USER GUIDE FOR HARMONIE MUSC

This draft is provided for the MUSC working week 29.11-2.12.2011 as a template for further development

Ulf Andrae, Eric Bazile, Cisco de Bruijn, Laura Rontu, Sami Saarinen ...

E-mail: *Laura.Rontu@fmi.fi*

Abstract:

aaaa

1. INTRODUCTION

Single-column models are used for development and testing of physical parametrizations of NWP models. In principle, such models contain in principle the whole source code of the prognostic model. This should allow to run idealised experiments focusing on the atmospheric physics in a simplified framework, but based on the same code that is used for weather prediction and research. Simplified framework means

- Time-integration is done in a single atmospheric column
- The initial state, possible atmospheric forcing during time-stepping and surface description are provided as input
- Horizontal advection is not included and the model dynamical formulations are applied in minimal extent

Because of the simplifying assumptions, a single-column model is not suitable for real forecasting. Its value is in the possibility of study sensitivity of the model to different formulations of the physical parametrizations in different realistic atmospheric conditions. It is possible to quickly run experiments with any combination of physical parametrizations in a workstation environment. Application of very high resolution in vertical should be possible. On the other hand, any results and suggestions based on single-column studies, need careful testing in the full model. A special approach and additional tools are required for handling input and output of a single-column model.

The first version of the Single Column Unified Model (SCUM) was derived from the full IFS code (cy32?) by Sylvie Malardel in 200?. . Later SCUM was renamed, for aestethical reasons, to MUSC - Model Unifie Simple Colonne. MUSC has been further developed and applied in Meteo France by Eric Bazile, in KNMI by Cisco de Bruijn et al., by Javier Calvo et al. in AEMET. Versions based on IFS cycle 33 and 35 have been defined. A user guide **?** was written and updated at KNMI. During this development, MUSC started to diverge from the three-dimensional reference code.

The first of HARMONIE MUSC is being built on IFS cycle 37, taking a MUSC export version of cycle 37t1 from Meteo France as a starting point. The aim is to bring MUSC back to the original idea of being fully compatible with the three-dimensional HARMONIE. Presently, a separate branch harmonie_MUSC exists in hirlam.org subversion repository. In the future, this branch should converge with the trunk of HARMONIE. In the first step of conversion, it should be possible to run any three-dimensional HARMONIE experiment based on the code of the MUSC branch.

The specific features of the HARMONIE MUSC code include

- Source code is maintained and made available in the hirlam.org repository
- Compilation of the code is based on the HARMONIE makeup
- A HARMONIE file-handling utility gl may be applied for extraction of input data from the three-dimensional HARMONIE files

2. Setup of harmonie_MUSC

2..1 Structure of the code

harmonie_MUSC branch contains HARMONIE (cycle37h) source code + dedicated utilities and scripts. The idea is that all basic HARMONIE source code should be available together with some MUSC-specific additional tools for handling of input, output and MUSC experiments.

It is suggested that in the workstation environment, a structure with reference directory, possibly under a reference user's home directory open for reading to all users, and the user home and scratch directories are kept separetely. This is the structure used in HIRLAM and HARMONIE 3D experiments with or without mini-SMS user interface.

For example:

/home/hirlam/harmonie_MUSC/			
	src/		
	scr/		
	reflib/		
	util/		
	bin/		
/home/rontu/musc			
	scr/		general scripts
	EXP/EXPm - EXPn/		$MU_WD:
		src/	modified source codes
		OUTPUT/	output directories
	scratch/EXPm...EXPn/		$MU_WRK:
		lib/src	compiled and source code
		bin/	compiled MASTER

2..2 Getting the source code, utilities and test data

cd /home/hirlam
svn -co https://svn.hirlam.org/branches/harmonie_MUSC

You need a password for hirlam.org for this. With subversion checkout you will get the a README file source code, utilities, scripts and a testdata package.

2..3 Compilation with makeup

First please check you have needed software in your computer. You will need gcc and gfortran > v. 4.4, mpi libraries (e.g. in Debian libmpich2-dev), lapack-blas library (e.g. in Debian liblapack-dev), ksh, flex, bison. Please ask your systems package installation tool to bring these to you, e.g. in Debian or Ubuntu 'apt-get install bison' etc. There are MUSC configurations for debian, ubuntu, redhat gfortran, work with fedora has been started. The configuration files are in util/makeup .

Next, build the code:

cd /home/hirlam/harmonie_MUSC
util/makeup/build config.debian.gfortran

This will produce the main executable file /home/hirlam/harmonie_MUSC/bin/MASTER, which can be later linked to your experiment. In addition, several utilities will be compiled and added to the same directory.

You will need DDH toolbox. Compile it separately:

cd /home/hirlam/harmonie_MUSC/util/ddhtoolbox/tools
sh ./install.sh

Please find documentation of toolbox usage in
/home/hirlam/harmonie_MUSC/util/ddhtoolbox/documentation.

3. Running a MUSC experiment

3..1 Structure of an experiment

Figure 1: Structure of a MUSC experiment from input to visualisation.

3..2 Script Runmusc

```
# exp=ARPE
TSTEP=300
NSTOP=h12
mod=ARPEGE
klev=70
ver=25
```

```
cycle=37t1
#
expmusc=$basemusc/EXP/$EXP
NAM_PATH=$expmusc
FILE_PATH=$expmusc
namfil=$NAM_PATH/namus_$cycle_$mod
inifil=$FILE_PATH/initfile_$ver_L$klev
OUTNAME=$mod_$ver_$TSTEPs_L$klev_$exp
#
OUT_PATH=$expmusc/OUTPUT
WRK_PATH=$wrkmusc/rundir
test -d $OUT_PATH || mkdir -p $OUT_PATH
test -d $WRK_PATH || mkdir -p $WRK_PATH
cd $WRK_PATH
if [ $cycle == "37t1" ] ; then
test -e $namfil || echo ; echo ========== ERROR: namelist missing, exiting ============== ; exit ; && cp $namfil namprovi
# Fichier et namelist SURFEX
# cp $FILE_PATH/AROME_PREPSURF.lfi TEST.lfi
# cp $FILE_PATH/EXSEG1.nam EXSEG1.nam
fi
#
# run time optimization options
# export ZOPT="-Wl,-d100,-g250,-e1"
date
#
#ADVEC=eul
ADVEC=sli
#
cat < namprovi
#
# Fichier initial
test -e $inifil && /bin/cp $inifil ICMSHL075INIT || echo ; echo ========== ERROR: inifile missing, exiting ============== ; exit ;
FILENAME1='ICMSHL075INIT'
FILENAME2='ICMSHL075INIT'
#
# init file and coupling file
#
/bin/cp $FILENAME1 ICMSH$expINIT
#
#
# *********************************
# * Execution *
# *********************************
#
set +x
echo ''
echo ' Acquisition de l executable '
echo ' Get the executable file '
echo ''
set -x
#
# Get the executable file
#
if [ "$MYOWNEXE" != '' ] ; then
ln -s $MYOWNEXE MASTER
# chmod 755 MASTER
```

```
else
echo "GIVE MYOWNEXE"
exit
# get $GOUROUBIN
# mv $GOUROUBIN MASTER
fi
#
set +x
echo ''
echo ' Copie de la namelist sur fort.4 '
echo ' Namelist copy on fort.4 '
echo ''
set -x
#
cat namprovi | awk 'print $1' FS=";" > fort.4
#
set +x
echo ''
echo ' Execution du job ALADIN '
echo ' ALADIN job running '
echo ''
set -x
#
#
#export DR_HOOK=1
#
./MASTER -c001 -vmeteo -maladin -e$exp -t$TSTEP -f$NSTOP -a$ADVEC > lola 2>&1
#
rm fort.4
ls -l
#
#
set +x
echo ''
echo ' Listing d execution pour partie non parallelisee: fichier lola'
echo ' Listing for the not parallelised part: file lola'
echo ''
set -x
cat lola
#
if [ -a NODE.001_01 ]
then
for file in NODE*
do
set +x
echo ''
echo ' Listing d execution pour partie parallelisee: fichier' $file
echo ' Listing for the parallelised part: file' $file
echo ''
set -x
cat $file
done
fi
#
#
# ********************************
# * Envoi des resultats du modele *
# * Save model results *
```

```
# *********************************
#
set +x
echo ''
echo ' Sauvegarde des fichiers historiques '
echo ' Historic files saving '
echo ''
set -x
#
# define name of output files on delage
#
mkdir $FILE_PATH/OUTPUT/$OUTNAME
$FILE_PATH/OUTPUT/$OUTNAME/*
#
# save model results
#
mv Out* $FILE_PATH/OUTPUT/$OUTNAME
mv OUTPUT_* $FILE_PATH/OUTPUT/$OUTNAME
mv NODE* $FILE_PATH/OUTPUT/$OUTNAME
#
#
set +x
echo ''
echo ' Fichiers presents sur le workdir $WRK_PATH '
echo ' Present files on the workdir $WRK_PATH '
echo ''
set -x
ls -l
# *********************************
# * Nettoyage final *
# * Final Cleaning *
# *********************************
#
set +x
echo ''
echo ' Nettoyage final du workdir $WRK_PATH '
echo ' Final cleaning of the workdir $WRK_PATH '
echo ''
set -x
rm namprovi
rm MASTER
# suppression des fichiers ALADIN / removal of the ALADIN files.
for file in ICMSH* DHF*
do
rm $file
done
for file in AROMOUT*
do
rm $file
done
for file in ELSCF*
do
rm $file
done
#
rm ECHIS
rm ifs.disp
rm ifs.stat
```

```
rm lola
if [ -a NODE.001_01 ]
then
for file in NODE*
do
rm $file
done
fi
rm -f core*
cd ..
rmdir $WRK_PATH
cd $OUT_PATH
ln -s $basemusc/lfa2ascii.sh .
sh ./lfa2ascii.sh $OUTNAME
```

3..3 Test input and expected output

The test data for your experiment are in the package musctest.tgz. Please open it in your own home directory:

```
cd
cp /home/hirlam/harmonie_MUSC/musctest.tgz .
tar xvfz musctest.tgz
```

and copy also the reference scripts:

```
cd musc
cp -r /home/hirlam/harmonie_MUSC/scr .
cp scr/mur $HOME/bin/
```

Please check the scripts mur and Runmusc for the directory definitions and usage alternatives!

Now you are ready to say:

```
scr/mur MUTEST
```

The script 'mur' will set your experiment environment and bring you to the new experiment directory $HOME/musc/EXP/MUTEST. It is also called $MU_WD for your convenience. With the test package, you already got test data to your new $MU_WD. 'mur' creates a defitinion file tool.path and advises you to run it as '. ./tool.path' to make utilities and scripts available in your $PATH. Do not forget to do as advised!

In $MU_WD you should say

```
./Runmusc > musc.log
```

to run Eric's test case and use the ddh tools with Eric's script lfa2ascii.sh to convert main output files to ASCII. You can have a look at them or try something. However, we are working with the next step, i.e. visualisation tools for the lot of ASCII files of the instant and accumulated profiles or different variables.

4. Preparation of input for experiments

You might want to produce your own atmospheric column for MUSC input, instead of the initfile provided by the test package. For this,

```
gl -l -n naminterp_fa ICMSHHARM+0006_aladin
```

(you should have got naminterp_fa and ICMSHHARM+0006_aladin with the test package, newest version!). gl should produce you a file ICMSHHARM+0006_aladin.fa, which is ready for input to your experiment. Not that gl should be the one recently compiled in $refmusc/bin! Please copy the reference Runmusc to Runmusc_testinput and modify it by changing klev value as needed (command 'gl -l ICMSHHARM+0006_aladin | grep lev' gives the needed number). 'chmod 755 Runmusc_testinput' and ./Runmusc_testinput. And yes, copy the ICMSHHARM+0006_aladin.fa to a file with the name Runmusc_testinput expects. Check the directory OUTPUT for new files. Works? Got the idea? By the way, it might be in the future better to create a directory $MU_WRK/muscin and go there for input preparation, to avoid extra mess with the files.

4..1 Available tools: gl and acadfa + pgd-prep

4..2 Preparation of atmospheric column

4..3 Preparation of surface dat

5. Modifying of namelists for MUSC experiment

5..1 Run-time namelist

5..2 SURFEX-special definitions

5..3 Example for AROME / ALARO / ARPEGE

6. Modifying source code of an experiment

6..1

You can modify harmonie_MUSC source codes in your $MU_WD. For example, to modify apl_arome.F90 in an experiment called MY01 you do this:

```
mkdir -p src/arp/phys_dmn
cp $refmusc/src/arp/phys_dmn/apl_arome.F90 src/arp/phys_dmn
cd src/arp/phys_dmn
[modify apl_arome.F90 as wanted, save]
cd $MU_WD
mur MY01 recompile
```

Your modifications will be copied over the reference code in $MU_WRK/lib/src/arp/phys_dmn and compiled. After successful compilation, you can again say ./Runmusc and see the results. Please note that if you indeed modified apl_arome.F90, you will see no difference with the testset MUTEST because it does not use AROME!

7. Handling of MUSC output

7..1 Output variables

7..2 Conversion of LFA files

7..3 Visualisation possibilities

8. Development tasks

REFERENCES

Boer, G. J., N. A. McFarlane, R. Laprise, J. D. Henderson, and J.-P. Blanchet, 1984:The Canadian Climate Centre spectral atmospheric general circulation model. *Atmos. Ocean.*, **22**, 397–429.

Appendix

- gl for extraction from HARMONIE files
- acadfa + pgd + prep for academic cases
- ddh toolbox
- xmgrace, gnuplot âĂę for simple plotting
- grads, matplotlib, metview - for data analysis and plotting

TYPE	LEVELS	NAME	DESCRIPTION
C	1	INDICE	EXPERIENCE
R4	nlev	RHO	DENSITY
R4	nlev+1	RHO_fluxI	DENSITY FLUX
I4	1	KLEV	NO VERTICAL LEVELS
R4	1	TSPHY	PHYSICS TIMESTEP
I4	1	NINDAT	DTG AS INTEGER
I4	1	NSSSSSa	initial time in seconds (e.g. for 12h, 43200)
I4	1	KCLPH	LEVEL OF PBL
R4	1	RSTATI	NUMBER OF SECONDS SINCE START OF THE MODEL
R4	nlev	PU	U-COMPONENT
R4	nlev	PV	V-COMPONENT
R4	nlev	PVENT	WINDSPEED
R4	nlev	PDIRVENT	WINDDIRECTION
R4	nlev	PT	TEMPERATURE
R4	nlev	PQ	SPECIFIC HUMIDITY
R4	nlev	ZH	HEIGHT OF HALF-LEVELS
R4	nlev	THETA	POTENTIAL TEMPERATURE
R4	nlev	THETAV	VIRTUAL POTENTIAL TEMPERATURE
R4	nlev	THETAL_KE	
R4	nlev	THETAL_BS	
R4	nlev	THETAL	LIQUID WATER POTENTIAL TEMPERATURE
R4	nlev	THETAVL	LIQUID WATER POTENTIAL VIRTUAL TEMPERATURE
R4	nlev	PECT	
R4	nlev	PQI	SPECIFIC HUMIDITY ICE
R4	nlev	PQL	SPECIFIC HUMIDITY LIQUID WATER
R4	nlev	PQSN	
R4	nlev	PQR	
R4	nlev	PQG	
I4	1	KGL1	
I4	1	KGL2	
I4	1	KSGST	
I4	1	KSTEP	TIMESTEP
I4	1	KCSS	
R4	1	PMU0	SOLAR ZENITH ANGLE
R4	1	PGEMU	
R4	1	PGELAM	
R4	nlev+1	PAPHI	geopotential height "gz" at half levels
R4	nlev+1	PAPRS	hydrostatic pressure at half levels
R4	nlev	PAPHIF	geopotential height "gz" at half levels
R4	nlev	PAPRSF	hydrostatic pressure at full levels
R4	nlev	PALPH	COEFFICIENTS OF THE HYDROSTATICS
R4	nlev	PDELP	PRESSURE DIFFERENCE ACROSS LAYERS
R4	nlev	PLNPR	
R4	nlev	PRDELP	
R4	nlev	PCP	
R4	nlev	PR	

TYPE	LEVELS	NAME	DESCRIPTION
R4	nlev	PVERVEL	VERTICAL VELOCITY
R4	nlev	ZWA	
R4	nlev	ZOMEGA	LARGE SCALE VERTICAL VELOCITY
R4	nlev+1	PEMTD	TOTAL DOWNWARD LONGWAVE EMISSIVITY
R4	nlev+1	PEMTU	TOTAL UPWARD LONGWAVE EMISSIVITY
R4	nlev+1	PTRSO	TOTAL SHORTWAVE TRANSMISSIVITY
R4	1	PCLON	COSINE OF LONGITUDE
R4	1	PSLON	SINE OF LATITUDE
R4	nlev+1	PFRSO	NET SW RADIATIVE FLUX
R4	nlev+1	PFRTH	NET LW RADIATIVE FLUX
R4	1	PTCLS	2M TEMPERATURE
R4	1	PUCLS	10M U-COMPONENT
R4	1	PVCLS	10m V-COMPONENT
R4	1	PQCLS	2M SPECIFIC HUMIDITY
R4	1	PCLCT	TOTAL CLOUD COVER
R4	1	PCLCH	HIGH CLOUD COVER
R4	1	PCLCM	MEDIUM CLOUD COVER
R4	1	PCLCL	LOW CLOUD COVER
R4	1	PCLCC	CONVECTIVE CLOUD COVER
R4	1	VENTCLS	10M WIND SPEED
R4	1	DIRCLS	10M WIND DIRECTION
R4	1	PFCLL	LATENT HEAT FLUX OVER LIQUID WATER (OR WET SOIL)
R4	1	PFCLN	LATENT HEAT FLUX OVER SNOW (OR ICE)
R4	1	PFCS	SENSIBLE HEAT FLUX AT SURFACE LEVEL
R4	nlev	PNEB	FRACTIONAL CLOUDINESS FOR RADIATION

TYPE	LEVELS	NAME	DESCRIPTION
R4	1	ECT_INT	
R4	1	LWP	LIQUID WATER PATH
R4	1	IWP	ICE WATER PATH
R4	1	CWP	TOTAL LIQ+ICE PATH
R4	1	WVP	WATER VAPOR PATH
R4	nlev+1	PFPLSL	LIQUID PRECIPITATION
R4	nlev+1	PFPLSN	SNOW PRECIPITATION
R4	nlev+1	PFPLSG	GRAUPEL PRECIPITATION
R4	nlev+1	PFPLSHL	HAIL PRECIPITATION
R4	nlev+1	PREC_TOT	TOTAL PRECIPITATION
R4	nlev	PQICE	SPECIFIC ICE CONTENT
R4	nlev	PQLI	SPECIFIC LIQUID WATER CONTENT
R4	nlev	PRH	RELATIVE HUMIDITY
R4	1	PTS	SURFACE TEMPERATURE
R4	1	PQS	SURFACE SPECIFIC HUMIDITY
R4	1	PFRSODS	SURFACE DOWNWELLING SW
R4	1	PFRTHDS	SURFACE DOWNWELLING LW
R4	1	PITM	
R4	nlev+1	PSTRTU	
R4	nlev+1	PSTRTV	
R4	2	PFRSOC	SW CLEAR SKY NET RADIATION TOA/SFC
R4	2	PFRTHC	LW CLEAR SKY NET RADIATION TOA/SFC
R4	1	THETAS	SURFACE POTENTIAL TEMPERATURE
R4	1	HCLA	hauteur de la couche limite atmospherique
R4	1	HCLA_AY1996	
R4	nlev+1	ZCP_flux	
R4	nlev+1	ZLH_flux	LATENT HEAT
R4	1	ZUSTAR	FRICTION VELOCITY
R4	1	SAT_DEF_0-15	
R4	1	SAT_DEF_1-15	
R4	1	SAT_DEF_2.5-4.5	
R4	1	SAT_DEF_1.4-3	
R4	1	SAT_DEF_QV_0-15	
R4	1	SAT_DEF_QV_1-15	
R4	1	SAT_DEF_QV_2.5-4.5	
R4	1	SAT_DEF_QV_1.4-3	
R4	1	ZCLDTOP	
R4	1	ZCLDBAS	
R4	1	ZMAXFRAC	