A Single Column Model of HARMONIE in KNMI Parametrization Testbed (CY33T1 version)

Sylvie Malardel
Météo-France/CNRM
sylvie.malardel@meteo.fr
Cisco de Bruijn and Wim de Rooy
cisco.de.bruijn@knmi.nl, wim.de.rooy@knmi.nl
KNMI/Weather-Research

9 september 2010

Chapter 1

General description

1.1 Introduction

Most of the NWP or research 3D models have a corresponding single column version dedicated to the validation of physical parametrisations. In most cases, these single column models are « off-line » codes with a simple dynamical part (data flow, forcings) which calls the physics/dynamics interface of the corresponding 3D model.

In the context of the collaboration between Météo-France and the Aladin and Hirlam consortiums and the multiplications of the sets of physics which will be available in the Arpège/IFS system, it may be interesting for all the community to have a common 1D tool for physics validations and intercomparisons.

The simplest way to automaticaly benefit of all the evolutions of the Arpège/IFS code would be to use an "embedded" 1D version of the model. Such a version could be systematicaly maintained, validated and distributed with the 3D (and 2D) versions of the code.

1.2 A pseudo-1D model

A 2D meridional vertical plane version is available in Arpège/Aladin for academic cases and it is validated for each official cycle of Arpège/IFS. This 2D version is used with a minimal configuration as a single column model.

The 2D version of Arpège/Aladin is a particular configuration of Aladin (limited area model, LELAM=T) with some modification to have a cartesian geometry (LMAP=F) and a zonal wavenumber NMSMAX=0 with only one grid point in the x-direction (NDLON=1).

In theory, a « strict » single column model would correspond to a meridional wavenumber NSMAX=0 and only one point in the meridional direction (NDGL=1). But such a configuration was more problematic than a pseudo-1D configuration with NSMAX=1 and 4 identical columns (NDGL=4).

With this pseudo-1D configuration, the spectral computations are still done (by the way, it may be a simple way to check the code) but the derivatives are zero. If the Coriolis

parameter is set to zero (the default in cartesian geometry), if the physics is switch off and if there is no forcing, any set of four identical profils remains stationnary. If the physics and the forcings are applied identicaly to the four columns, they remain identical (this is checked for semi-lagrangian and eulerian dynamics in 2 or 3 time levels).

1.3 A case database for SCUM

Chapter 2

SCUM documentation

2.1 Introduction

This chapter documents the procedure to be followed in order to run an ideal case (as described in an intercomparaison experiment for exemple) with SCUM. We suppose that the FA file containing the initial profil and the forcing was already prepared (see chapter 3 for informations on ACADFA1D, a basic tool to help for the 1D case preparation).

2.2 Starting

2.2.1 Starting from a "ready" 1D namelist

If the case you want to run was already prepared but ran with an other set of physics, it would be easier to take the namelist used with the other physics and adapt it to your own one.

2.2.2 Starting from a 3D namelist

If you want to run a 1D case with the physics you're currently using in 3D, the following sections gives the necessary modifications which should be done in your 3D namelist to be able to run a 1D case with this physics.

2.2.3 Starting from scratch

If you're used to work with Arpege/IFS, you should be able to built your own namelist with the information found in the following sections. If not, please call me (00 33 (0)5 61 07 96 34 or send me an email (sylvie.malardel@meteo.fr) for help.

2.3 Setting of the pseudo-1D geometry

The setups for limited area (LELAM=T) and cartesian (LMAP=F) are directly known trough the 1D FA file.

In the namelist you have to setup the parameter **NPROMA** in **NAMDIM** to -4 (the minus to impose your choice of NPROMA (if you set up NPROMA to 4, your choice will be submitted to an optimised computation of NPROMA and then it may be changed depending on the machine)). For the pseudo-1D model, the total number of columns is 4, then you will have only one block of NPROMA columns for your only processor (this configuration is the only one validated for 1D case, but other configurations may work too...)

In the namelist **NAMDIM** you may specify the number of vertical level but it is not compulsory (known from the FA initial fine).

2.4 Setting of the pseudo-1D "dynamics"

In 1D, there is generally no reason to run with NH-dynamics. So, SCUM is validated only for hydrostatic dynamics (LNHDYN=.FALSE.). But, the NH version of the dynamics (LNHDYN=.FALSE.) should give nearly identical results (a good way of validating the NH code?).

You can run SCUM in eulerian with 3 time levels (leap frog) or semi-lagrangian with 2 or 3 time levels. The type of scheme (eul or sli) is usually chosen in the execution command line in your running script. If you are running in eulerian, only 3TL is available (the value of LTWOTL in NAMCT0 is not used). But if you are running in semi-lagrangian, you have to specify **LTWOTL=.TRUE.** in **NAMCT0** if you want to run in 2TL. Some simulations were done do test these different options with SCUM, and the results were very similar with all options (except some problems in eulerian probably in the advection of the rain in CY30T1, problems which were not present in CY29T2).

The other setups for the dynamics are mainly in **NAMDYN**. They concern essentially the horizontal diffusion and the semi-implicit.

2.5 Technical setups

SCUM is validated only in monoproc. To run in monoproc, you have to specify **NPROC=1** in **NAMPAR0**. et le reste de NAMPAR0 et NAMPAR1 ?

2.6 Large scale atmospheric forcing

2.6.1 Setting of the atmospheric forcing in namelist

The most commom atmospheric forcings used in 1D cases a implemented in SCUM (more will be added in next cycles). Even if they are time dependant, all the forcings have to be present in the initial FA file (see chapter 3 for more details). In the model, the forcing fields are known in the data flow trough a dedicated "GFL" structure that we will call here GFL%YFORC (see annexe **??** if you do not know what a GFL is).

The main switch **LSFORC** in **NAMCT0** has to be set to .TRUE. to activate the large scale forcings

GFL%YFORC is an array of size (NGPTOT, NFLEVG, NGFL_FORC) where NGFL_FORC is the total number of forcing. **NGFL_FORC** has to be specified in the namelist **NAMGFL**. In the default, the attributs of GFL%YFORC specify that forcings are grid point fields, that they have to be read in the input file, but not written in the output files. The forcing are not treated by the model time stepping (not present in GFL for T9 or T1).

So, in the namelist **NAMGFL**, you have to specify the total number of forcing in **NGFL_FORC** (default is 0) and the name of each of this forcing in **YFORC_NL(i)%CNAME** where i is the number of the forcing. The names specified in **YFORC_NL(i)%CNAME** have to be the same that the ones chosen when you create your initial FA file. You will find below and exemple of **NAMGFL** in the case of 3 forcings which are known in the FA file as FORC01, FORC02 and FORC03 :

```
&NAMGFL
 NGFL_FORC=3,
 YFORC_NL(1)%CNAME='FORC01',
 YFORC_NL(2)%CNAME='FORC02',
 YFORC_NL(3)%CNAME='FORC03',
/
```

The detail of the fields in GFL%YFORC is decribed in the namelist NAMLSFORC. This namelist contains the following parameters :

- LGEOST_UV_FRC = T to integrate a geostrophic forcing (with a constant Coriolis parameter)
- RCORIO_FORC = f_plan value of the Coriolis parameter used in the geostrophic forcing. Default is RCORIO_FORC=10-4 s^{-1}.
- LT_ADV_FRC = T to apply a large scale temperature advection on the temperature field
- LQV_ADV_FRC = T to apply a large scale specific humidity advection on the specific humidity field

- LSW_FRC = to simulate a large scale vertical advection (specified with $w = Dz/Dt$ in m/s) of all the pronostic fields

These main switches are completed by a set of 3 parameters for each activated forcing. For exemple, if the large scale vertical advection is activated (LSW_FRC = T), the frequency (in second) between two forcing profils, the index in the GFL%EXT structure of the first vertical velocity profil and the number of vertical velocity profils have to be given in NAMLSFORC by :

- NLSW_TIME : time step between 2 forcing times (in seconds)
- NLSW_DEB : index of the first forcing LSW in GFL%EXT
- NLSW_NUM : total number of forcing of the type LSW (if NLSW_NUM=1 a constant forcing is applied)

In this exemple, GFL%YFORC(NLSW_DEB) is the first profil of w (time 00 of the simulation) in the GFL%YFORC structure, GFL%YFORC(NLSW_DEB+1) is the second profil of w (NLSW_TIME after the beginning of the simulation) and GFL%YFORC(NLSW_DEB+NLSW_NUM is the last profil of w (NLSW_TIME*NLSW_NUM after the begining of the simulation). In practice, a linear interpolation of the forcing is computed at each time step between two forcing times. Note that if NLSW_NUM=1 a constant forcing is applied during all the simulation. The following example described the contains of a GFL%YFORC with 19 forcings (NGFL_FORC=19) which are available for a simulation of 2 hours.

```
&NAMLSFORC
LGEOST_UV_FRC=T,
RCORIO_FORC=1.E-4,
NGEOST_U_DEB=1,
NGEOST_U_NUM=3,
NGEOST_U_TIME=3600,
NGEOST_V_DEB=4,
NGEOST_V_NUM=3,
NGEOST_V_TIME=3600,
LT_ADV_FRC=T,
NT_ADV_DEB=7,
NT_ADV_NUM=6,
NT_ADV_TIME=1800,
LQV_ADV_FRC=T,
NQV_ADV_DEB=13,
NQV_ADV_NUM=6,
NQV_ADV_TIME=1800,
LLSW_FRC=T,
NLSW_DEB=19,
NLSW_NUM=1
/
```

The 3 first fields are a zonal geostrophic wind at 00, 01 and 02 hours. The 3 following fields are a meridionnal geostrophic wind at 00, 01 and 02 hours. The geostrophic forcing will be applied with a Coriolis parameter caracterictic of the midlatitudes.

The fields from GFL%YFORC(7) to GFL%YFORC(12) are temperature tendencies due to large scale advection. They are given every half an hour. The fields from GFL%YFORC(13) to GFL%YFORC(18) are temperature tendencies due to large scale advection. They are given every half an hour.

The last field is a large scale vertical velocity. The large scale vertical advection will be constant during the 2 hours of simulation.

In the version of SCUM in CY31T1, if the forcing are not constant, they have to cover the all period of simulation (one forcing time at the begining of the simulation and one at the end). We plan for a next version to have the possibility to prescibe the forcing only for a period of time during the simulation.

2.6.2 The forcing routines

The namelist NAMLSFORC is read in the setup routine SULSFORC and the forcing parameters are saved in the module YOMLSFORC.

The "tendency-like" forcing are added to the pronostic variables at time $t + \delta t$ in the grid point part of the calculation (subroutine CP_FORCING called in CPG_DYN) :

$$\phi(t + dt) = \phi(t + dt) + FORC(t) * \delta t$$

In 2TL or 3TL, the forcing are always interpolated at time t.

For the large scale forced vertical advection, an approximated computation of the vertical velocities necessary for the vertical advection in η-coordinate is done in the subroutine GPCTY_FORC called in CPG_GP using the values of the forced w (in m/s) interpolated at time t.

2.7 Setting of the "easy" diagnostics

A 1D model is a research or development tool which should be as flexible as possible in term of diagnostics. This constraint is not easy to fullfil in the context of an operationnal model like Arpege/IFS. To help for the analyses of the 1D case, a possibility to have any physical diagnostics in the output is offered. This possibility work like a "print" in the sense that the user has to modify the code where he wants the supplementary diagnostics and then recompile the routine.

As for the forcing, a GFL structure called GFL%EZDIAG is dedicated for case to case diagnostics. The fields of this GFL are not read in the initial file, but they are written in the output files. The number of fields in GFL%EZDIAG is **NGFL_EZDIAG**. It has to be set in **NAMGFL**. The name of each field has also to be set up in the namelist. For example :

```
YEZDIAG_NL(1)\%CNAME='MIXING_LENGTH',
YEZDIAG_NL(2)\%CNAME='SIGMAS',
YEZDIAG_NL(3)\%CNAME='EDMF_THL_FLUX',
NGFL_EZDIAG=3
```

With this declaration in namelist, the GFL structure GFL%EZDIAG is known until MF_PHYS and APL_AROME. If you need to use it at a other level of code, it has to be passed in argument.

With the NAMGFL namelist above, a few lines of code have to be added in APL_AROME for example to fill the first field of GFL%EZDIAG with the mixing length computed by the turbulent scheme. The second field is filled by the cloud variance σ_s and the third one by the vertical flux of θ_l computed from the EDMF scheme.

Please remember that in a GLF the fields are 3D fields with the same structure than the 3D pronostic variables of the model. But of course, it is not forbidden to use GFL%EZDIAG for diagnostic of 2D fields (one field for 1 level)...

2.8 Problems with surface fields and surface forcings

2.8.1 Surface fields

Your surface scheme is SURFEX

If you use SURFEX in a 3D or in a pseudo-1D configuration (in CY31T1), you need to create a special surfex surface file. For 3D, this file is prepared with a E927+PREPSURFEX configuration. This possibility was not yet tested in 1D (NSMAX=1). In 3D or 1D, you also need to start from a file (PGD file) with the physiographic informations compatible with SURFEX (ECOCLIMAP in particular) which has to be prepared with a special SURFEX tool. A set of SURFEX surface file for SCUM are available for the most classical 1D cases. If you are using SURFEX and you need a SURFEX initial file which is not available in the data base and you don't know how to create it with the SURFEX preparation tools, please, do not hesitate to contact me.

Note that, even if your case is defined with surface flux forcings, you need to create this SURFEX file.

Your surface scheme is ISBA_oper

The ACADFA1D tool used to create the atmospheric initial+forcing file is also able to add to this file the list of surface fields necessary for ISBA_oper. See chapter 3 for more details.

Your surface scheme is something else

Bad luck ... Please, contact me, we can study the problem together.

2.8.2 Surface forcings

The description of surface forcing is very much case dependent and the way these forcings may be done are often very much parametrisation dependant.

Your surface scheme is SURFEX

Your surface scheme is something else

2.9 Running SCUM

2.9.1 You're already running 3D cases on a supercomputer

You're already running 3D cases on a supercomputer with a physics you want to test in 1D. Then :

- Get the namelist from the web site corresponding to the 1D case you want to run and adapt it to your physics or create one as decribed in section **??**.
- Get a 1D FA file from the web site or prepare it with ACADFA1D.
- Modify your favorite 3D script to get the 1D namelist and the 1D FA initial file instead of the 3D ones.
- Submit your script on your supercomputer.
- If it is not working, do not become nervous immediatly, do not become angry against me (not yet), and send me a kind email (and maybe also flowers can help if you are in a hurry...) describing the problem. I'll try to help (or at least, I'll try to find the right person to help!).

2.9.2 You want to run SCUM on PC linux

You need (quickly evolving, please check with me if any doubt)

- pgf90 (6.1) (but soon G95 should be OK, we hope)
- install gmkpack on your PC
- compile tools delivered with CY31T0 (gribex etc)
- get sources for CY31T0-T1 (main CY31T0 + modifs corresponding to futur T1) from Ryad or me
- compile (with the scripts built by gmkpack) the main CY31T0 in one main pack and and the T1 modifs in a local pack.

If everything goes well, you get an executable binary in the bin directory of your local pack.

To run this executable binary, it will be easier to write a simple script very similar to what is used on the supercomputer. A simple example of script will be available on the web site (but as the web site is not ready, please ask me for the time being...)

Once you have your executable binary, the ingredients necessary to run SCUM are

- an initial FA file (result of ACADFA1D)
- an ARPEGE/IFS namelist
- a script to make the things easier...

If you are using SURFEX, you also need :

- a SURFEX file (.lfi + .des)
- a namelist for surfex

Once you have ran the model, you will get an historic FA file at each output time you asked (default name is usually ICMSHxxx). If you use SURFEX, you will also have surfex output files if you ask some (default name is AROMOUT_xxx).

Chapter 3

ACADFA1D : initial file and forcing preparation for a 1D case

3.1 Introduction

In the context of 1D model, the acadfa tool (acadfa1D) may more or less be seen as a **ascii2FA** tool (acadfa1D does not do vertical interpolation for exemple). It is a very basic code in F90 which may very easily be adapted to a new configuration if needed.

It is working with a namelist/ascii file which contains all the informations necessary to create the FA file.

Then, the main work when you create a new case is to create the namelist/ascii file as explain below.

3.2 Installation of the acadfa1D software

From the tar file acadfa1D.tar, you should recover 3 directories : util, src and run. The sources are under src. An exemple of Makefile (for PC linux and pgf90 compiler) is delivered with the fortran source code necessary for the compilation of the tool acadfa1D (under acadfa1D/src). A linking with the xrd library (FA software, spectral transforms) is necessary. A light version of the xrd library of Arpège is proposed with the acadfa1D tool under src/xrd. Makefiles are supplied for the compilation of the xrd sources. If some changed are introduced in the xrd routines in a new cycle, it may happen that acadfa1D has to be phased.

Run acadfa1D

You need to have in the same directory the executable file (acadfa1D.exe) created during the installation of the software and a namelist/ascii file called nam1D which contains all the necessary information for your case (see below for information about the creation of a namelist/ascii nam1D). You will find an example of a namelist/ascii file under

acadfa1D/run.

After execution, acadfa1D creates a pseudo-1D FA file called 1D.file. You can check the contents of this file with the tool **frodo** which returns the main informations about your file (the source of frodo may be found under acadfa1D/util).

The namelist/ascii file nam1D

The namelist/ascii file nam1D contains some parameters in a namelists, the profils needed to described the initial state of your 1D atmosphere in an ascii format and the forcings at all the forcing times in an ascii format.

The namelists **NAM1D** contains :

IFLEV : number of vertical levels

ZDELY : size of the horizontal grid (exact value usually non important in 1D)

LNHDYN : switch of hydro or non-hydro

LALAPHYS : switch for Arpège/Aladin basic surface fields

LREASUR : switch for complementary surface fields in the case of Arpège/Aladin version of ISBA

LQVSP : switch for q_v in spectral space

LQVGRP : switch for q_v in grid point space

LQCGRP : switch for q_c in grid point space

LQIGRP : switch for q_i in grid point space

LQRGRP : switch for q_r in grid point space

LQSGRP : switch for q_s in grid point space

LQGGRP : switch for q_g in grid point space

NFORC : total number of forcings

Exemple of **NAM1D** (with default values) :

```
&NAM1D
  IFLEV     =100,
  ZDELY     =250000.,
  LNHDYN    =.FALSE.,
  LALAPHYS  =.FALSE.,
  LREASUR   =.FALSE.,
  LQVSP     = .FALSE.,
```

```
  LQVGRP    = .TRUE.,
  LQCGRP    = .FALSE.,
  LQIGRP    = .FALSE.,
  LQRGRP    = .FALSE.,
  LQSGRP    = .FALSE.,
  LQGGRP    = .FALSE.,
  LCFGRP    = .FALSE.,
  LSRCGRP   = .FALSE.,
  NFORC     = 0
/
```

The different blocks of the ascii format start with a key word.

The block starting with the keyword **ETA** contains the description of the vertical discretisation (function A and B of the hybrid coordinate). A and B are known at the full level of the model (NFLEV+1 values). A has to be 0 at the surface, and $\neq 0$ at the top of the model, and B has to be 1 at the surface and 0 at the top of the model (the top has to be a pure pressure level). In 1D, you can easily transform the hybrid coordinate in pressure coordinate using B=1 at the surface and 0 above and A=0 at the surface and $A = p/p_{OO}$ above where p is the pressure of the level.

The block starting with the keyword **ATMOSPHERE** contains the initial profils of the pronostic variables :

- altitude of the surface (surface orography)
- surface pressure p_s
- u at half level (NFLEV values)
- v at half level (NFLEV values)
- T at half level (NFLEV values)
- vapor and other hydrometeor fields (if used in the microphysics) at half level (NFLEV values)

The block starting with the keyword **FORCING** contains all the atmospheric forcings at half levels. If the forcing are time dependant, the different instants have to be one after the other.

If you want to use the Arpege/Aladin surface scheme, the initial file has to contain some information about the surface (scalar information). They are given in an separate **SURFACE** block. For technical reason some surface fields have to be present in the initial file even if you are using SURFEX. This should change in the futur (at least I hope so...).

Preparation of a case in practice

If you already have a case working in hybrid or pressure coordinate with u, v, T and q_v as pronostic variables, the preparation of the initial nam1D file should be easy (except maybe the surface forcing description).

We may develop if needed a few basic tools to transform profils initially in z-coordinate to an hybrid or pressure coordinate, θ into T profils, r_v into q_v profils etc.

Chapter 4

FA2ASCII and simple diagnostics : flux software

4.1 Introduction

A simple fortran code was developped by the 2D developpers to make easier the utilisation of the output 2D FA files (software known in the academic case community as **flux**. maybe we should find an other name ?). This code may be seen as a FA2ascii tool but it allows also some simple diagnostics as computation of potential temperature or transformation from specific humidity to mixing ratio. If it is proved to be usefull, this code may be improved or completed.

An other option would be to adapt FULLPOS for ideal cases.

The flux tool was adapted for linux PC. It is not very user-friendly (yet), and then it is strongly advised to have a look at the source before to start to run it.

The version of flux delivered in June 2006 to the first SCUM users transforms an historic output FA file of SCUM into a series of ascii files, each of them containing a vertical profile for a given variable. By default, you will get profile of u,v,w hydro, qv (called q), qc, qi, qr, qs, qg, T, theta, p, tke, cloud fraction. but it is really very easy to modify flux to get a profile or a scalar value of any field present in your output FA files. The only argument of flux is the name (full path if needed) of you FA file. A little script (run_moreflux) is proposed to run flux on a full series of output files.

Chapter 5

Implementation for KNMI Parametrization Testbed

Input and control of the model is controlled via namelists. Namelists are created by a simple NCL script. There are namelists for the preparation of the initial SURFEX file and forecast model. Modeloutput is available in FA and ascii files. A special utility fa2ascii gathers all available output data and put everything in a NETCDF file To avoid that the 1D-model drifts away from the driving 3D model a relaxation takes place which keeps the model in pace.

```
#! /bin/bash
#
# job_MUSC_Testbed_daily.sh <dtg> <location> <hostmodel>
#
# Testbed control script for Harmonie
#
# Actions (chronological):
#
# Read netcdf driver file
# Make namelist including forcings, profiles for relaxation etc.
# Use namelist to make fa (fichier Arpege) and surfex (lfi) inputfile
# Run SCM of HARMONIE also called (MUSC, SCUM)
# Make netcdf testbed output file
#
# 26-08-2009 Wim de Rooy & Cisco de Bruijn
#
#

set -x
cd /net/bhw251/nobackup/users/bruijnde/
```

```
home=‘pwd‘


dtg=$1
location=Cabauw
hostmodel=RACMO.HARMONIE-L60
nstep=2160
tstep=120
nave=1

neggers=’/net/bhw276/nobackup/users/neggers’
input=${neggers}/Testbed/archive/drivers/${location}/${hostmodel}
output=${neggers}/Testbed/archive/model/${location}/${hostmodel}

#
# Make namelist
#
cd  $home/AROME/makeNamelists
./makeNamelist.sh ${dtg} ${hostmodel} ${location}
status=$?
if [ ! $status -eq 0 ];then
 echo "makeNamelist went wrong"
 exit
fi

#
# Make FA input file
#
cd $home/AROME/ascii2fa/scr
./HARMONIE_ascii2fa.sh  > log_ascii2fa
status=$?
if [ ! $status -eq 0 ];then
 echo "ascii2fa went wrong"
 exit
fi

#
# Make surface LFI input file
#
cd $home/AROME/surfex
./prep_surfex.sh
status=$?
if [ ! $status -eq 0 ];then
```

```
 echo "prep_surfex.sh went wrong"
 exit
fi

#
# Run MUSC in two flavours
#
for version in EDMF EDKF
do
cd $home/AROME/RUN_PACK/Testbed
./job_MUSC_33t1_Testbed.sh  $version  $nstep $tstep
status=$?
if [ ! $status -eq 0 ];then
 echo "job_MUSC_33t1_Testbed.sh went wrong"
 exit
fi
#
# Make netcdf output file
# Arguments TSTEP and NSTOP should be adapted manually
# in the namelist of fa2kpt.sh
#
cd $home/AROME/fa2ascii/scr
./fa2kpt.sh $dtg $version $nstep $tstep $nave > log_fa2kpt
status=$?
if [ ! $status -eq 0 ];then
 echo "fa2kpt.sh of " $version "went wrong"
fi
done
```

Chapter 6

Installation of AROME

```
HOW TO INSTALL AROME
        ====================

The files:

-rwxr-xr-x 1 moene w-ond 12676657 2007-05-21 08:05 33t1_main.01.tar.gz*

Source code of Cycle 33T(oulouse)1

-rwxr-xr-x 1 moene w-ond   872842 2007-05-21 08:05 auxlibs_ecmwf.0.2.tgz*

Source code of auxiliary libraries from ECMWF

-rwxr-xr-x 1 moene w-ond 13873766 2007-05-21 08:06 auxlibs_meteo-france.0.2.tgz*

Source code of auxiliary libraries from Meteo France

-rwxr-xr-x 1 moene w-ond   144090 2007-05-21 08:06 gmkpack.6.2.3.tgz*

GMKPACK

-rwxr-xr-x 1 moene w-ond  3792350 2007-05-21 08:06 local_33_t1.tgz*
-rw-r--r-- 1 moene w-ond       53 2007-05-21 08:07 README
-rwxr-xr-x 1 moene w-ond  1672894 2007-05-21 08:06 tora_local.tar.gz*

Added to ~/.bash_profile:

export GMKROOT=/nobackup/users/moene/AROME/gmkpack.6.2.3
export ROOTPACK=/nobackup/users/moene/AROME/pack
export HOMEPACK=$ROOTPACK
```

```
export HOMEBIN=$HOMEPACK
export GMKFILE=PGI.LINUX
export GMKTMP=$TMPDIR
export PATH=$GMKROOT/util:$PATH
export MANPATH=$MANPATH:$GMKROOT/man
export ECMWFLIB=/nobackup/users/moene/AROME/auxlibs_ecmwf.0.2_thelibs
export METEOFRANCELIBS=/nobackup/users/moene/AROME/auxlibs_meteo-france.0.2_thelibs
```

(The last two for the PGI.LINUX configuration file).

Created auxlibs_ecmwf.0.2_thelibs to store the compiled auxiliary libraries from ECMWF.

Untarred auxlibs_ecmwf.0.2.tgz and updated the make_everything script:

```
UNAME=Linux
R64=R64
USE_GNU=no
INSTALL_DIR=/nobackup/users/moene/AROME/auxlibs_ecmwf.0.2_thelibs
READONLY_LIB=no
```

and then run it, it creates:

```
drwxr-xr-x 4 moene w-ond 4096 2007-05-21 08:22 bufrdc_000240/
-rwxr-xr-x 1 moene w-ond  778 2006-08-29 13:02 clean_everything*
drwxr-xr-x 4 moene w-ond 4096 2007-05-21 08:22 ec_001/
drwxr-x--- 9 moene w-ond 4096 2007-05-21 08:23 gribex_000281/
-rwxr-xr-x 1 moene w-ond 1364 2007-05-21 08:22 make_everything*
-rw-r--r-- 1 moene w-ond 1755 2006-08-29 13:02 make_everything.vpp
-rw-r--r-- 1 moene w-ond 1909 2006-08-29 13:04 README
```

Created the directories auxlibs_meteo-france.0.2_thelibs for the auxiliary libraries and auxlibs_meteo-france.0.2_include for the auxiliary include files from Meteo France.

Untarred auxlibs_meteo_france.0.2.tgz and updated the make_everything script:

```
UNAME=Linux
R64=R64
USE_GNU=no
INSTALL_DIR=/nobackup/users/moene/AROME/auxlibs_meteo-france.0.2_thelibs
INSTALL_HEADERS=/nobackup/users/moene/AROME/auxlibs_meteo-france.0.2_include
READONLY_LIB=no
```

```
and then run it, which creates:

-rw-rw-rw- 1 moene w-ond   2384 2007-05-21 08:32 libC_codedummy_000R64.a
-rw-rw-rw- 1 moene w-ond   3364 2007-05-21 08:32 libfdbdummy_000R64.a
-rw-rw-rw- 1 moene w-ond   3854 2007-05-21 08:32 libibmdummy_000R64.a
-rw-rw-rw- 1 moene w-ond  22900 2007-05-21 08:32 libmpidummy_000R64.a
-rw-rw-rw- 1 moene w-ond   1336 2007-05-21 08:32 libnaglitedummy_000R64.a
-rw-rw-rw- 1 moene w-ond   2246 2007-05-21 08:32 liboasisdummy_000R64.a
-rw-rw-rw- 1 moene w-ond  11986 2007-05-21 08:32 libodbdummy_000R64.a
-rw-rw-rw- 1 moene w-ond 580632 2007-05-21 08:32 librgb_000R64.a
-rw-rw-rw- 1 moene w-ond   1770 2007-05-21 08:32 libwamdummy_000R64.a

Untarred gmkpack.6.2.3.tgz; then edited the configuration file
PGI.LINUX in directory gmkpack.6.2.3/arch:

ECMWFLIB=/nobackup/users/moene/AROME/auxlibs_ecmwf.0.2_thelibs
METEOFRANCELIBS=/nobackup/users/moene/AROME/auxlibs_meteo-france.0.2_thelibs

# "Read Grib from BDAP":
LD_USR01 = $METEOFRANCELIBS/librgb_000R64.a
# "Bufr decoding":
LD_USR02 = $ECMWFLIB/libbufrdcR64.a
# "ec":
LD_USR03 = $ECMWFLIB/libecR64.a
# "Gribex (or emos)":
LD_USR04 = $ECMWFLIB/libgribexR64.a
# "ecmwf field database":
LD_USR05 = $METEOFRANCELIBS/libfdbdummy_000R64.a
# "ecmwf wave model":
LD_USR06 = $METEOFRANCELIBS/libwamdummy_000R64.a
# "C code generated by blacklist":
LD_USR07 = $METEOFRANCELIBS/libC_codedummy_000R64.a
# "Nag":
LD_USR08 = $METEOFRANCELIBS/libnaglitedummy_000R64.a
# "OASIS":
LD_USR09 = $METEOFRANCELIBS/liboasisdummy_000R64.a

LD_SYS01 = $METEOFRANCELIBS/libibmdummy_000R64.a

LD_MPI01 = $METEOFRANCELIBS/libmpidummy_000R64.a

DUMMY_INCPATH = /nobackup/users/moene/AROME/auxlibs_meteo-france.0.2_include/mpidummy_00
```

and create the following link:

ln -s PGI.LINUX PGI.LINUX.x

Then ran build_gmkpack in directory gmkpack.6.2.3.

Create the directory for the new pack:

mkdir /nobackup/users/moene/AROME/pack

Then run gmkpack:

gmkpack -r 33t1 -a -p arome

and go to directory 33t1_main.01.G95V091.x/src/local.

Untar the sources

tar zxvf ../../../../33t1_main.01.tar.gz

Then run the compile script from the 33t1_main.01.G95V091.x directory:

./ics_arome

It will abort because of a bug in flex 2.5.31.

Change sys/odb98/lex.yy.c so that

```
#define INITIAL 0
#define LEX_NORMAL 1
#define LEX_INCLUDE 2
#define LEX_SET 3
#define LEX_TYPE 4
#define LEX_TABLE 5
#define LEX_VIEW 6
#define LEX_FROM 7
#define LEX_ORDERBY 8
#define LEX_EXCLUDED_BY_IFDEF 9
```

precedes their uses.

Re-compile the lex.yy.c file:

```
cc -c -I. -I../../src/local/odb/compiler lex.yy.c
```

and re-run the compile script 'ics_arome'. This is a bug, confirmed by the flex maintainers (http://flex.sourceforge.net).

The following routines have to be compiled without debugging (no -g option), probably as a consequence of a bug in the PGF90 5.2.4 compiler not present in the 6.1.6 version:

```
modn_prep_isba.o
modn_prep_seaflux.o
modn_prep_teb.o
modn_prep_watflux.o
modn_assim.o
modn_chs_orilam.o
modn_dst.o
```

in directory src/local/mse/module

Now create a local pack (in directory pack):

```
gmkpack -r 33t1 -b main -u local -p arome
```

Go to directory 33t1_local.01.G95V091.x/src/local/mse/externals and edit aro_ground_param.mnh (change all RESHAPEs of arguments to vector copies). This is probably a bug in the PGF90 5.2.4 compiler not present in the 6.1.6 version.

then execute ics_arome in 33t1_local.01.G95V091.x