ERA-Interim: How to calculate daily total precipitation

Last modified on Nov 08, 2023 11:34

> **ERA-Interim production to stop on 31st August 2019**
>
> For ERA-Interim (1st January 1979 to 31st August 2019) access through the ECMWF Web API **stopped on 01 June 2023**
>
> Its successor ERA5 is available from the Climate Data Store (CDS) (What are the changes from ERA-Interim to ERA5?) and users are strongly advised to migrate to ERA5 (How to download ERA5).
>
> For those users who still need access to ERA-Interim after 01 June 2023 (subject to further notice), they can do so via the Climate Data Store (CDS) API.

This knowledge base article shows you how to calculate daily total precipitation using ERA-Interim data.

Before you continue, make sure you read through knowledge base articles listed below:

- How to download ERA-Interim data from the ECMWF data archive
- ERA-Interim: 'time' and 'steps', and instantaneous, accumulated and min/max parameters

You are also supposed to know how to work with Python under Linux, in particular, how to install packages using pip. You are recommended to use the latest release of packages listed here:

- Climate Data Store (CDS) API - required for step 1
- netCDF4 (tested with 1.4.0) - required for step 2
- numpy (tested with 1.14.5) - required for step 2

Step-by-step guide

1. Use script below to download daily total precipitation ERA-Interim data for 1st January 2018. This script will download total precipitation from two forecasts at 00 UTC and 12 UTC, both at step 12. This will give you 24 hours coverage.
    a. Time 00 UTC and step 12 will give you 00 - 12 UTC total precipitation data
    b. Time 12 UTC and step 12 will give you 12 - 24 UTC total precipitation data

```
#!/usr/bin/env python
   import cdsapi
   c = cdsapi.Client()
   c.retrieve('reanalysis-era-interim', { # Requests follow MARS syntax
                                         # Keywords 'expver' and 'class' can be dropped. They are
obsolete
                                         # since their values are imposed by 'reanalysis-era-interim'
       'date'    : '2018-01-01',         # The hyphens can be omitted
       'levtype' : 'sfc',                 # Pressure levels, 'ml' for model levels, 'sfc' for surface
       'param'   : '228',                # Full information at https://apps.ecmwf.int/codes/grib/param-
db/
                                         # The native representation for temperature is spherical
harmonics
       'stream'  : 'oper',               # Denotes atmospheric fields. Wave fields use 'wave'.
       'time'    : '00/12',
       'type'    : 'fc',
           'step'    : '00/to/12'
       'area'    : '80/-50/-25/0',       # North, West, South, East. Default: global
       'grid'    : '1.0/1.0',            # Latitude/longitude. Default: spherical harmonics or reduced
Gaussian grid
       'format'  : 'netcdf',             # Output needs to be regular lat-lon, so only works in
combination with 'grid'!
   }, 'ERAI-pl-temperature-subarea.nc')   # Output file. Adapt as you wish.
```

2. Run a second script to calculate daily total precipitation. All it does is to add up the two values for 00 - 12 UTC and 12 - 24 UTC for a given day.

```
#!/usr/bin/env python
"""
Save as file calculate-daily-tp.py and run "python calculate-daily-tp.py".

Input file : tp_20180101.nc
Output file: daily-tp_20180101.nc
"""
import time, sys
```

```
from datetime import datetime, timedelta

from netCDF4 import Dataset, date2num, num2date
import numpy as np

day = 20180101
f_in = 'tp_%d.nc' % day
f_out = 'daily-tp_%d.nc' % day

d = datetime.strptime(str(day), '%Y%m%d')
time_needed = [d + timedelta(hours = 12), d + timedelta(days = 1)]

with Dataset(f_in) as ds_src:
    var_time = ds_src.variables['time']
    time_avail = num2date(var_time[:], var_time.units,
            calendar = var_time.calendar)

    indices = []
    for tm in time_needed:
        a = np.where(time_avail == tm)[0]
        if len(a) == 0:
            sys.stderr.write('Error: precipitation data is missing/incomplete - %s!\n'
                    % tm.strftime('%Y%m%d %H:%M:%S'))
            sys.exit(200)
        else:
            print('Found %s' % tm.strftime('%Y%m%d %H:%M:%S'))
            indices.append(a[0])

    var_tp = ds_src.variables['tp']
    tp_values_set = False
    for idx in indices:
        if not tp_values_set:
            data = var_tp[idx, :, :]
            tp_values_set = True
        else:
            data += var_tp[idx, :, :]

    with Dataset(f_out, mode = 'w', format = 'NETCDF3_64BIT_OFFSET') as ds_dest:
        # Dimensions
        for name in ['latitude', 'longitude']:
            dim_src = ds_src.dimensions[name]
            ds_dest.createDimension(name, dim_src.size)
            var_src = ds_src.variables[name]
            var_dest = ds_dest.createVariable(name, var_src.datatype, (name,))
            var_dest[:] = var_src[:]
            var_dest.setncattr('units', var_src.units)
            var_dest.setncattr('long_name', var_src.long_name)

        ds_dest.createDimension('time', None)

        # Variables
        var = ds_dest.createVariable('time', np.int32, ('time',))
        time_units = 'hours since 1900-01-01 00:00:00'
        time_cal = 'gregorian'
        var[:] = date2num([d], units = time_units, calendar = time_cal)
        var.setncattr('units', time_units)
        var.setncattr('long_name', 'time')
        var.setncattr('calendar', time_cal)
        var = ds_dest.createVariable(var_tp.name, np.double, var_tp.dimensions)
        var[0, :, :] = data
        var.setncattr('units', var_tp.units)
        var.setncattr('long_name', var_tp.long_name)

        # Attributes
        ds_dest.setncattr('Conventions', 'CF-1.6')
        ds_dest.setncattr('history', '%s %s'
                % (datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
                ' '.join(time.tzname)))

        print('Done! Daily total precipitation saved in %s' % f_out)
```

For simplicity, data in the output NetCDF file of the second script is unpacked. You may want to pack the data to save some disk spaces. Refer to https://www.unidata.ucar.edu/software/netcdf/workshops/most-recent/bestpractices/Packing.html for detailed information.

Related articles

- Transformation or regridding of ECMWF Reanalyses data
- Model grid box and time step
- ERA-Interim: documentation
- ERA-Interim: How to calculate daily total precipitation
- ERA-Interim known issues