How to run MUSC and prepare new cases easily

David Němec (CHMI, Prague)

Czech
Hydrometeorological
Institute

Outline

- running MUSC
- namelist options
- github repository with cases
- creation of new cases

What is MUSC?

- Modèle Unifié, Simple Colonne
- 1D model
- 4 points
- forcing: prescribed values for selected variables
- less feedback
- good for testing physical parameterizations and debugging

Tree puzzle pieces to run MUSC

1. model executable used for 3D runs
2. case-specific initfile (`ICMSHALADINIT`), containing forcing
3. case-specific namelist, containing forcing assingment

- github repository for the latter two

Czech Hydrometeorological Institute

Technical differences to 3D model: Input

- forcing: upper-air, surface: in `ICMSHALADINIT`
- surface with SURFEX: in `ICMSHALADINIT.sfx`
- no coupling
- one processor, one thread

Technical differences to 3D model: Output

- ICMSH files, not convenient
- LFA files:
  - every timestep
  - more variables
  - easy to add new variables (see later)
  - read with DDH tools (`lfaminm`, `lfac`)

Namelist differences to 3D model

namelist	variable	comment
`&NAMCTO`	`LSFORC=.TRUE.`	upper-air forcing + LFA files
`&NAMCTO`	`LSFORCS=.TRUE.`	surface forcing (with `LMSE=.F.`)
`&NAMDYN`	`NPROMA=-4`	4 points

- simplified dynamics

Changes to namelists between cases

- across namelists:

namelist	variable	description
&NAMGFL	NGFL_FORC	number of upper-air forcings
&NAMPHY	LRAY	switch for radiation (ACRANEB2)
&NAMPHYDS	NSFORC	number of surface forcings
&NAMRIP	CSTOP	duration of simulation
&NEMGEO	RLAT_ACAD	latitude for radiation
&NEMGEO	RLON_ACAD	longitude for radiation

- special namelist &NAMLSFORC

Namelist `&NAMLSFORC`

- `LMUSCLFA=.TRUE.` for LFA output
- forcing and nudging assigned from `ICMSHINIT` file

Czech Hydrometeorological Institute

Forcing assignment in `&NAMLSFORC`

- forcing/nudging stored in `ICMSHINIT` file: variable `SXXXFORCYYYY` (XXX...level or URF, YYYY...number)
- in `&NAMLSFORC` one sets:
  - start of variable's values `SXXXFORCYYYY`: `N<variable>_DEB=YYYY`
  - number of variable's forcing values: `N<variable>_NUM`
  - update frequency: `N<variable>_FREQ`
  - forcing times explicitly: `NL_<variable>_TIME`
- same for surface variables, but they do not have `N<variable>_FREQ`

Upper-air forcing and nudging in `&NAMLSFORC`

- forcing:
  - geostrophic wind, w, ω
  - advections of q_v, T, and wind
- nudging:
  - T, q_v, wind components

Surface forcing in `&NAMLSFORC`

- without SURFEX, surface driven by sensible and latent heat fluxes
- other variables:
  - u_*, T_{surf}
  - Coriolis parameter must be forced extra
  - albedo, emissivity

Creation of new cases

1. select profiles for initial condition (T, q_v, wind, ...)
2. select variables for forcing/nudging
3. determine variables' values at model levels
4. add surface fields
5. create FA file
6. update namelist

Github repository with idealized cases

- currently 6 prepared cases from literature (for turbulence and microphysics)
- scripts, which assign values from paper to model levels
- one can change the number of vertical levels
- private github repo: `https://github.com/DavNemec/musc_cases`
- write me and i will add you :)

Github repository with idealized cases

DavNemec Update README.md c161b2f · 4 days ago 4 Commits

armcu	initial	last week
dycoms2_rf01	initial	last week
dycoms2_rf02	initial	last week
gabls1	initial	last week
mpace	initial	last week
scld	initial	last week
README.md	Update README.md	4 days ago

Database of cases

Name	Last commit message
..	
eta_L87_chmi	initial
make_init_mpace.sh	initial
mpace.f90	initial
mpace.pdf	initial
nam_ascii2fa_mpace	initial
namelist_mpace	initial
surf_fields_mpace.dat	initial

How to create the case?

- each case has `make_init_case.sh`:

```
vertical_levels_file=eta_L87_chmi # file with A and B coefficients

...

# run program for assingment to model levels and output an ASCII file
./case ${vertical_levels_file} > data_case.dat

# create FA file using Ján Mašek's ascii2fa
../ascii2fa nam_ascii2fa_case
```

My approach of assingment to model levels

- from literature (idealized cases):
  1. compute π (hydrostat. pressure) from A and B of η-coordinate:

$$\pi_{jlon} = A_{jlon} + B_{jlon}\pi_s \tag{1}$$

  2. move in small Δp from ground to atmosphere while recomputing $p, z, T, R, q_v, \ldots$ using hydrostatic assumption
  3. if $p < \pi_{jlon}$ of next level is crossed, assign current value on this model level
  4. error less than 0.002% compared to `SUSTA` for std. troposphere
- from radiosonde easy: directly using $p = \pi_{jlon}$

Adding new variables to LFA files

1. put this in the beginning of the routine:
   ```
   USE YOMLSFORC, ONLY: LMUSCLFA, NMUSCLFA
   ```

2. save the variable by putting:
   ```
   IF (LMUSCLFA) CALL WRSCMR(NMUSCLFA,"variable’s output name", &
   & variable,KLON,KLEV)
   ```

Summary (1/4): running MUSC

- for MUSC one needs only:

  1. executable used for 3D experiments
  2. case-specific initfile
  3. case-specific namelist

Summary (2/4): Namelist

- a few changes outside of `&NAMLSFORC`
- `&NAMLSFORC` assigns `SXXXFORCYYYY` from `ICMSHINIT` file
- do not forget to set latlon for radiation
- Coriolis parameter extra in `&NAMLSFORC`

Summary (3/4): Github repository with cases

- on github: `https://github.com/DavNemec/musc_cases`
- easy to change vertical levels
- script: small steps in Δp
- ascii2fa for ASCII$\rightarrow$FA conversion

Summary (4/4): Output

- convenient LFA files
- easy to add variables to them
- use `lfac` for extracting vertical profiles

Thank you for your attention

David Němec
david.nemec@chmi.cz