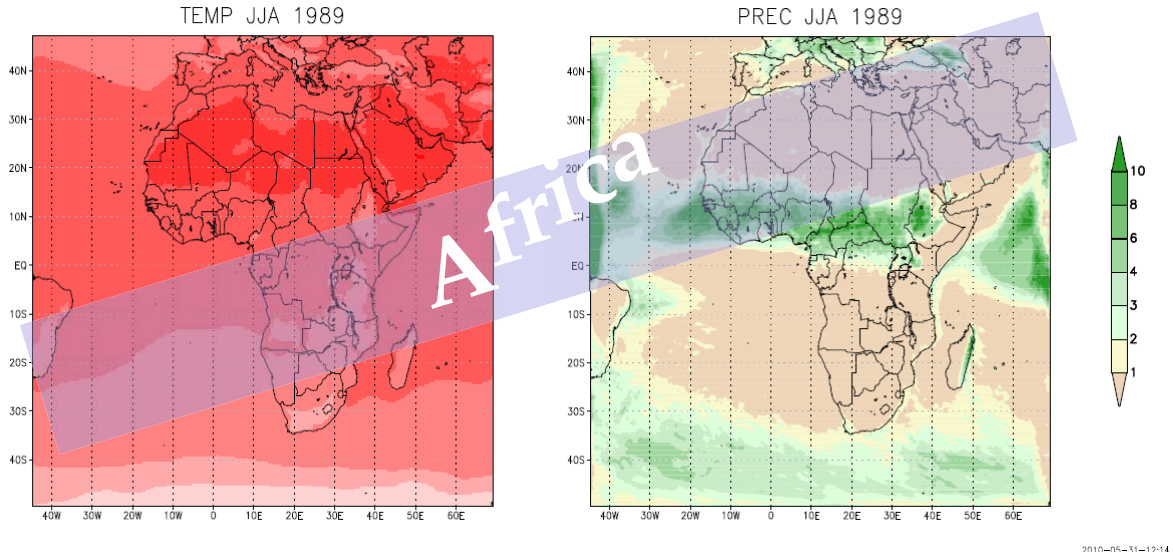
- 10 unevenly spaced soil layers solved for temperature/hydrology explicitly
- 5 snow layers
- 1 vegetation layer with 15 vegetation categories
- 3 unique land unit types (glacier, wetland, lake)
- Divides canopy into sunlit and shaded
- Tile mosaic method
- Zeng Lake Model

RegCM4 examples domains (in Example dir)





RegCM4 examples domains (2)



Benchmarking: v3 vs v4

- Preliminary results:
  - European Domain 160x192 1 month simulation
  - 4 CPUS on Opteron 2.2Ghz
- Regcm4 : 6117 seconds
- Regmc3: 5758 seconds

Reg4 ~ 6% slower than RegCm3
on short time simulation
Lot of room to improve

Future activities from computational side

- Insert new MPI parallel coding (2D decomposition)
- Complete clean-up of the code and distribution
- Complete modularizing the regmc code (Main directory)
- Add data types in the code
- Project a new I/O layer for the Model
- Increase performance

What is going to be obsoleted

- USGS input with 25 Landuse categories
- Some input modules to ICBC (relative to older GCM)
- Some old tools
- Anything the audience of this workshop think it is useless

Improve parallelism of RegCM MPI model code

- **Regcm-3 uses simple broadcast/scatter/gather for a total of ~250 calls to the library (except for ABORT, INIT, FINALIZE or BARRIER) in about 15 fortran source files:**

bdyin.F bdyuv.F chsrfem.F conadv.F conmas.F holtbl.F htdiff.F output.F param.F regcm.F splitf.F spstep.F tend.F tracbud.F tracdiag.F

  - **Any single parameter needs to be broadcasted**
  - **REGCM-4.0 limits parallelism to the j index limiting maximum number of processors to the maximum number of points in that direction (jx)**

What should be done for more scalability

- Use MPI_CART routines to implement a Cartesian decomposition of the model domain.
  - Standard implementation which arranges model domain among all processes
- Communicate using dedicated type all parameters with just a single MPI broadcast call

Documentation

- Create a Regcm4 FAQ
  - Frequently Asked Questions
- Increase the quality of documents and tutorials
- Prepare/setup tools for tutorials for self-learning approach

Who should do this ?

A Community effort:
do not ask what RegCm can do for you
But ask what you can do for regCM

Tips for Regmc3 Users:

- Recompile the model ONLY if changed the core code:
  - NO RECOMPILATION NEEDED UPON CHANGING THE NAMELIST regcm.in.
- ALWAYS keep the namelist input file TOGETHER with the output:
  - it is a part of the model I/O and identify the simulation. The C++ PostNc needs it to produce the output (informations about the simulation are included in the NetCDF files).

Tips for new users:

- Use new feature like domname, terdir, globdir, outdir to isolate different experiments.
- Examinate carefully the PreProc output BEFORE launching a computational intensive model run: GARBAGE in -> GARBAGE out

For developers: what if ..

- .. Want to add some more input parameters:
  Add a variable or a full stanza to regcm.in namelist and:
  - IF this is GLOBAL to all model pieces (PreProc+Main) modify eventually mod_dynparam.F90 in the Config directory to add the variable(s) and add if needed the broadcast in the initparam subroutine.
  - IF just for Main modify as usually the param.F90 to read and if needed broadcast the variable(s) or namelist stanza.

For developers: what if ..

.. Want to add a vector/matrix with domain dimensions (IY,JX,KZ)

- Identify which MODULE to place/needs the variable
- Declare the variable as ALLOCATABLE in the DATA section of MODULE
- Add the ALLOCATE line in the allocate_mod_xxxxx subroutine

For developers: what if ..

- Want to add a new subroutine/function:
  - To an existing/already identified MODULE, i.e. MYSUB into mod_XXXX.F90
  - Add the subroutine in the MODULE after the CONTAINS line
  - -Add to the caller the line USE mod_XXXX , only : MYSUB
  - Add the dependency if not already there in the Makefile between the calling object and the module object
  - The compiler will issue an error/warning if number and types of arguments will not match between module and subroutine

For developers what if..

- Add from scratch a new MYSUB
  - Create a new module, i.e. mod_mymod.F90
  - Create data section (if needed)
  - If domain dimension function space
    - Add variable with ALLOCATABLE attribute
    - Add allocate_mod_mymod subroutine with allocate line
    - Add the CALL allocate_mod_mymod into param.F90
  - Add the CONTAINS line
    - Write after that Your code
    - Add the caller the line USE mod_mymod , only : MYSUB line
    - Edit the makefile and add mod_mymod.F90 to MODOBJS
    - Add at bottom the line with dependencies for mod_mymod.o
    - Add the dependency to mod_mymod.o for the caller object