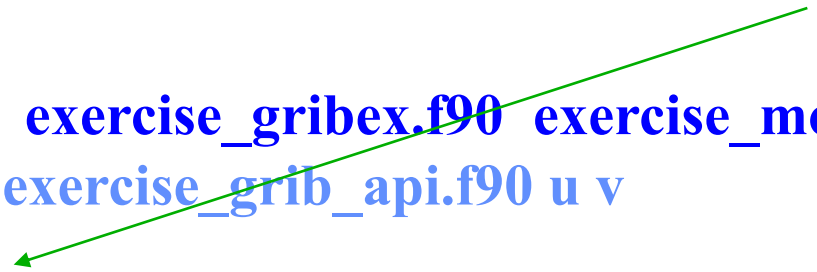
Practical 1: Intro to 'GRIB API' APIs

For the practicals, get the GRIB API practicals' archive:

```
> cd $SCRATCH
> tar xvf ~trx/grib_api/practicals.tar
> cd practicals/exercise1/
  # select F90 or C or Python
> cd F90
> ls
  Makefile  exercise_gribex.f90  exercise_mod.f90
  solution  exercise_grib_api.f90 u v
> make
> ./exercise_gribex
```

Compile/link exercise_gribex.f90 with EMOS lib



- The Fortran (or C) code exercise_gribex.f90(.c) uses GRIBEX to decode two GRIB files and to compute wind field and direction. The objective is to make all the necessary changes to use GRIB API.
- No Python interface for GRIBEX.

Practical 1: Main program

```
program exercise_gribex
  use exercise_mod
  implicit none

! Global variables
 real (kind = nbytes_dp), dimension(:), allocatable :: u,v,speed, direction

call read_fields('u',u)
call read_fields('v',v)
call compute_fields()
call clean_fields()
end program exercise_gribex
```

You have to modify the subroutine read_fields

```
> grib_ls –p parameter,shortName,dataDate,numberOfCodedValues,gridType,packingType u v
parameter  shortName  dataDate  numberOfCodedValues  gridType  packingType
  131          u       20080201      4131             regular_ll  grid_simple
...
  132          v       20080201      4131             regular_ll  grid_simple
...
2 of 2 total grib messages in 2 files
```

Practical 1: Read_fields (1/2)

Input arguments
Output arguments

```
subroutine read_fields(filename, values)
  use exercise_mod
  implicit none

 character(len=*), intent(in)                                    :: filename
 real (kind = nbytes_dp), dimension(:), allocatable, intent(out) :: values

external pbopen, pbclose, pbgrib, gribex

! Local variables
…
integer(kind=i4b), parameter :: junpack = 1280000
…
real(kind=nbytes_dp), dimension(junpack) :: zsec4
…
call pbopen(kunit, filename, open_mode, status)
…
call pbgrib(kunit, inbuff, junpack* nbytes_dp, &
            length, status)
…
call gribex(isec0, isec1, isec2, zsec2, isec3, zsec3, &
            isec4, zsec4, ipunp, inbuff, &
            ilenb, iword, yoper, istatus)
…
```

use grib_api

Use grib_get_size and dynamically allocate values. Isec, zsec* are not needed any more*

call grib_open_file

call grib_new_from_file

There is no equivalent call to gribex. Use grib_get to get only what you need!

ECMWF

Practical 1: Read_fields (2/2)

- How to proceed? You need to search for isec*, zsec* in your program and find the corresponding GRIB-API key

```
…
edition = isec0(2)
…
date = ((isec1(21)-1)*100 + isec1(10))*10000 + isec1(11)*100 + isec1(12)
…
```

GRIB edition number
To be replaced by a call to grib_get with key = 'edition'

isec1(21): *Century of reference time of data*

isec1(10): *Year of century*

isec1(11): *Month*

isec1(12): *Day*

- To find the right GRIB API key, use tables defined at

https://software.ecmwf.int/wiki/display/GRIB/GRIBEX+keys

- Try to use "Recommended keys" i.e. keys valid both for GRIB-1 and GRIB-2

Practical 1: The objectives

- Two levels of difficulties:
  - If you are confident and familiar with GRIBEX, you can start from exercise_gribex.f90(.c)
  - Or start from exercise_grib_api.f90(.c) where you will only have to include the grib_api I/O statements and make the appropriate calls to grib_get to replace GRIBEX.
- Compile/link with:

  Fortran: gfortran –o exercise_grib_api exercise_grib_api.f90 \
  exercise_mod.f90 $GRIB_API_INCLUDE $GRIB_API_LIB

  C: gcc -o exercise_grib_api exercise_grib_api.c \
  $GRIB_API_INCLUDE $GRIB_API_LIB -lm

  or use make.
- For Python, run with:

  python exercise_grib_api.py

Practical 1: The objectives

- Run the resulting code

  ./exercise_grib_api # for fortran or C

  python exercise_grib_api.py # for python

- Compare with the output produced by exercise_gribex
- Now change the links for the input files to u.grib2 and v.grib2 (GRIB-2) (from ~trx/grib_api/data) and run the two executables again.
  - rm u v
  - ln –s ~trx/grib_api/data/u.grib2 u
  - ln –s ~trx/grib_api/data/v.grib2 v

TIPS

- Use ecgate, not the local desktop!
- For C and Fortran, use make (‘make [-f <Makefile>] [clean]’)
- Documentation for GRIB API can be found at

  https://software.ecmwf.int/wiki/display/GRIB/GRIB+API
- Error codes are listed under:

  https://software.ecmwf.int/wiki/display/GRIB/Error+codes
- See lecture notes, or ask us …