# Generation of input files for MUSC from real 3D cases

## Ulf Andrae, SMHI

- About file formats

- How 1D columns are created from a 3D file

  - For atmosphere

  - For surfex

- Examine your files with gl

# Why should we recreate 3D in 1D

- Test new ideas and debug on real cases. Either clean or semi-academic.

- A well working 3D to 1D setup is a good way to ensure that MUSC is kept a live between cycles. Should be included in the HARMONIE testbed.

- Approach: Modify gl, the LBC generator and Victorinox in HARMONIE, to produce 1D output.

# FA/lfi files, the short story

- FA (Fichier Arpege) output format of HARMONIE

  - "Pseudo grib"

  - Header with geometrical, time, spectral information

  - Data in grid point or spectral space, compressed or uncompressed

  - Converted to GRIB in HARMONIE

# FA/lfi files, the short story

- LFI (Logiciel de Fichiers Indexés) output format of SURFEX

  - Pure binary format

  - Headers with a lot of SURFEX configuration information

  - Geometrical information

  - Date time information ( only valid time )

  - Uncompressed data with a lot of missing values

  - Converted to GRIB in HARMONIE

# Basic idea of gl input (1/2)

- Find nearest gridpoint for a given lat/lon.

```
OUTGEO%NLON         =    1 ,
OUTGEO%NLAT         =    1,
OUTGEO%NLEV         =   -1,
OUTGEO%PROJECTION =    0,
ORDER               =    0
LINTERP_FIELD       = F,
GPLAT            = $LAT,
GPLON            = $LON,
```

- Select the atmospheric fields

```
READKEY(1:9)%name  = 'SPECSURFGEOPOTEN','SURFPRESSION'     ,'SNNNWIND.U.PHYS',
                     'SNNNWIND.V.PHYS' ,'SNNNTEMPERATURE' ,'SNNNHUMI.SPECIFI',
                     'SNNNLIQUID_WATER','SNNNPRESS.DEPART','SNNNVERTIC.DIVER',
READKEY(1:9)%lll  = 000,000, -1, -1, -1, -1, -1, -1, -1,
READKEY(1:9)%ttt  = 105,105,109,109,109,109,109,109,109,
READKEY(1:9)%nnn  =   1,  1,  1,  1,  1,  0,  0,  0,  0,
```

- And the surface fields ( arome example )

```
READKEY(10:19)%name  = 'SURFAEROS.SEA    ',     'SURFAEROS.LAND  ',
 'SURFAEROS.SOOT  ',      'SURFAEROS.DESERT',     'SURFA.OF.OZONE  ',
 'SURFB.OF.OZONE  ',      'SURFC.OF.OZONE  ',     'SURFTEMPERATURE',
'SURFRESERV.NEIGE',      'SURFIND.TERREMER ',
```

# Basic idea of gl input (2/2)

- Generate the forcing fields as copies of the 1D column

```
PPPKEY%name = '#','SNNNFORC001','SNNNFORC002','SNNNFORC003'
PPPKEY%ttt = 109,109,109,109,
PPPKEY%lll = -1,-1,-1,-1,
PPPKEY%ppp = 32,-1,-1,-1,
PPPKEY%nnn = 0,0,0,0,
PPPKEY%lwrite = F,T,T,T,
IFORCE      = 011,051,032,
```

- What other fields do we use as forcing? What should the forcing look like?

- Select output

  - OUTPUT_FORMAT    = 'MUSC_FORCING_FA', creates an FA file that we can start directly from.

  - OUTPUT_FORMAT    = 'MUSC_FORCING', creates and ascii file that can be given to acadfa.

# Usage and limitations

- Run with

```
gl -l -n namelist_file FAFIILE -o MUSCFILE
```

- Does not work with HIRLAM/ECMWF/ARPEGE input.

- No vertical interpolation is done

- At the moment the subsequent forecast works only with the non surfex run

# SURFEX 3D to 1D

- Start from a SURFEX output file from e.g. an AROME run.

```
OUTPUT_FORMAT = 'MUSC_FORCING_LFI',
OUTGEO%NLON        =    1 ,
OUTGEO%NLAT        =    1,
OUTGEO%NLEV        =   -1,
OUTGEO%PROJECTION =    0,
ORDER              =    0
LINTERP_FIELD     = F,
GPLAT          = $LAT,
GPLON          = $LON,
```

- Pick all information from a gridpoint including the configuration information in the lfi file.

- Change the geometrical variables NDIM_*, XHAT,YHAT,IMAX,JMAX,LATORI,LONORI,LAT0,LON0

# SURFEX 3D to 1D

- Can we turn tiles on and off by setting NDIM_SEA, NDIM_WAT, NDIM_TEB, NDIM_NATURE? ( I think we can )

- Is the geographical/projection values important?

- Use with

```
gl -l -n namelist_file LFIFILE -o MUSCSFXFILE
```

# Status of 3D 2 1D tests

- Running the ARPEGE example from an ALADIN file works!

- Running AROME from a modified 3D namelist crashes

  - Had to turn off the lfa writing.

  - Writes +0000 files but crashes at the first timestep.

# Other ways of using gl

- List the content of a file

  gl -l FAFILE/LFIFILE

- -g also prints the FA/LFI headers

- -m gives min/mean/max values

- -musc converts the file to ASCII ( works both for LFI and FA files )

- -c converts an FA/LFI file to GRIB

  https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/PostPP/gl

# Finally

- Shortcomings with the current approach?

- Suggestions for other ways of using this?

Questions?