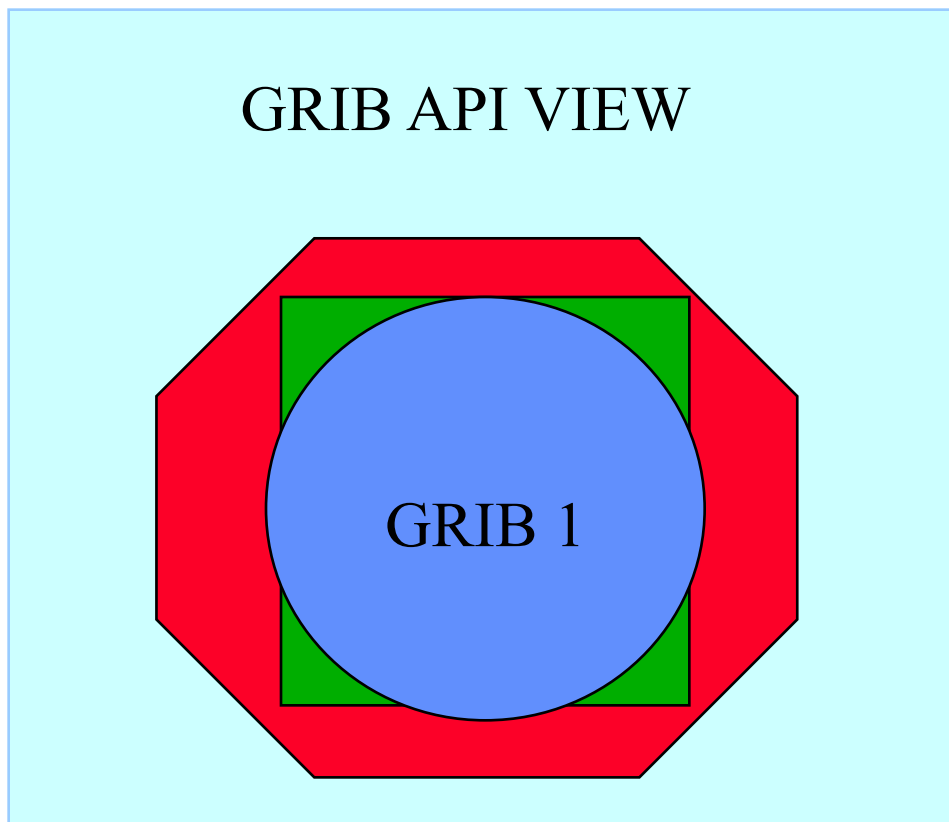


# **GRIB API: Keys**

**Shahram Najm  
Development Section  
Forecast Department**

# GRIB API keys



# GRIB API keys

- Each key has a **native type (real, integer, string)** and conversions are provided from one type to another when possible.
- The **set of keys available changes from one message to another** as it depends on the content of the message.
- Changing the value of some keys can cause some other keys to disappear and new keys to be available.
- The value of a key is **not always coded** in the GRIB message because it can be the result of the combination of several other keys through a given algorithm or just temporary (transient). Therefore we talk about
  - ❖ **CODED** keys ( coded in the message as they are )
  - ❖ **COMPUTED** keys ( temporary or computed from other keys )

# GRIB API keys: Namespace

- A **namespace** is a name for a set of keys.
- A key belonging to a namespace can be get/set by prefixing it with the namespace or simply without any prefix:

`time.step == step`

`parameter.paramId == paramId`

- Several namespaces are available e.g.

- ❖ parameter
- ❖ time
- ❖ geography
- ❖ vertical
- ❖ statistics

# GRIB API keys: **THE REFERENCE**

- **GRIB1**

<http://old.ecmwf.int/publications/manuals/d/gribapi/fm92/grib1/>

- **GRIB2**

<http://old.ecmwf.int/publications/manuals/d/gribapi/fm92/grib2/>

- **Edition independent**

<http://old.ecmwf.int/publications/manuals/d/gribapi/keys/>

# GRIB API keys

- **The easiest way to inspect a GRIB file is by using the tools**
  - `grib_ls` to get a summary of the content
  - `grib_dump` to get a more detailed view
  - `grib_filter` to get a custom output format
- **Most of what will follow will make more sense once you've had more hands-on experience! Bear with me 😊**

# GRIB API keys: file related

- **count**

- Message number in a file

- **countTotal**

- Message number in a set of files

- **offset**

- Position in bytes of the start of a message in a file

# GRIB API keys: data values

## ● values

- array of all the data values and missing values

## ● numberOfCodedValues

- number of values in the data sections (missing values excluded)

## ● numberOfPoints

- number of grid points = size of the values array

## ● numberOfMissing

- number of missing values

## ● max, min, average

- maximum, minimum and average of the field



# GRIB API keys: geography

- **distinctLatitudes, distinctLongitudes**

- array with the longitude/latitude distinct values

- **latitudes, longitudes**

- array with all the latitudes/longitudes for each point of the grid

- **latLonValues**

- array with all the latitudes/longitudes/values for each point of the grid
  - (lat1,lon1,value1,lat2,lon2,value2,...,latN,lonN,valueN)

# GRIB API keys: time

## ● Start of the forecast run

- **dataDate**        YYYYMMDD    (20070212)
- **dataTime**        (0000, 0600, 1200,...)

## ● Forecast step

- **stepType** (instant, accum, avg, max, min, diff, rms, sd, cov, ...)
- **stepUnits** (s, m, h, 3h, 6h, 12h, D, M, Y, 10Y, 30Y, C)
- **startStep**
- **endStep**
- **stepRange** ( “startStep-endStep” “endStep” )
- **step**

## ● Validity of the forecast

- **validityDate**
- **validityTime**

# GRIB API keys: grid ( gridType )

- For both editions:

- regular\_ll
- reduced\_ll
- mercator
- lambert
- polar\_stereographic
- UTM
- simple\_polyconic
- albers
- miller
- rotated\_ll
- stretched\_ll
- stretched\_rotated\_ll
- regular\_gg
- rotated\_gg
- stretched\_gg
- stretched\_rotated\_gg

- reduced\_gg
- sh
- rotated\_sh
- stretched\_sh
- stretched\_rotated\_sh
- space\_view

- For edition 2 only:

- triangular\_grid
- equatorial\_azimuthal\_equidistant
- azimuth\_range
- cross\_section
- Hovmoller
- time\_section

# GRIB API keys: geography

- Number of points:
  - **Ni (Nx)** along a parallel or x axis
  - **Nj (Ny)** along a meridian or y axis
- All latitude and longitude parameters are provided in a InDegrees version
  - longitudeOfFirstGridPoint -> longitudeOfFirstGridPointInDegrees
  - latitudeOfFirstGridPoint -> latitudeOfFirstGridPointInDegrees
  - longitudeOfLastGridPoint -> longitudeOfLastGridPointInDegrees
  - latitudeOfLastGridPoint -> latitudeOfLastGridPointInDegrees
  - iDirectionIncrement -> iDirectionIncrementInDegrees
  - jDirectionIncrement -> jDirectionIncrementInDegrees

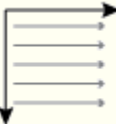
# GRIB API keys: geography


- Scanning mode should not be accessed directly. Use the following keys available for both editions.
  - **iScansNegatively**
  - **jScansPositively**
  - **jPointsAreConsecutive**
- “i” direction: west to east along a parallel, or left to right along an X-axis
- “j” direction: south to north along a meridian, or bottom to top along a Y-axis

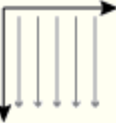
# GRIB API keys: geography

- To play with this flags and the grid definition see: <http://tigge.ecmwf.int/grid.html>

- Examples

Scanning mode:   i scans negatively  j scans positively  
 j points are consecutive  alternate row scanning

Scanning mode:   i scans negatively  j scans positively  
 j points are consecutive  alternate row scanning

Scanning mode:   i scans negatively  j scans positively  
 j points are consecutive  alternate row scanning

# GRIB API keys: geography

- For the following **gridTypes** a list of the latitudes and longitudes of the grid points can be obtained with **`grib_get_data (tools)`** and through a **`grib_iterator`** in **C/F90/Python**.
  - **`regular_ll`** (regular lat lon)
  - **`reduced_ll`** (reduced lat lon)
  - **`regular_gg`** (regular gaussian)
  - **`reduced_gg`** (reduced gaussian)
  - **`lambert`** (lambert conformal)
  - **`polar_stereographic`** (polar stereographic)
- The list of latitudes/longitudes can also be obtained if a **bitmap** is present.

# GRIB API keys: vertical

- typeOfLevel: surface, cloudBase, cloudTop, isothermZero, adiabaticCondensation, maxWind, tropopause, nominalTop, seaBottom, isothermal, isobaricInhPa, isobaricInPa, isobaricLayer, meanSea, heightAboveSea, heightAboveSeaLayer, heightAboveGround, ...
- level
- topLevel
- bottomLevel
- pv list of coefficients of the vertical coordinate



## GRIB API keys: levels (GRIB 2)

- **typeOfFirstFixedSurface (type and units)**
- **scaleFactorValueOfFirstFixedSurface**
- **scaledValueOfFirstFixedSurface**
- **typeOfSecondFixedSurface (type and units)**
- **scaleFactorValueOfSecondFixedSurface**
- **scaledValueOfSecondFixedSurface**

## GRIB API keys: parameter

- The definition of the parameter is very different in the two editions
- GRIB API provides some edition independent keys to identify a parameter :
  - **paramId**
  - **shortName**
  - **name**
  - **units**
  - **centre**

(Parameters will be covered in more depth later)

# GRIB API keys: **packingType**

- For GRIB edition 1:

- grid\_simple
- grid\_simple\_matrix
- grid\_second\_order
- spectral\_complex
- spectral\_simple

- For GRIB edition 2:

- grid\_simple
- grid\_simple\_matrix
- grid\_second\_order
- spectral\_simple
- spectral\_complex
- grid\_simple\_log\_preprocessing
- grid\_jpeg
- grid\_png
- grid\_ieee



## **It's almost lunchtime! Questions ?**