

# ERA5 Hourly Data Downloader and Extractor

## Overview:

This script is designed to work with ERA5 reanalysis data from ECMWF using both the CDS API and the MARS (Meteorological Archive and Retrieval System). MARS is ECMWF's archive retrieval system that enables users to request data using a strictly defined syntax. Detailed information on MARS request syntax and best practices can be found in the official [MARS User Documentation](#).

The script supports two operational modes:

### 1. Download & Process:

- Downloads ERA5 data in monthly chunks from the CDS API (using MARS syntax rules).
- Processes the resulting GRIB files to extract a pre-defined set of meteorological and oceanographic variables using Inverse Distance Weighting (IDW) for interpolation.
- Saves the combined data into a CSV file, sorted by datetime.

### 2. Extract Only:

- Skips the download phase and directly processes all available GRIB files locally.
- Uses parallel processing with progress monitoring to efficiently extract data.
- In addition to matching GRIB messages by their short names, it also extracts data using param IDs when available.

## Detailed Functionality:

### 1. CDS API and MARS Requests:

- The request dictionary is built following the strict syntax required by the MARS system. For example, keys such as `product_type`, `format`, `param`, `year`, `month`, `day`, `time`, `area`, and `grid` must be provided in the correct format.
- This script builds the request using only official ERA5 param IDs (as strings) to avoid ambiguity.
- The `area` key is specified as `[North, West, South, East]` and `time` values are provided in "HH:00:00" format.
- For further details, refer to the [MARS User Documentation](#).

### 2. Unified Variable Mapping:

- A unified dictionary called `VARIABLES` contains both the official ERA5 param ID and the expected GRIB short name for each variable.

- From this mapping, a list of param IDs (`PARAM_IDS`) is derived for the CDS API request and a GRIB message key mapping (`GRIB_KEY_MAP`) is created to map the GRIB message short names to internal keys.

### 3. GRIB File Processing:

- GRIB files are processed using the `pygrib` library.
- For each GRIB message, the script first attempts to match the short name to our internal keys.
- If the short name is not found, it falls back to comparing the GRIB message's parameter number (if available) to the expected param IDs.
- The script uses Inverse Distance Weighting (IDW) interpolation to estimate the value at the target coordinate.
- Extracted data from all GRIB files are combined into a pandas DataFrame, sorted by datetime, and exported as a CSV file.

### 4. Robust Error Handling and Logging:

- Detailed logging records each major step and any encountered issues.
- The download process is retried multiple times with increasing delay intervals if failures occur.
- After processing, the script checks for missing monthly GRIB files and issues warnings accordingly.

## Usage:

When executed, the user is prompted to choose between:

- **Option 1:** Download ERA5 data via the CDS API (using MARS syntax) and process the downloaded GRIB files.
- **Option 2:** Only process existing GRIB files in the data directory.

Dependencies include Python 3.x and libraries such as `cdsapi`, `pygrib`, `pandas`, `numpy`, `tqdm`, and `logging`.

**Note:** ECCODES is required for the proper functioning of `pygrib`.

## ECMWF Data Information:

- **Website:** [ECMWF](#)
- **ERA5 reanalysis dataset:** [ERA5 Dataset](#)
- **Parameter reference:** [Parameter Database](#)

For more detailed information on CDS API and MARS request syntax, please refer to the [MARS User Documentation](#).