

GRIb2v5D USER'S GUIDE

Converting grib files to Vis5d format

Davide Cesari¹
ARPA-SMR, Bologna

July 1, 2005

¹e-mail: cesari@users.sourceforge.net

Contents

Chapter 1

Introduction

1.1 Features

The program grib2v5d converts the content of a grib file into a file that can be viewed with Vis5d, a popular program for 3-D visualisation of meteorological fields. The vertical level types supported on input are sigma, eta (Mesinger), hybrid sigma/pressure (LM type only), height and pressure. The fields can be interpolated, on output, at height levels (interpolation to pressure levels may be added in the future), or can be kept at the same levels as in input (this is especially useful for viewing data at model levels without the filter of interpolation). The horizontal grid should be regular lon/lat, either rotated or not; no horizontal interpolation is made since Vis5d supports directly this type of grid. Optional vertical profiles at specified coordinate in ASCII formatted file can be produced. The program is mainly written in fortran 90 and is derived from a previous version made by Paolo Patruno at ARPA-SMR; it makes use of a Fortran77 library for managing access to grib files, a C library for reading and decoding grib and a C subroutine (modified from the one provided with Vis5d) for writing topography files. The program has been developed in the framework of the COSMO (Consortium for Small scale MOdelling) project (<http://www.cosmo-model.org>) which includes, among its participants, German, Greek, Italian and Swiss meteorological Services.

1.2 Future improvements

Use of DWD routines for accessing grib files. Automatic creation of a namelist from a given grib file. Support for stereographic grids. Support for different vertical coordinates. Porting to Cray. Possibility to display model topography instead of Vis5d internal topography (already done but not working).

Chapter 2

Installing and compiling

The grib2v5d package requires that vis5d and the picogrib and grib_util libraries (provided by ARPA-SMR) be already installed. You can download these libraries from SourceForge following the links at the URLs

http://download.sourceforge.net/gributil/grib_util.tar.gz

and

<http://download.sourceforge.net/picogrib/picogrib.tar.gz..>

For more updated links see also the grib2v5d home page:

<http://grib2v5d.sourceforge.net/>.

2.1 Installing Vis5d

In order to compile grib2v5d, Vis5d (or Vis5d+) sources should be correctly installed under the directory \$VIS5D or \$VIS5D/vis5d-last. If otherwise you have Vis5d+ binaries with libraries and header files installed, then the sources are not needed, provided the configure script can locate the Vis5d+ installation.

2.2 Installing grib libraries

Prior to installing the libraries you should define in your .profile or similar file a UNIX environment variable \$USOHOME pointing to the directory where the libraries will be installed (and where grib2v5d will look for them) and create the directory \$USOHOME/lib. You can then extract the libraries tar archives and build them with the commands

```
gzip -c gributil.tar.gz | tar xvf -; cd gributil; make
```

and

```
gzip -c picogrib.tar.gz | tar xvf -; cd picogrib; make;
```

if all goes the right way you should have your libraries picogrib.a and grib_util.a in the directory \$USOHOME/lib.

2.3 Building grib2v5d

Extract the tar archive `grib2v5d.tar` and enter the newly created directory `grib2v5d`. Since version 2 of `grib2v5d` an autoconfiguration script has been created, dealing with most of the system-dependent problems you may encounter. However you may still have to modify something in the file `Configure.set` according to the instructions inside it, in particular, if you have Vis5d+ correctly installed, you should unset the environment variable `$VIS5D` in order to link with the installed libraries, unless you want to compile the Vis5d sources required in `grib2v5d` and link them statically. Moreover, if your operating system is not one of those that have already been tested with the configuration script, you may have to create a `configure.uname` file based on `configure.template`, containing OS-specific compiler invocation, command line parameters, etc. Type then `./configure` to perform the configuration: a `Make.set` file will be created to be included by the makefiles; type `make` to build the `grib2v5d` executable and, optionally, `make install` to install `grib2v5d` executable and documentation. The autoconfiguration procedure has been tested under AIX-IBM, OSF-Alpha, Linux-i386, Irix-SGI and SUNOS; options for other Unix flavours from users are welcome.

Chapter 3

Running grib2v5d

The program `grib2v5d` reads an unique grib file, looking inside it for fields of variables specified in a namelist file, then outputs these fields in a Vis5d file. The namelist file is named by default `v5d.nam1`. The philosophy of `grib2v5d` is to guess as much information as possible from the grib file, so grid parameters, vertical coordinate type and initial date are all deduced from the content of the grib file. All the other information must be provided in a namelist file described in chapter ??, however, for an initial use, that chapter can be skipped and one can directly refer to chapter ?? and use the standard namelist files provided, keeping most of the configuration to its default status.

3.1 Command-line arguments

The syntax for executing `grib2v5d` is:

```
grib2v5d [-n <namelist file>] [-i <grib file>] [-o <Vis5d file>]  
[-mi <grib file1> [<grib file2> ...]]
```

All arguments are optional, if `-n` is not specified `v5d.nam1` is used as namelist file; if `-i` or `-mi` are not specified, then the input grib file name should be set in the namelist names (see section ??), while if `-o` is not specified, the input file name with suffix `.vis5d` is used; if `-mi` is specified then all the following arguments, possibly specified by shell wildcards, are taken as file names, and each of them must coincide, in the right order, with the time levels specified in the namelist file.

3.2 Format of the grib input file

Since, as explained above, some information is deduced from the content of grib input file, `grib2v5d` makes some assumptions about the correctness of that file. If the input level type (`type1`, see chapter ??) is not specified in the namelist, the grid parameters, the vertical coordinate type and the initial date

are determined from the first grib field of a requested variable at an upper air requested level (i.e. whose grib level type is one of 100, 103, 107, 109, 110, 119) which lies on Arakawa H points. This means that, in order to avoid unpredictable results, all the fields in the grib file should have the same grid (apart from fields on U/V points of an Arakawa C-type grid for which the program makes a proper interpolation), initial date and vertical coordinate type (apart from surface fields). If `type1` is specified, then the field providing the information is looked for according to the previous rule while also matching the vertical level type provided. Once gathered this information from the file, the search of the requested fields is made sequentially with a possible rewind of the file, so the order of the fields inside the grib file is unimportant (provided that there is a unique field matching each set of parameter-level-date-forecast time in the file). For accumulated fields the forecast time matched is the one relative to the end of the accumulation period. When a requested field is not found in the file, a tentative explanation of what didn't match is printed by the program.

Chapter 4

Input namelist and files

The `grib2v5d` namelist file contains two namelists: `config`, which determines the conversion procedure, and `names`, which specifies the input and output file names. Most of the namelist variables have a default value so that they may be not included in the file, but some of them are mandatory (see section ??).

4.1 The namelist config

The namelist config can contain the following scalar variables:

Name	Type	Meaning	Default
numtimes	INT	number of time levels in output	deduced from vector tabvar
numvars	INT	number of variables in output	deduced from vector tabfth
compressmode	INT	Vis5d compress mode (number of bytes per datum, 1, 2 or 4)	1
typel	INT	input grib vertical level type for upper air fields	deduced from grib file
vctype	INT	input grib type of hybrid vertical coordinate	deduced from grib file
redfact	INT	linear reduction factor for the grid	1
fixedfth	INT	<0 to operate in “forecast mode”, ≥0 to operate in “analysis mode”	-1
itt	INT	position of temperature field in variables list	0
iqq	INT	position of specific humidity field in variables list	0
ivu	INT	position of x component of wind field in variables list	0
ivv	INT	position of y component of wind field in variables list	0
ivw	INT	position of z component of wind field in variables list	0
ips	INT	position of surface pressure field in variables list	0
ipt	INT	position of top-of-atmosphere pressure field in variables list	0
ipp	INT	position of pressure field in variables list	0
ipd	INT	position of pressure departure field in variables list	0
igo	INT	position of surface height or geopotential field in variables list	0
ire	INT	position of reciprocal of sigma field in variables list	0
izh	INT	position of half model level height (for output) in variables list	0
izf	INT	position of full model level height (for output) in variables list	0

Name	Type	Meaning	Default
ptdef	REAL	top-of-atmosphere pressure value if not found in grib (Pa)	10000
fill_below	LOG	TRUE to fill below ground the fields defined up to the ground level	TRUE
noproj	LOG	TRUE to avoid the use of Vis5d projection and define a generic horizontal coordinate	FALSE
invertl	LOG	TRUE to scan the vertical level list backwards while reading input file	TRUE
sounding	LOG	TRUE to produce ASCII formatted sounding output file	FALSE

The values for `type1` are those from grib code table 3, so 100 for pressure levels, 103 for height levels, 107/108 for sigma levels/layers, 109/110 for hybrid levels/layers or 119/120 for eta levels/layers. If the value is not provided it is taken from the first upper-air field on H points found in the grib file (see next section). When temperature and specific humidity are used for computing geopotential height in the cases of sigma/eta/hybrid coordinates, the temperature is assumed to be on layers, regardless of what is specified in the namelist (this is generally true for the usual vertical staggering schemes used in numerical models).

The types of hybrid coordinates supported are standard pressure-normalised (like Lokal Modell, `vctype=2`) and Simmonds-Burridge surface pressure-normalised (like ECMWF model, `vctype=1`). If not specified `vctype` is automatically guessed with an heuristic algorithm.

Set `redfact` to a value greater than 1 in order to produce a smaller output and a faster display at the expense of a poorer resolution.

The variables `itt`, ... `ire` indicate the position, in the vector of model variables, of some key parameters; if their value is 0 it means that the corresponding parameter is not present.

The flag `noproj` disables the interpretation of the geographic projection of original data, so that a generic rectangular system is used; moreover a Vis5d topography file with the model orography is generated.

The namelist `config` can contain the following vectors dimensioned as `numvars`:

Name	Type	Meaning	Default
varname	CHAR()	variable names for Vis5d	none
tabvar	INT()	grib parameter indicator	none
tab2ver	INT()	grib table 2 version	-1
nl	INT()	number of input vertical levels for each variable	none
lowlev	INT()	lowest input level -1 for each variable	none
nzlev	INT()	number of output vertical levels for each variable	n. of elements in vert_args or 1 if surface field
lowzlev	INT()	lowest output z level -1 for each variable	contents of lowlev
gridtype	INT()	type of C-grid point for each variable (0=H, 1=U, 2=V)	0
cumulate	INT()	>0 to accumulate each field from beginning of fore- cast, <0 to subtract each field from the previous in time	0
const	LOG()	TRUE for fields that are constant in time	FALSE
input	LOG()	TRUE for fields to be read from grib, FALSE for fields to be computed	TRUE
output	LOG()	TRUE for fields that should be output	TRUE

The names of the variables defined in the vector `varname` can take any value, however, to make sure that Vis5d recognizes wind and thermodynamic variables for further computation, the main variable names should have a standard name as 'T', 'Q', 'U', 'V', 'W'.

The variable `tab2ver` can be left to its default value of -1 if a unique version of table 2 is used throughout the file (this should be usually be the case).

If the vector `nzlev` is not provided (i.e. all its elements are 0) then the output vertical level type and values of levels (`vert_args`) are assumed to be the same as those for input levels and no interpolation is made.

The variable `const` should be set to true for fields which are constant in time, in order to avoid reading them more than once (it is automatically set to true for the model orography field).

The namelist `config` can contain the following vector dimensioned as `numtimes`:

Name	Type	Meaning	Default
tabfth	INT()	forecast times to read from grib in hours	0
tabftd	INT()	forecast times to read from grib in days	0

If `numtimes` is not specified, the number of time levels is determined from the number of elements in `tabfth` or `tabftd` vectors.

If `fixedfth`<0, the elements of `tabfth` and `tabftd` vectors are treated as forecast times starting from the same initial date (default); if `fixedfth`≥0 then the elements of `tabfth` and `tabftd` vectors are considered as offsets for computing the emission date from the initial date while the forecast time searched for is +`fixedfth` hours (mainly useful for displaying analysis data with `fixedfth`=0). The

emission date is determined from the first valid grib field found in the input grib file.

The namelist `config` can contain the following vectors dimensioned as the maximum number of input and output vertical levels respectively:

Name	Type	Meaning	Default
<code>tablev</code>	<code>INT()</code>	input vertical levels coordinate values to be found in grib	none
<code>vert_args</code>	<code>REAL()</code>	values of output vertical levels (in Km if height)	none

4.2 The namelist names

The namelist names, can contain the following scalar variables:

Name	Type	Meaning	Default
<code>inname</code>	<code>CHAR</code>	Name of grib input file	none
<code>outname</code>	<code>CHAR</code>	Name of vis5d output file	grib input file name with suffix <code>.vis5d</code>

This namelist is optional; values for the file names can be given also as command-line arguments (see section ??) if enabled in compilation, so that the namelist is read only if `inname` is not given on the command line.

4.3 Mandatory variables in the namelist

A minimal namelist file should contain at least the following variables: `varname`, `lowlev`, `nl`, `tabfth`, `tablev`; moreover, if the fields should be interpolated on height levels, also `vert_args` and `igo` should be specified and the grib should contain the orography field (either as height or geopotential).

4.4 The input file `SOUNDSTAZ.DAT`

This file will be used if the variable `sounding` is enabled in namelist `config`. It describes how many files will be produced and the coordinate for vertical profiles.

```

3
SMRER
PMPMO
PMPVE
1      101  45.02775      9.666667      -1.000000      Piacenza
1      102  44.80000     10.333333      -1.000000      Parma
1      103  44.66667     10.91667      -1.000000      Modena
1     16429  37.91667     12.500000     14.000000      TRAPANI/BIRGI

```

1	16560	39.25000	9.066667	5.000000	P CAGLIARI/ELMAS
2	1	44.65164	10.92667		
2	3	44.54083	10.85389		
3	1	45.45306	12.24167		
3	2	45.49581	12.24444		
4	1	43.60889	10.89862		
4	2	43.60250	11.24366		
4	3	43.73079	11.07557		
4	4	43.72385	11.42131		
4	5	43.85882	10.90674		
4	6	43.85241	11.25326		

1. the first record describes how many files will be created
2. the second up to the Nth are the prefixes for the name of the files
3. all remaining records describe the virtual sounding stations
4. the first field is the ordinal number of the file in output
5. the second is a fictitious station number
6. third is the latitude
7. fourth is the longitude
8. all the following fields are optional description field like height and name

Chapter 5

Optional output files

5.1 The optional sound output files XXXSOUNDLMM.YYYYMMDDHH

Those files contain the vertical profiles calculated on virtual station point.

Example:

LMM	05	03	2002	00	01	01	21600
	1	45.9330		7.8500			
36	699.8	2999.	257.4	1.076	-1.3	0.3	-0.016
35	697.2	3027.	263.8	2.551	-2.6	0.6	-0.025
34	691.5	3091.	264.7	2.471	-4.0	2.3	-0.034
33	684.6	3169.	264.4	2.299	-4.1	3.5	-0.033
32	676.4	3262.	264.2	2.134	-4.4	4.7	-0.027
31	667.0	3370.	264.1	1.899	-4.7	6.3	-0.016
30	656.4	3495.	263.6	1.707	-4.8	7.2	-0.016
29	644.8	3633.	263.6	1.446	-5.1	8.3	-0.018
28	632.6	3780.	262.9	1.333	-4.8	9.1	-0.020
27	619.8	3937.	262.6	1.196	-4.5	9.9	-0.024
26	606.1	4109.	261.7	1.165	-3.7	10.6	-0.024
25	590.9	4304.	260.7	1.198	-3.0	11.4	-0.019
24	573.8	4528.	259.4	1.312	-2.2	12.3	-0.009
23	554.8	4783.	257.5	1.453	-1.4	13.3	0.003
22	534.2	5067.	255.5	1.526	-0.7	14.4	0.023
21	512.7	5374.	253.2	1.459	-0.1	15.3	0.052
20	490.5	5701.	250.6	1.102	0.8	15.6	0.079
19	467.6	6050.	248.1	0.660	1.7	15.3	0.089
18	444.4	6418.	245.1	0.390	1.6	14.5	0.067

```

17  421.1  6802.  241.9  0.338    1.5   13.1  0.021
16  397.8  7204.  238.4  0.329    2.5   12.6  0.003
.....
.....

```

In simbolic form for two stations, two forecast times:

```

code  idd imm iyy ihh  icheck ista idtch
istaz rlatstaz rlonstaz
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
.....
istaz rlatstaz rlonstaz
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
.....
code  idd imm iyy ihh  icheck ista idtch
istaz rlatstaz rlonstaz
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
.....
istaz rlatstaz rlonstaz
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
il rp rz rt rq ru rv rw
.....

```

Legend:

code	ascii description field
idd	day reference time of data
imm	month reference time of data
iyy	year reference time of data
ihh	Hour reference time of data
icheck	checkpoint numer
ista	number of station written every checkpoint
idtch	time step from checkpoint
istaz	virtual station number (false station)
rlatstaz	latitude virtual station number
rlonstaz	longitude virtual station number
il	layer number
rp	pressure at layer (Hpa)
rz	height at layer (m.)
rt	temperature at layer (K.)
rq	specific humidity at layer (g./Kg.)
ru	U component of wind at layer (m./s.)
rv	V component of wind at layer (m./s.)
rw	vertical velocity at layer (m./s.)

Chapter 6

Discussion about vertical levels

Since Vis5d is designed mainly for dealing with 3-dimensional data, some problems arise when 2-dimensional fields are considered, especially if they don't lie on a coordinate Vis5d surface, such as surface fields. For this reason an additional level should be added at the beginning of the input level list (`tablev`) to specify the ground; in terrain-following coordinates this should coincide with a coordinate surface having a value of 1, while for pressure or height surface this should have a fictitious value, maybe 0.

In the case of terrain-following coordinates, fields that are stored in the lowest level can be either (near) surface fields (with `nl=1` and `lowlev=0`) or lowest model level fields (with `lowlev=0`); this also allows to mix variables defined only in the upper-air levels, such as temperature of wind, with the corresponding near-surface variable, such as 2m temperature or 10m wind. Moreover, if the flag `fill_below` is enabled (and it is by default) all the fields that are defined at the model surface are extrapolated with constant value below the orography up to the sea level, so that their surface value can be visualized in a horizontal cross-section at 0 height.

Chapter 7

Quick guide

If all this seems too long to deal with, then you can use one of the namelists already provided with the package:

- `hybht.nam1` for converting fields from Lokal Modell vertical hybrid coordinate system to Vis5d height system
- `hybhyb.nam1` for viewing Lokal Modell hybrid vertical coordinate surfaces directly into Vis5d
- `sigmaht.nam1` for converting fields in sigma vertical coordinate system to Vis5d height system
- `etaht.nam1` for converting fields from Mesinger-Eta vertical coordinate system to Vis5d height system
- `prespres.nam1` for viewing generic pressure coordinate data directly into Vis5d
- `prespresec.nam1` for viewing ECMWF pressure coordinate data directly into Vis5d
- `complete.nam1` is a version of `hybht.nam1` which, in addition, explicitly contains all the variables that do not need to be changed from their default value for a standard use
- `lmsound.nam1` is a version which, in addition, produces additional files for vertical sounding emulation. File are ASCII formatted with vertical profiles at specified coordinates.

For Lokal Modell `hybht.nam1` should be the best one, so copy it to another file named `v5d.nam1`, edit this file in the following way:

- insert into `tabfth` vector the list of forecast times (in hours) expected in the grib file
- set the character variable `inname` to the name of the grib input file available
- set the character variable `outname` to the name of the vis5d output file desired

and execute `grib2v5d`. After some time and many lines of information from the program you should get your Vis5d data in the output file previously specified.