BASICS of NWP

&

Practices at TSMS

Meteorological Data Formats

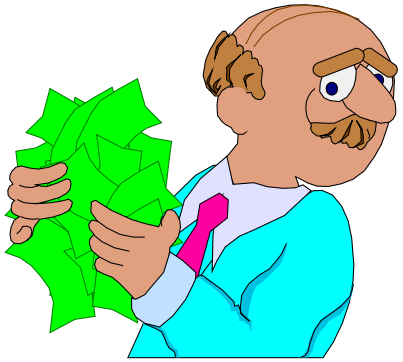
(netCDF / GRIB2 / BUFR – CREX)

Tayfun DALKILIÇ
Meteoroloji Müh.
tdalkilic@meteor.gov.tr

Before Meteorological code forms!..

Global Observations ***GOS*** ⇒ *Global Telecominication* ***GTS*** ⇒ *Global Data Processing System* ***GDPS***

Forecasters



Observers

ASCII code forms!...

Global Observations ***GOS*** ⇒ *Global Telecominication* ***GTS*** ⇒ *Global Data Processing System* ***GDPS***

Observers

Forecasters

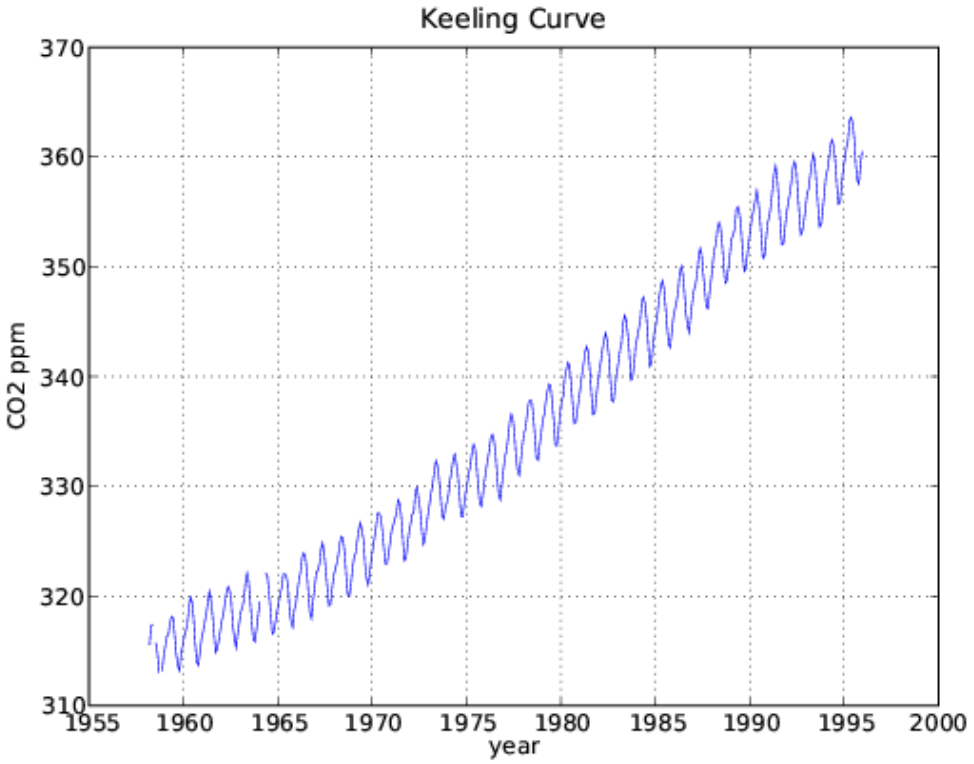
BINARY kod formları!...

Global Observations ***GOS*** ⇒ *Global Telecominication* ***GTS*** ⇒ *Global Data Processing System* ***GDPS***

Forecasters

Automatic Observations and Automatic Coding

Overview of data formats



Time Series
(CO2 concentration)

ASCII

1960	320 ppm
1965	340 ppm
.	.
.	.
.	.
1995	360 ppm
2000	365 ppm

a) Unix or Dos ?

b) JUN1995 or 061995 ?

c) Missing data ?

d) Header or README ?

e) File size and transfer ?

Overview of data formats

BINARY

- *1025 shown as "4-byte integer"*

  00000000 00000000
  00000100 00000001 (little endian)

  00000001 00000100
  00000000 00000000 (big endian)

- 32 bit ==> 0 - 4,294,967,295

  64 bit ==> 0 - 18,446,744,073,709,551,615

a) Little-endian or Big-endian ?
b) Formated or Unformated ?
c) 32-bit or 64-bit ?

TDCF

(Table Driven Code Forms)

GRIB / BUFR / CREX

Problems:

- The increasing size of data
- Precision of data (T= 273.1523...)
- Complex model and observation data
- Forms of the old code is a fixed format
- More frequent observation data (min, sec etc.)

Solutions:

- Self Describing feature
- Flexibility
- Extensibility
- Data compression

TDCF

- **GRIB** – **GRI**dded **B**inary
    - Forecast , Analyse, Climate, Satelite data
- **BUFR** - **B**inary **U**niversal **F**orm for the **R**epresentation of meteorological data.
    - Observations and Satelite
- **CREX** – **C**haracter form for the **R**epresentation and **EX**change of meteorogical and other data.
    - ASCII

TDCF

- **Definition:**
  GRIB/BUFR/CREX
- **Header:** Tarih, zaman, tablo versiyonu...
- **Optional section:**
  Metadata (XML ...)
  ...
- **Data description:**
  Explanation about “Data”
- **Data:** “Data stored here”
- **End:** “7777”

NETCDF

(NEtwork Common Data Format)

NETCDF

(Network Common Data Format)

- **Self-Describing:**
- **Portable:**
- **Direct-access:**
- **Machine-independent:**
- **Appendable:**
- **Sharable:**
- **Archivable:**

NETCDF

(Network Common Data Format)

a) Source Code: http://www.unidata.ucar.edu/software/netcdf/

b) Fortran, C/C++, Perl, MATLAB, Python, Java

c) NetCDF interfaces and programs:

i. ncview

ii. VisAD

iii. PyNGL and PyNIO

iv. CDO (Climate Data Operators)

v. MM5toNetcdf

vi. WRF

NETCDF (New Record)

NF90_CREATE ! - *Create a new record*

...

NF90_DEF_DIM ! - *Defining the size of data (i, j)*

...

NF90_DEF_VAR ! - *Definition of variables (name, type, etc.)*

...

NF90_ENDDEF ! - *End of definition section*

...

NF90_PUT_VAR ! - *Assigning variables*

...

NF90_CLOSE ! - *Saving the file and closed*

TEXT > FORTRAN 90 > netCDF

```
program netcdf_yaz
  use netcdf
  implicit none
  character (len = *), parameter :: FILE_NAME = "sample.nc"
  ! 6 x 12 grid.
  integer, parameter :: NDIMS = 2
  integer, parameter :: NX = 6, NY = 12
  integer ::retval,x,y,x_dimid, y_dimid
  integer :: ncid, varid, dimids(NDIMS)
  integer :: data_out(NY, NX)
    do x = 1, NX
      do y = 1, NY
        data_out(y, x) = (x - 1) * NY + (y - 1)
      end do
    end do
  retval=nf90_create(FILE_NAME, NF90_CLOBBER, ncid)
    retval=nf90_def_dim(ncid, "x", NX, x_dimid)
    retval=nf90_def_dim(ncid, "y", NY, y_dimid)
    dimids =  (/ y_dimid, x_dimid /)
    retval= nf90_def_var(ncid, "data", NF90_INT, dimids, varid)
  retval= nf90_enddef(ncid)
  retval= nf90_put_var(ncid, varid, data_out)
  retval= nf90_close(ncid)
end program netcdf_yaz
```

description section

Fortran 90 Source code

gfortran -ffree-form write-netcdf.f -o write-netcdf.x -I/usr/local/include -lnetcdf

NETCDF (Old Record)

```
NF90_OPEN !             - Open existing NetCDF record
   ...
   NF90_INQ_DIMID !  - Inquire
   ...
   NF90_INQ_VARID ! - Inquire
   ...
   NF90_GET_VAR !  - Read the data
  ...
NF90_CLOSE !           - Close the record
```

netCDF > FORTRAN 90 > TEXT

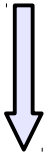
```
program read_netCDF
   use netcdf
   implicit none
   character (len = *), parameter :: FILE_NAME = "deneme.nc"
   !6 x 12 grid.
   integer, parameter :: NX = 6, NY = 12
   integer :: data_in(NY, NX)
   integer :: ncid, varid,retval
   integer :: i,j

   retval=nf90_open(FILE_NAME, NF90_NOWRITE, ncid)
   retval=nf90_inq_varid(ncid, "data", varid)
   retval=nf90_get_var(ncid, varid, data_in)
   retval=nf90_close(ncid)

   print*,data_in

end program read_netCDF
```

Fortran 90 kaynak kod

gfortran -ffree-form read-netcdf.f -o read-netcdf.x -I/usr/local/include -lnetcdf

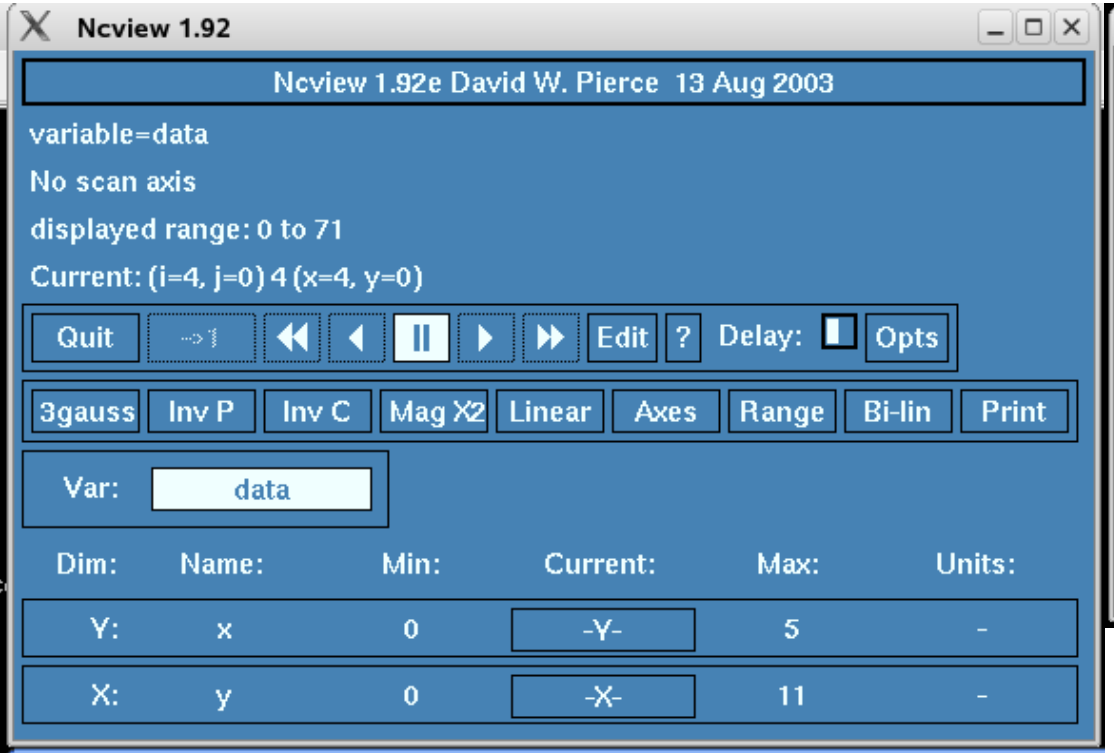
netCDF > PYTHON (pyNGL)

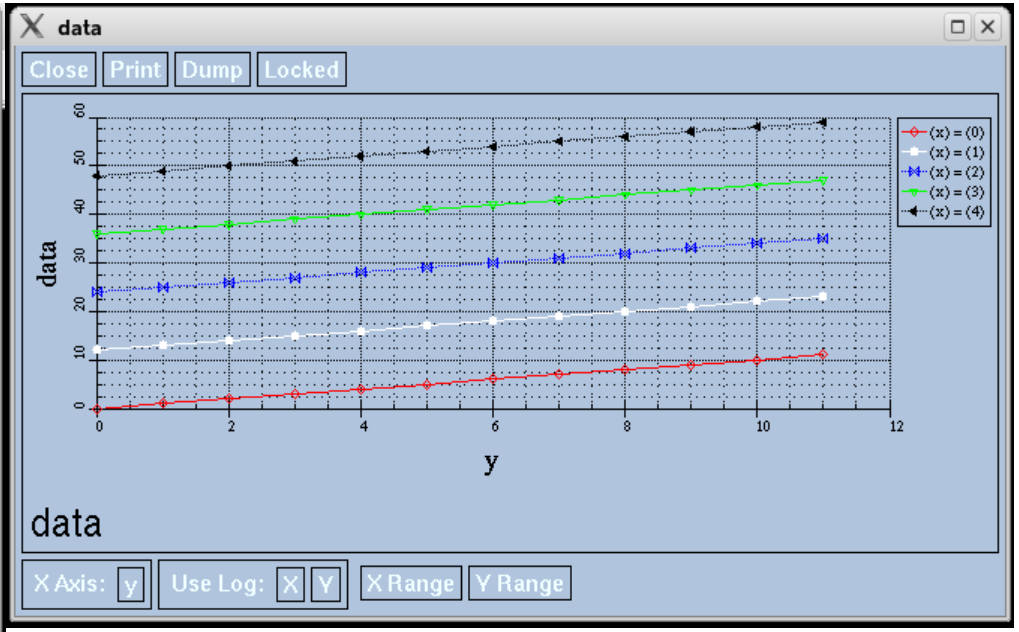
```
#!/usr/bin/python
import Numeric,Nio
netcdf_dosya=Nio.open_file("deneme.nc","r")
veri=netcdf_dosya.variables["data"][:,:]
print veri
```

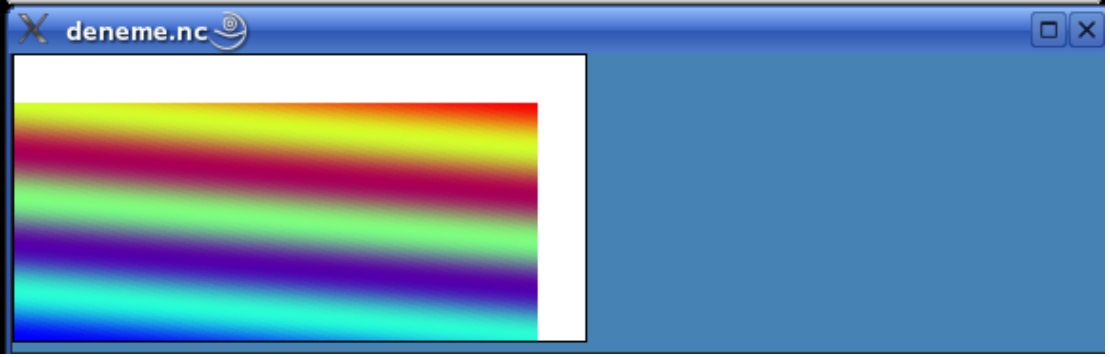


```
[[ 0  1  2  3  4  5  6  7  8  9 10 11]
 [12 13 14 15 16 17 18 19 20 21 22 23]
 [24 25 26 27 28 29 30 31 32 33 34 35]
 [36 37 38 39 40 41 42 43 44 45 46 47]
 [48 49 50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69 70 71]]
```

netCDF > NCVIEW








```
[[ 0  1  2  3  4  5  6  7  8  9 10 11]
 [12 13 14 15 16 17 18 19 20 21 22 23]
 [24 25 26 27 28 29 30 31 32 33 34 35]
 [36 37 38 39 40 41 42 43 44 45 46 47]
 [48 49 50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69 70 71]]
```

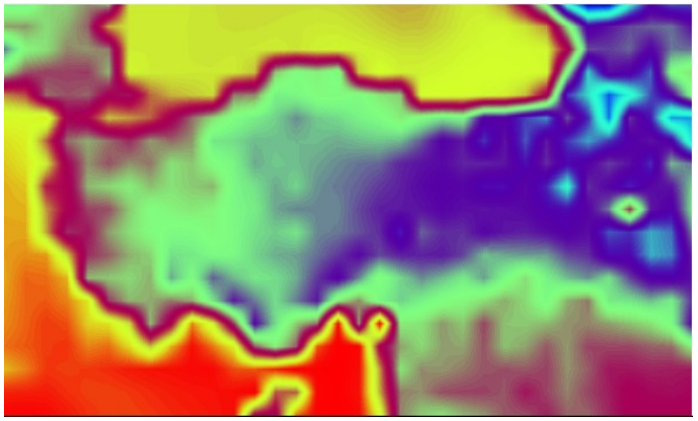
WRF

(Weather Research & Forecast)

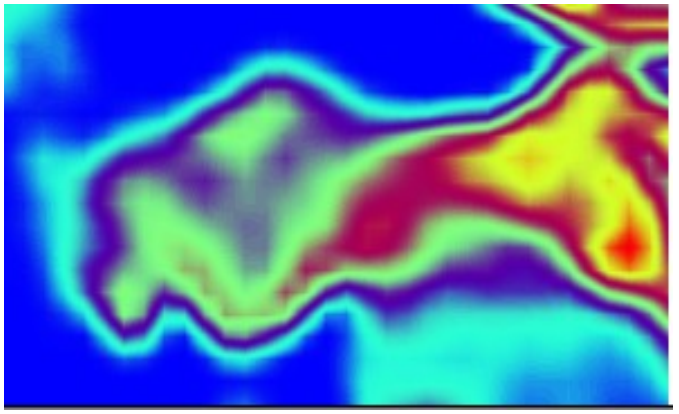
The Weather Research and Forecasting (WRF) Model is a next-generation mesocale numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs

WRF sample maps

SST



LAND USE



GRIB2

(**GRI**ded **B**inary)

GRIB Edition 2

GRID, GRIB, GRIB2

- **GRID**,
  - ASCII
- **GRIB**, binary
  - Edition 0, 1985
  - Edition 1, 1990
- **GRIB** Edition 2 (i.e. GRIB2), EPS data, ...

Why GRIB2 ?

- Archiving and data transfer:
- Multi-dimensional data
- Long periyodical data (iklim modelleri)
- EPS
- “missing data” is not available
- Floating point (IEEE) is mising
- No support for small “time-steps”
- No support for Cross-sections
- No support for Hovmöller Diagramları

GRIB1

- Section 0 Indicator section: “GRIB”
- Section 1 Product definition section
- Section 2 Grid description section
- Section 3 Bit map section
- Section 4 Binary data section
- Section 5 End section: “7777”

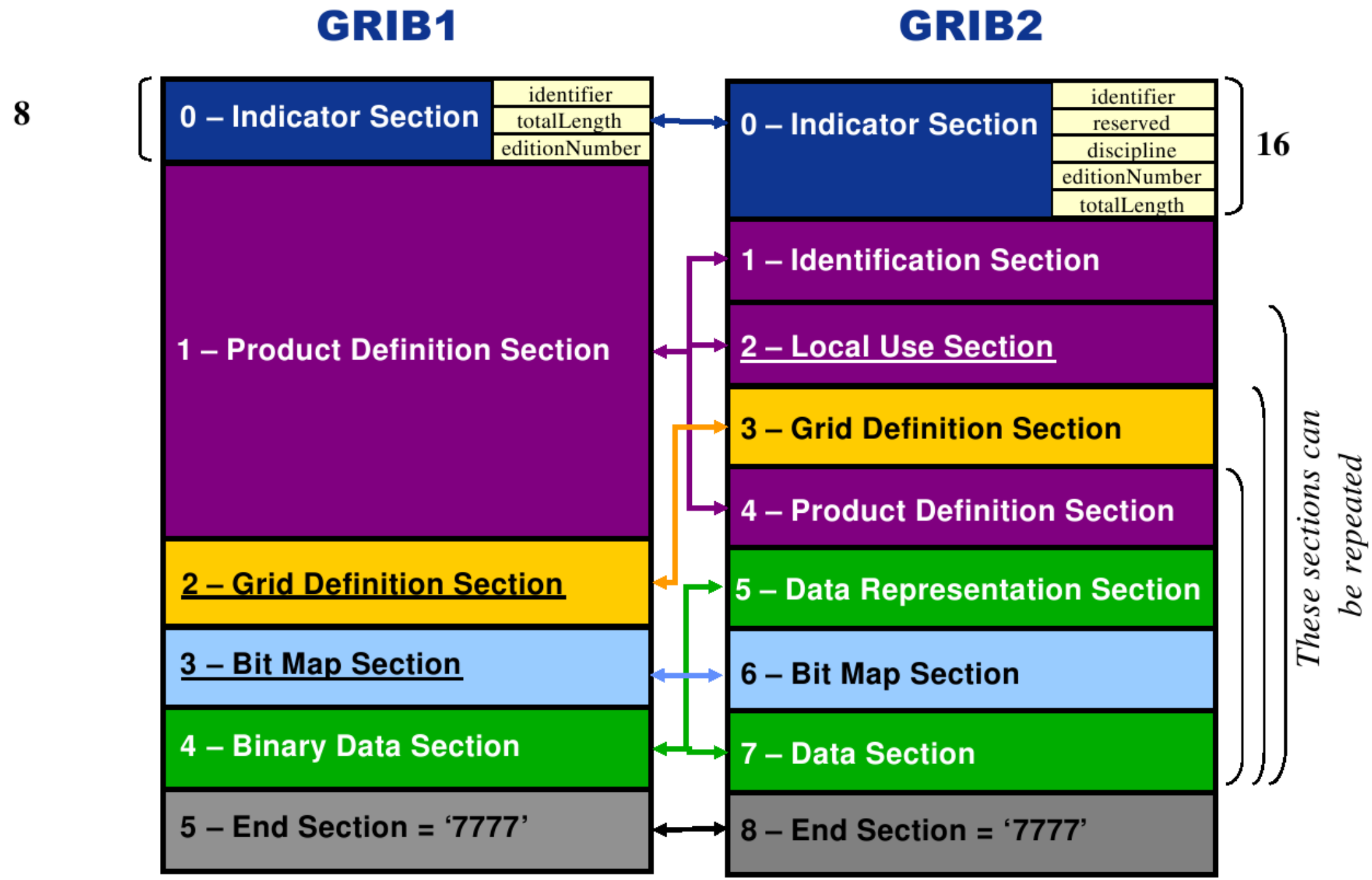
GRIB1 Operational Usage:

GTS
World Area Forecast Centres (WAFC) – ICAO
ECMWF
Archiving
etc....

GRIB2

Section Number	Section Name	Section Contents
Section 0:	Indicator Section	"GRIB", Discipline, GRIB Edition number, length of message
Section 1:	Identification Section	Length of section, section number, characteristics that apply to all processed data in the GRIB message
Section 2:	Local Use Section (optional)	Length of section, section number, additional items for local use by originating centres
Section 3:	Grid Definition Section	Length of section, section number, definition of grid surface and geometry of data values within the surface
Section 4:	Product Definition Section	Length of Section, section number, description of the nature of the data
Section 5:	Data Representation Section	Length of section, section number, description of how the data values are represented
Section 6:	Bit-Map Section	Length of section, section number, indication of presence or absence of data at each grid point, as applicable
Section 7:	Data Section	Length of section, section number, data values
Section 8:	End Section	"7777"

GRIB2


GRIB1
GRIB2
8
0 – Indicator Section
identifier
totalLength
editionNumber
1 – Product Definition Section
2 – Grid Definition Section
3 – Bit Map Section
4 – Binary Data Section
5 – End Section = '7777'
0 – Indicator Section
identifier
reserved
discipline
editionNumber
totalLength
16
1 – Identification Section
2 – Local Use Section
3 – Grid Definition Section
4 – Product Definition Section
5 – Data Representation Section
6 – Bit Map Section
7 – Data Section
8 – End Section = '7777'
These sections can be repeated

– Section 0

- Section 0: Indicator bölümü
  - Octets 1 - 4: “GRIB” in ASCII (i.e. 71 82 73 66)
  - Octets 5 - 6: Reserved (normally set to 0)
  - Octet 7: Discipline (CT 0.0, e.g. 3 for space)
  - Octet 8: GRIB edition number (2)
  - Octets 9 - 16: Total length in octets

– Section 1

- Section 1 Identification bölümü
  - Octets 1 - 4: Section length
  - Octet 5: Section number (1)
  - Octets 6 - 9: Originating centre and sub-centre
  - Octets 10 - 11: Master and local table version numbers
  - Octet 12: Time significance
    (CT 1.2, e.g. 3 for observation time)
  - Octets 13 - 19: Date / time
  - Octet 20: Production status of data
    (CT 1.3, e.g. 0 for operational data)
  - Octet 21: Type of data
    (CT 1.4, e.g. 7 for processed radar observations)

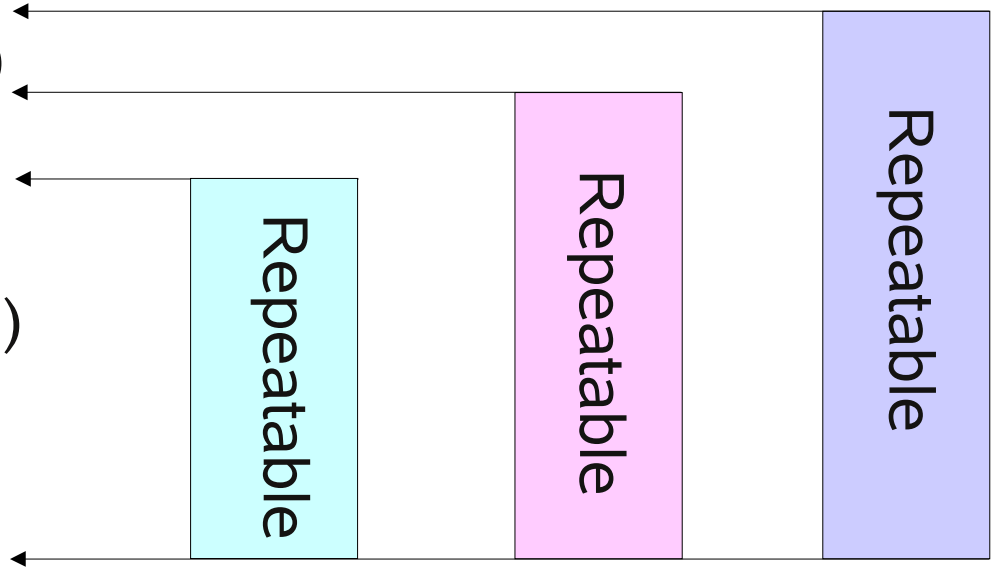
–

– Repetition

GRIB2 some of the sections can be repeated

(no support in GRIB1)

- Section 0 (Indicator)
- Section 1 (Identifier)
- Section 2 (Local use)
- Section 3 (Grid)
- Section 4 (Product)
- Section 5 (Data Rep^{n})
- Section 6 (Bit map)
- Section 7 (Data)
- Section 8 (Ending)



– Repetition

- Section 0 (Indicator)
- Section 1 (Identifier)
- Section 3 (Grid)
- Section 4 (Product)
- Section 5 (Data Rep^{n})
- Section 6 (Bit map)
- Section 7 (Data)

0600Z

- Section 4 (Product)
- Section 5 (Data Rep^{n})
- Section 6 (Bit map)
- Section 7 (Data)

1200Z

- Section 4 (Product)
- Section 5 (Data Rep^{n})
- Section 6 (Bit map)
- Section 7 (Data)

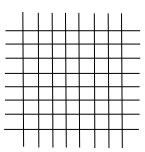
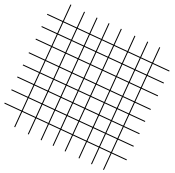
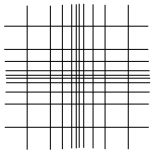
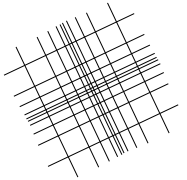
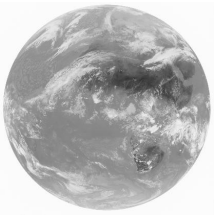
1800Z

Repetition 4-7

- Section 8 (Ending)

– Section 3

- Grid definitions (GRIB1, Section 2)
- Grid Definition Template (GDT)
- GDT
  - 3.0: Latitude/Longitude (or equidistant cylindrical, or Plate Carrée)
  - 3.1: Rotated Latitude/Longitude (or equidistant cylindrical, or Plate Carrée)
  - 3.2: Stretched Latitude/Longitude (or equidistant cylindrical, or Plate Carrée)
  - 3.3: Stretched and Rotated Latitude/Longitude (or equidistant cylindrical, or Plate Carrée)
  - 3.20: Polar Stereographic
  - 3.90: Space view perspective, or orthographic

– Section 4

- *Product Definition Section* (GRIB1, Section 1)
- PDT *(Product Definition Templates)*
  - 4.0: Analysis or forecast on horizontal level or layer at a point in time
  - 4.1 to 4.4: Various information pertaining to ensemble forecast systems
  - 4.5: Probability forecast on horizontal level or layer at a point in time
  - 4.7: Analysis or forecast error on horizontal level or layer at a point in time
  - 4.20: Radar products
  - 4.30: Satellite products
- Hybrid sigma levels.

– Section 5

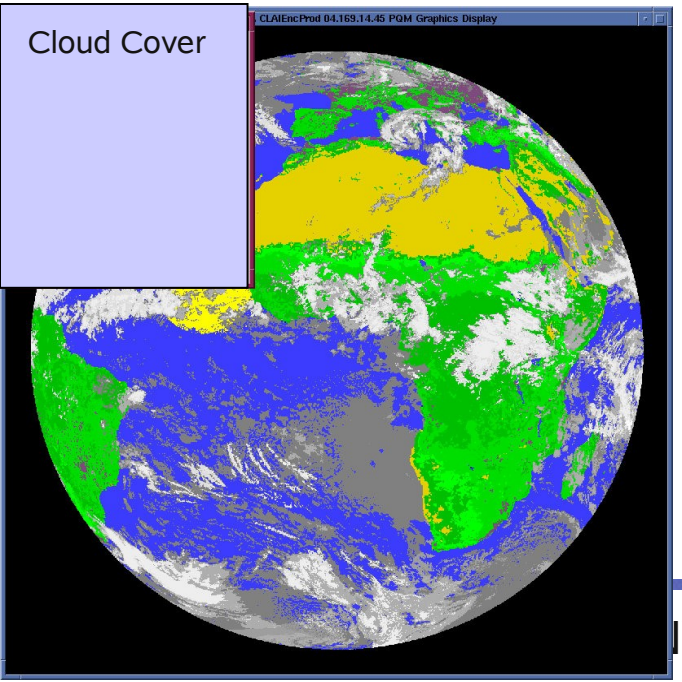
- Data representation (GRIB1, Section 4 ile)
- Some DRTs have been developed, *e.g.*
  - 5.0: Grid point data - simple packing
  - 5.1: Matrix values at grid point - simple packing
  - 5.2: Grid point data - complex packing
  - 5.3: Grid point data - complex packing and spatial differencing
  - 5.50: Spectral data - simple packing
  - 5.51: Spherical harmonics data - complex packing
- Section 5 also gives total number of data values to be found in Section 7

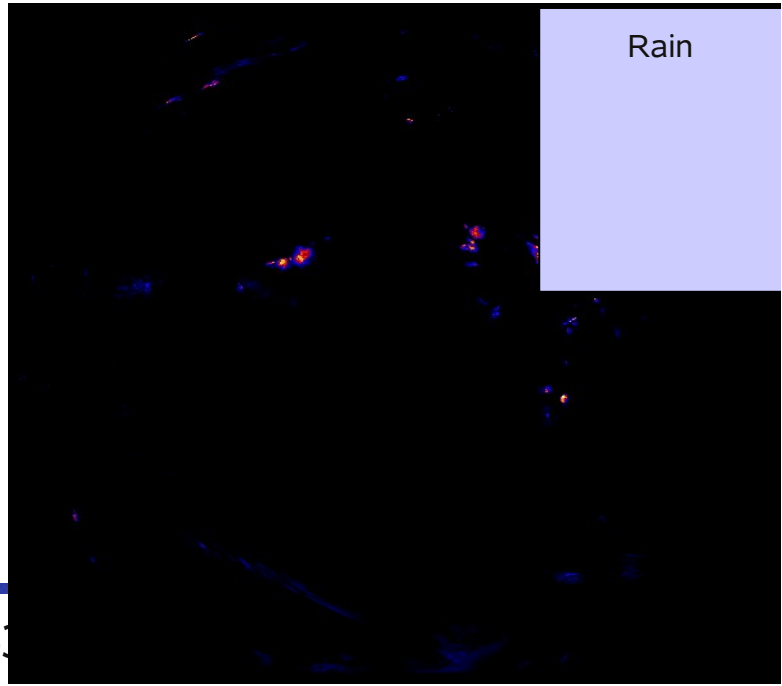
- **Section 6**

- Section 6: Bit map bölümü
  - Octets 1 - 4: Length of section
  - Octet 5: Section number (6)
  - Octet 6: Bit map indicator (CT 6.0)
  - Octets 7 - xx: Bit map (if present)

- **Section 7**
- *"DATA"*
- **Section 8**
  - *"7777"*
  -
- **- END -**

Cloud Cover





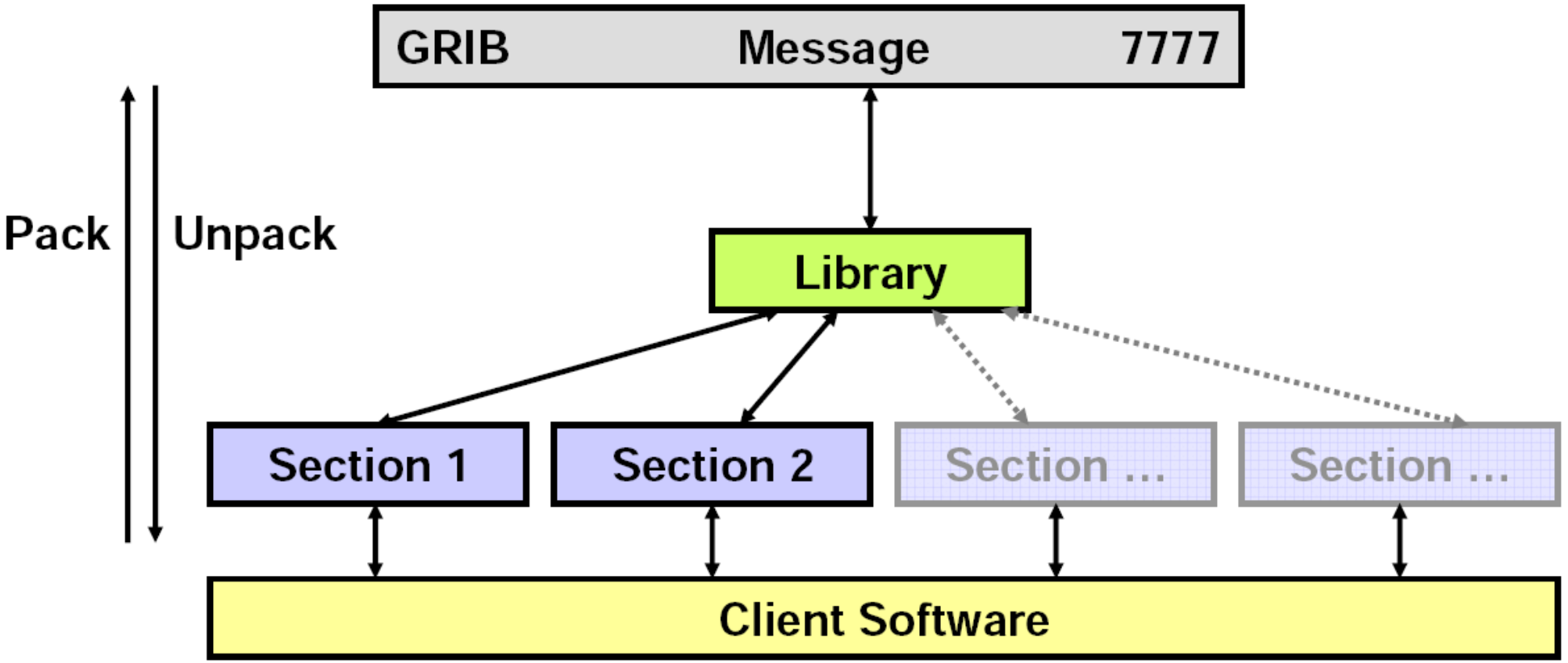
Rain

FM92 GRIB, Edition 2, Code Form and Tables

http://www.wmo.ch/web/www/DPS/grib-2.html

⇩

FM92-GRIB2-11-2003.pdf



“SIMPLE” and “COMPLEX” PACKING (grib2)

- Simple Packing:
  - $Y \times 10^D = R + (X1 + X2) \times 2^E$

- Complex Packing
  - $Y \times 10^D = R + (X1 + X2) \times 2^E$

– WGRIB2

Source Code:

http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/

Supported Platforms:

- Llinux x86_32 (Redhat), x86_64 (SUSE 10.1,gcc-4.1.0)
- AIX
- Solaris, “gnu make” ve “gcc” ye ihtiyaç var
- XP, cygwin kullanarak
-

Sample:

```
user@hostname:~> wgrib2 sample.grib2
1.1:0:d=2000070312:HGT:500 mb:24 hour fcst:
1.2:0:d=2000070312:TMP:850 mb:24 hour fcst:
1.3:0:d=2000070312:UGRD:850 mb:24 hour fcst:
1.4:0:d=2000070312:VGRD:850 mb:24 hour fcst:
1.5:0:d=2000070312:APCP:surface:6 hour acc,18 hour+ fcst:
```

Samples:...

1-) wgrib2 -center sample.grib2

2-) wgrib2 -ftime sample.grib2

3-) wgrib2 -nxny sample.grib2

4-) wgrib2 -T sample.grib2

5-) wgrib2 -Sec0 sample.grib2

6-) wgrib2 -text_col 129 -text deneme.txt sample.grib2

7-) wgrib2 sample.grib2 | grep ":APCP:" | wgrib2 -i

sample.grib2 -text_col 129 -text dump.txt

– PYGRIB2

Source Code:

http://www.cdc.noaa.gov/people/jeffrey.s.whitaker/python/grib2/

Supported Platforms:

- Python 2.3 veya daha üst bir versiyon
- numpy
- pyproj
-

Samples:

```
user@hostname:~> python
Python 2.4 (#1, Mar 22 2005, 21:42:42)
[GCC 3.3.5 20050117 (prerelease) (SUSE Linux)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import grib2
>>> grb = grib2.Grib2Decode('sample.grib2')
>>> print grb
```

:1:1:Geopotential height [gpm]:50000 Pa (Isobaric surface):24 Hour Forecast initialized 2000070312:Polar stereographic:

:1:2:Temperature [K]:85000 Pa (Isobaric surface):24 Hour Forecast initialized 2000070312:Polar stereographic:

:1:3:u-component of wind [m s-1]:85000 Pa (Isobaric surface):24 Hour Forecast initialized 2000070312:Polar stereographic:

:1:4:v-component of wind [m s-1]:85000 Pa (Isobaric surface):24 Hour Forecast initialized 2000070312:Polar stereographic:

:1:5:Total precipitation [kg m-2]:(Ground or water surface):6 Hour Accumulation from 18 Hour Forecast initialized 2000070312:Polar stereographic

– PYGRIB2

```
Python 2.4.2 (#1, May  2 2006, 08:13:46)
[GCC 4.1.0 (SUSE Linux)] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>> import grib2
>>> grb = grib2.Grib2Decode('ght925.grib2')
>>> data=grb.getfld(0)
>>> data[0][0]
672.058105469
```

– ECWMF GRIB ENCODE - DECODE

http://www.ecmwf.int/products/data/software/grib2.html

grib_api-0.9.31.tar.gz

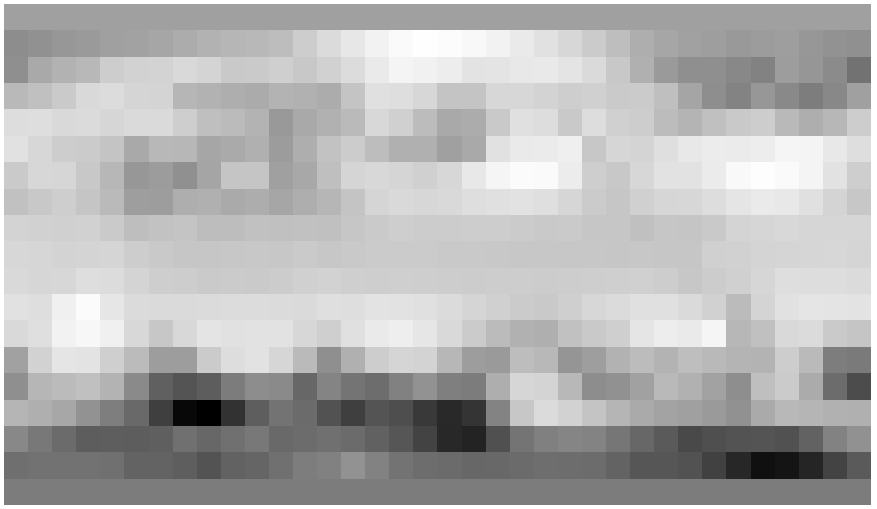
GRIB2 encoder/decoder

“C”

– GRIB_API

- grib_ls ght925.grib2 ==> list of parameters GRIB
- grib_dump -OtaH ght925.grib2 > dump.txt ==> contents of GRIB (octets)
- grib_copy -w lev=500 sample.grib2 out.grb ==> copy fields

grib2ppm ght925.grib2 > ght.ppm



– GRIB_API

```
#include <grib_api.h>

main(){
  grib_handle* g = NULL;
  int err = 0;
  FILE* f = fopen("ght925.grib2","r");
  g = grib_handle_new_from_file(0,f,&err);
  long date = 0;
  grib_get_long(g,"date",&date);
  printf("The date is %ld \n", date);
  fclose(f);
  grib_handle_delete(g);
}
```

wgrib2 -t ght925.grib2

grib_ls ght925.grib2

gcc -g -O2 -o grib2_date.x grib2_date.c -lgrib_api -lm

– NCEP GRIB ENCODE - DECODE

http://www.nco.ncep.noaa.gov/pmb/codes/GRIB2/

g2lib-1.0.8.tar

GRIB2 encoder/decoder

(Fortran90)

w3lib-1.3.tar

GRIB1 encoder/decoder

(Fortran90)

cnvgrib-1.1.3.tar

GRIB1 <--> GRIB2 dönüşümü. g2lib ve w3lib kütüphanelerine ihtiyaç duyar.

(Fortran90)

g2clib-1.0.4.tar

GRIB2 encoder/decoder ("C" versiyonu)

BUFR

(**B**inary **U**niversal **F**orm for the **R**epresentation of Meteorological data)

– BUFR

BUFR was created, circa 1989, with the goal of replacing the WMO's dozens of character-based, position-driven meteorological codes, such as SYNOP (surface observations), TEMP (upper air soundings) and CLIMAT (monthly climatological data).

- “Encoders/Decoders” can be used without making any change
- Data size makes it easy movement of large files
- Data can be processed efficiently
- Easy to store in “local database”

– BUFR

0100001001010101010001100101001000000000000000000001101000000001100000000000000000
0001001000000000000000000001110000000000000000000000000000000000000000100100000001
0000000100000100000111010000110000000000000000000000000000000000000111000000000

0000000000000001100000000000001000000010000000100000010000011000000010000000000
0000000000000000000010000000000010010000111101011101110001000000001101110011011 1
0011011100110111

Sections of BUFR

- 0 Indicator section
- 1 Identification section
- 2 Optional local use section
- 3 Data description section
- 4 Data section
- 5 End of message

- Octets 1-4 “BUFR”
- Octets 5-7 Total length of the mesg.
- Octet 8 Edition number

- Data can be compressed

– BUFR MESAJI NASIL GORUNUR?

'B' 'U' 'F' 'R' end of section 0 ➔ +

octet number	1	2	3	4	5	6	7	8	1	2
binary string	01000010010101010100011001010010000000000000000000011010000000011000000000000000									

octet number	3	4	5	6	7	8	9	10	11	12
binary string	000100100000000000000000001110000000000000000000000000000000000000000100100000001									

end of section 1 ➔ +

octet number	13	14	15	16	17	18	1	2	3	4
binary string	00000001000001000001110100001100000000000000000000000000000000000000111000000000									

end of section 3 ➔ +

octet number	5	6	7	8	9	10	11	12	13	14
binary string	00000000000000011000000000000010000000100000001000000100000110000000100000000000									

end of section 4 ➔ + '7' '7'

octet number	1	2	3	4	5	6	7	8	1	2
binary string	00000000000000000000100000000000100100001111010111011100010000000011011100110111									

'7' '7' + ➔ end of section 5

octet number	3	4
binary string	0011011100110111	

– BUFR / CREX
– (ENCODE - DECODE)

NCEP

http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/

ECMWF

http://www.ecmwf.int/products/data/software/bufr.html

BUFR

NCEP

http://www.nco.ncep.noaa.gov/sib/decoders/CREXLIB/

ECMWF

http://www.ecmwf.int/products/data/software/crex.html

CREX

BUFR ve CREX

Arasındaki Farklılıklar

BUFR

- BUFR “binary”
- BUFR “compressing”
- BUFR “SI unit system”

CREX

- CREX “ASCII”
- CREX “no support for compressing”
- CREX other units can be used

In general, BUFR and CREX are very similar to each other.
CREX is human readable

– CREX

- **CREX++**
- **T000101 A000 B11001**
  B11002++
- **260 0246+**
- **030 0137++**
- **7777**
-

- CREX edition1, WMO master table version 1
- OBS: Wind velocity and speed
- rüzgar 1: 260 degrees, 24.6 m/s
- rüzgar 2: 030 degrees, 13.7 m/s
-

THANK YOU!...