Ravazzani, G.[1]*

(1) Politecnico di Milano, Italy

(*) corresponding author: giovanni.ravazzani@polimi.it

ABSTRACT: In recent years, many distributed hydrologic models were proposed, varying in nature and complexity including advanced methodologies and computational methods that improved significantly models' performances. On the other hand, proliferation of distributed hydrological models gave birth to a myriad of different file formats used for storing results. Most of the models produce files that are digitally encoded in the ASCII format. There are also models that produce files in binary format, making the data less accessible. This has been a significant obstacle to data sharing and comparison across user communities. Moreover, the complexity of natural phenomena involved in hydrological processes often requires model coupling rising concerns on models interoperability.

Unlike other scientific communities, meteorologists being an example, hydrological researchers have never agreed a standard format for storing and sharing data and model results. Recently some initiatives addressing data interoperability for the simplification of the linking of hydrologic models claimed the need of standard format to be used in hydrological applications.

An emerging standard for data exchange in the scientific research community is the NetCDF (network Common Data Form), a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. This data format is now used by numerous earth science studies institutions and is supported by several commercial analysis and data manipulation and visualization packages.

This work presents MOSAICO, a set of Fortran 90 Modules for facilitating development of raster based hydrological applications and stimulating adoption of netCDF as a common format for sharing and comparing data among hydrological community. MOSAICO include routines for high level operations for input output and manipulation of gridded dataset. A test program is presented in order to show basic MOSAICO capabilities.

DESIGN PHILOSOPHY

MOSAICO is developed on top of netCDF. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data and metadata.

NetCDF data is:
- *Self-Describing*. A netCDF file includes metadata information about the data it contains.
- *Portable*. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- *Scalable*. A small subset of a large dataset may be accessed efficiently.
- *Appendable*. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- *Sharable*. One writer and multiple readers may simultaneously access the same netCDF file.

NetCDF has an ASCII analog format, CDL, that can be easily encoded to binary version. The netCDF data format structure includes dimensions, variables, and attributes. Adoption of metadata conventions extends interoperability by specifying special keywords with agreed meaning to variables and attributes. This enables users of data from different sources to decide which quantities are comparable, and preserves integrity of units of measurement. MOSAICO supports Climate and Forecast (CF) Metadata Convention. The CF convention is increasingly gaining acceptance and has been adopted by a number of projects and groups as a primary standard. The CF convention defines metadata that provide a definitive description of what the data in each variable represents, and the spatial and temporal properties of the data.

MOSAICO was developed in the Fortran 90 programming language. Fortran 90 was selected as the development language due to its high portability and efficiency. In fact, Fortran 90 compilers exist for almost every operating system and, due to long experience accumulated during years, they have been optimized to provide faster performance. These motivated the use of Fortran 90 for the development of many distributed hydrological models such as TOPKAPI (TOPographic Kinematic APproximation and Integration), DimoSOP (DIstributed hydrological Model for the SOP), AFFDEF and FEST (Flash – flood Event – based Spatially – distributed rainfall – runoff Transformation).

MOSAICO conforms to European standards for writing and documenting exchangeable Fortran 90 code (Andrews et al, 2011) and f90doc, a documentation tool for Fortran 90.

MOSAICO is composed of different modules that can be easily reused and integrated, even singularly, in larger projects. MOSAICO was tested on Microsoft Windows and GNU/Linux operating systems using Intel and GFortran compilers, but can virtually be used on any system where Fortran 90 compiler and netCDF library can be installed.

MOSAICO is distributed under the GNU/GPL v3 license. Source code of the library and a test program to show basic MOSAICO capabilities can be downloaded from https://github.com/gravazza/MOSAICO.

COMPONENTS OF MOSAICO

MOSAICO is composed of 5 main Fortran 90 Modules plus other 8 Modules for message logging, string manipulation, physical constants definition, and general purpose utilities.

- **GridLib**. This is the fundamental module of MOSAICO. It introduces two new types, **grid_real** and **grid_integer**, that define, respectively, a floating point and an integer gridded dataset (2D-grid) and its properties. Basic routines are: **NewGrid** for reading grid from file or creating a new one using an existing grid as a template; **ExportGrid** to write grid on a new file or appending to existing one; **GridDestroy** for deallocating grid from memory. In addition to netCDF file format, GridLib supports ESRI ASCII and BINARY format, for compatibility with existing hydrological models that use ESRI format easing migration to netCDF.

- **GridOperations**. This module contains routines for manipulating and analysing grid dataset. Basic routines are: **GridConvert** for transformation of coordinate reference system; **CrsIsEqual** to check if two grids have the same coordinate reference system; **GetIJ** and **GetXY** to return X and Y coordinate given position (I, J) in the local reference system, and vice versa; **GridResample** to change spatial resolution of a grid, optionally matching grid to a given mask; **CellArea** to compute area, in m^2, of a single cell of a grid (differs with latitude when coordinate reference system is not plane projected); **GetMean** and **GetSum** to compute, respectively, the mean and the sum of a grid.

- **GeoLib**. Structure of this module is inspired by GEOTRANS, a product of the National Geospatial-Intelligence Agency (NGA) and U.S. Army Engineering Research and Development Center. The module Geolib contains routines that allows the developer to convert geographic coordinates among a variety of coordinate systems, map projections, and datums. The function **Convert** is the heart of the GeoLib module. It is this function that actually performs all coordinate conversions and datum shifts..

Supported reference systems include Geodetic, Transverse Mercator, Universal Transverse Mercator, Gauss Boaga (Italy), Hotine Oblique Mercator, and Swiss Oblique Cylindrical.

- **Morphology**. Contains routines dealing with river and river basin morphology. Basic routines are: **HortonOrders**, to assign Horton–Strahler number to each branch of the river network; **FlowAccumulation** to create a raster of accumulated flow to each cell as
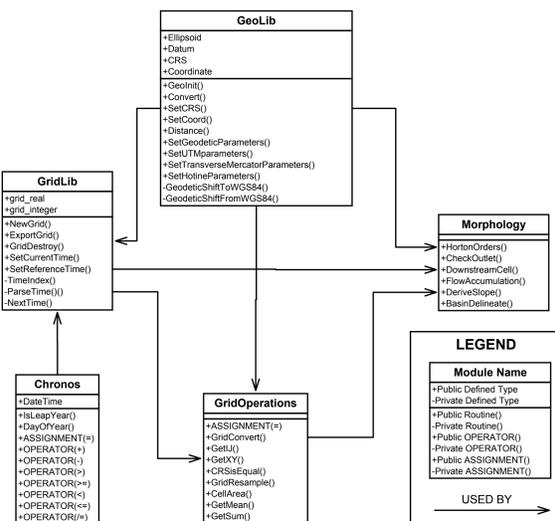
determined by accumulating the area (m^2) for all cells that flow into each downslope cell; **DeriveSlope** to compute local slope; **BasinDelineation** to extract river basin boundary given the coordinate of the outlet.

- **Chronos**. Introduces a new type, **DateTime**, and contains routines to manage date and time. Most of routines are callable by the overloaded operators and assignments

AN APPLICATION EXAMPLE

The netCDF file used for demonstration is the publicly available high-resolution gridded precipitation for the Greater Alpine Region of Europe (GAR; 4°E-19°E, 43°N-49°N) developed by the Climatic Research Unit as part of the ALP-IMP project (Efthymiadis et al., 2006). The precipitation dataset provides monthly precipitation totals, for the 1800-2003 period, in geodetic reference system gridded at 10-minute resolution. The dataset is based on 192 long-term homogenized precipitation series from meteorological stations across the study domain and a high-resolution precipitation climatology for the 1971-1990 period.
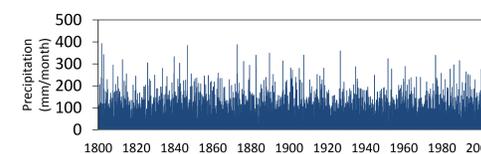
The goal of this test program is to calculate the mean monthly precipitation time series over the river Po basin (North Italy) whose extent is given in terms of a ESRI ASCII grid in Universal Transverse Mercator (UTM) zone 32 North (32N) European Datum 1950 (ED50) reference system. Alternatively the river Po basin mask could have been computed by calling BasinDelineation subroutine passing raster of flow direction and coordinate of the outlet.

```
!---------------------------------------------
!   read time variant precipitation field in netCDF dataset
!---------------------------------------------
!initialize time counter
timeString = '1800-01-01T00:00:00+00:00'
time = timeString
dt = month
!open file for output
fileUnit = GetUnit ()
OPEN (UNIT = fileUnit,file = 'output.txt')
!initialize Po precipitation map using mask as template
CALL NewGrid (layer = precipitation_po, grid = mask)
!Assign coordinate reference system to temporary grid
temporary_grid % grid_mapping = mask % grid_mapping
!loop
DO i = 1, 2460

    !read precipitation from netcdf
    CALL NewGrid (layer = precipitation, fileName = 'alp_pre_dat.nc' , &
    fileFormat = NET_CDF, variable = 'precipitation', time = time)

    !coordinate conversion
    CALL GridConvert (GridIn = precipitation, GridOut = temporary_grid, &
    cellsize = precipitation_po % cellsize)
    CALL GridResample (grid = temporary_grid, resampledGrid = precipitation_po)

    !compute mean precipitation over Po river basin
    mean = GetMean (precipitation_po)

    !write result
    WRITE (*,*) i, mean
    WRITE (fileUnit,*) i, mean
    time = time + dt
END DO
```



```
!--------------------------------------
!  initialize cartographic engine
!--------------------------------------

CALL GeoInit ('GeoLib.ini')

!--------------------------------------
!  read mask and set coordinate reference system
!--------------------------------------

!read grid
CALL NewGrid (layer = mask, fileName = 'po_basin.asc', fileFormat = ESRI_ASCII)

!set CRS
CALL SetCRS (CRStype = UTM, datumType = ED50, rs = mask % grid_mapping)

!set UTM parameters
CALL SetUTMparameters (system = mask % grid_mapping, zone = 32, &
hemisphere = NORTH, override = 1)
```