Documentation for release of MarkSim base data from WorldClim 2.0

Peter G. Jones. July 2018 to February 2019
Waen Associates Ltd, Dolgellau, Wales

CONTENTS.

Introduction

MetGrid files
- Reason
- Structure
- Data representation
- Indexing
- Rotation

MetGrid_Handler module
- Accessing MetGrid files from Fortran
- Module description
  - ***Variable structures***
  - ***Public variables***
  - ***Function descriptions***
    - Grid functions
    - Calendar functions
    - Input functions
    - Output functions
- Examples

CLI files
- Reason
- Structure
- User access

List of figures

List of tables

References

Introduction

From the first records in the 18th century in Europe daily weather has been observed and recorded; it was crucial for the agricultural revolution in England. Researchers have long used these data for agronomic and grazing studies, some of these go back to not long after the records began. The ethos of data collection for the agricultural world spread to European colonies around the world. These were used to plan and manage plantations. Tea in Kenya and Assam, cacao in western Africa, rubber in the Malay peninsula and cotton in America.

All of these analyses were done on a point basis; if meteorological data were need then they had to be recorded at the point of use. Early methods of interpolation relied on climate classifications, which could be mapped; Köppen (1884) is an example. Digital versions of climate data, for example Hijmans *et. al* (2005) are useful for GIS application where environmental indices can be calculated from the monthly climate data.

Daily data are generally needed For crop modelling, but these are far too voluminous to store in a high-resolution global coverage. This problem is compounded by the need to include realistic variation, both short term and annual, in modelling analyses. One solution to this is to use a weather generator. MarkSim is a daily rainfall generator devised for the tropics in CIAT and ILRI in the late 1990's building on work at CIAT from the previous 15 years; Jones & Thornton, (1993, 1997, 1999, 2000), Jones *et. al*. (2002). More recently, see Jones & Thornton (2013), we have added a capability to simulate future weather sequences from a suite of GCM models, MarkSimGCM a stand-alone version is available in an EXE as MarkSim_Standalone.

The weather generator needs all of the monthly climate values in one record and so the MetGrid file was born. This eventually used the data layers of WorldClim. This work is to update the MarkSim application data with the new data from WorldClim v2.0, Fick & Hijmans (2017). The outputs are a new set of MetGrid files at 10, 5 and 2.5 arc minutes and 30 arc seconds, and sets of CLI files, which can be used as input to the stand-alone version of MarkSim or as input to other weather generators. Until now, MetGrid files have not been available to the general user; this changes with this version and they will be available for download from CCAFS at CIAT. As they are a highly specialised format, a Fortran module, MetGrid_Handler, has been produced for general distribution to assist their use.

MetGrid

Reason

The basic MetGrid structure was created in the late ‘80s for various applications in CIAT that used long-term climate data from the CIAT Climate Database. In those days GIS was rudimentary and using single variable coverages for any calculation more complicated than simple overlay was out of the question. The CIAT Climate Database (later to be incorporated into WorldClim) held individual station data. These were interpolated into 10 minute and 5 minute grids, but instead of storing the grids by monthly variates, I devised the MetGrid file so that calculations such as water balance, length of season, climate and land classification and varietal suitability could all be calculated directly, pixel by pixel, using Fortran programs.

Structure

For a start, a MetGrid file is not a grid file. It is a space-conserving file can be used to construct derived climate grids throughout the world. The base data are from WorldClim v2.0 with the exception of the number of rain days that are derived from MarkSim v1.0. MetGrid files come in four resolutions; 10 arc-minutes, 5 arc-minutes, 2.5 arc-minutes and 30 arc-seconds. The big difference from the WorldClim files is that all variates are represented in one record.

The MetGrid files are not raster grids, or, in fact, any sort of grid, and so they have to carry additional data to identify the record. This is simply the Latitude, longitude, and median elevation of the pixel. MetGrid v1.0 used the latitude and longitude of the lower left corner of the pixel, but we found that rounding errors made it difficult to translate easily to normal GIS grids. In version 2 we’ve decided on the latitude and longitude of the centre of the pixel. The user will now find no problem with translating any MetGrid to a raster image using the functions in the MetGrid_handler module.

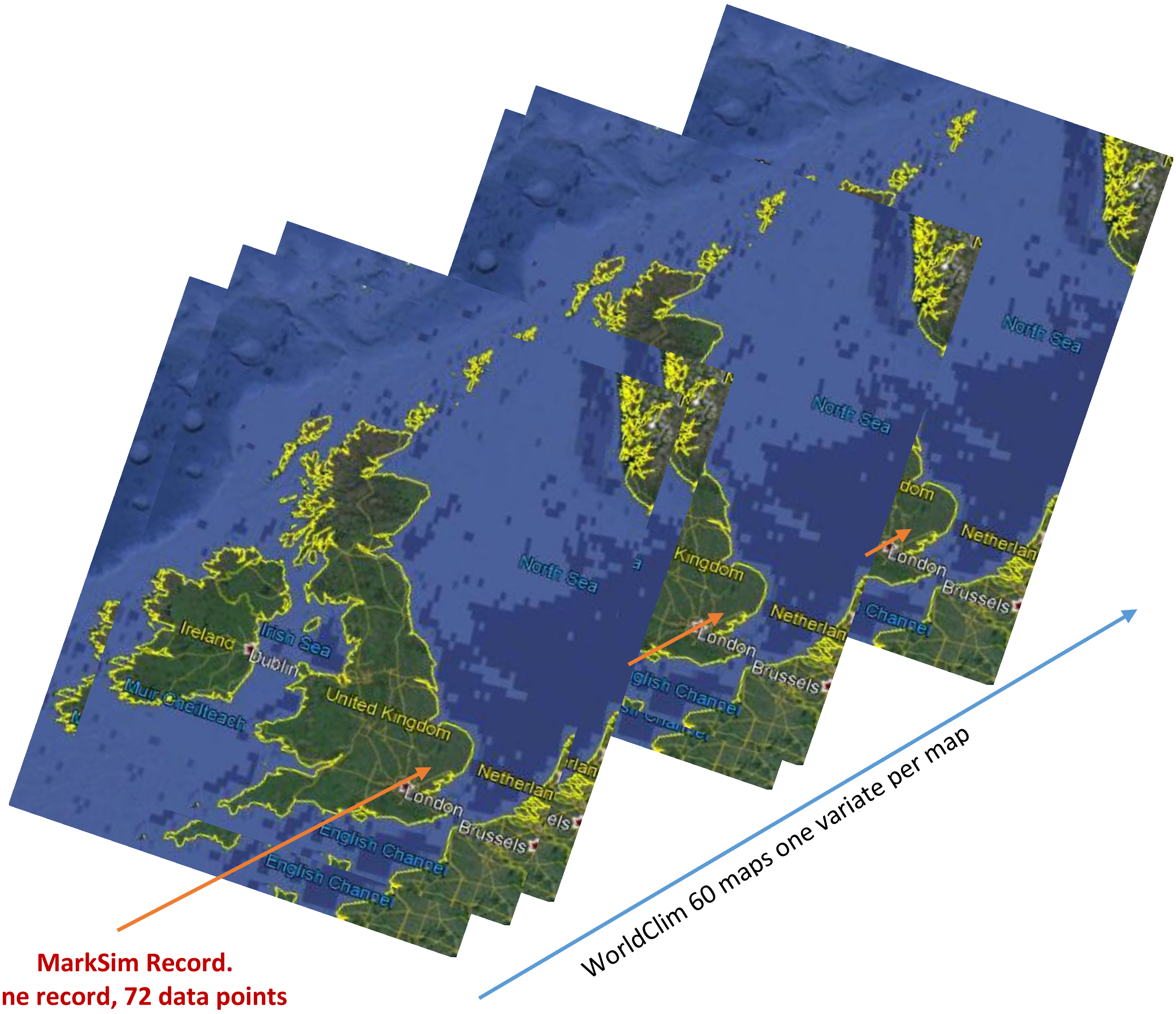


Fig 1, Illustration of the different data structures of MetGrid and WorldClim

Data representation

MetGrid records are stored as binary record in order from North to South in lines of longitude from West to East, but non-land areas are not recorded. There is therefore no simple relationship between a record number in the MetGrid file and a raster map. In order to overcome this the full MetGrid fileset is a group of three files with dedicated file extensions. They should always be copied together as a unit or the full functioning of the MetGrid will be impaired.

They consist of a file header, extension .hdr, which describes the fileset. It is a small ASCII file such as this one for the 10 arc minute dataset. Note that although the contents appear to describe a raster grid dataset, the MetGrid is not; what is being described here is the grid that it represents.

```
Name            World2_10m.mtg
Author          P.G.Jones, Jul '18
Cols                 2160
Rows                  870
Pixels             583580
Max_lat              85.0
Min_lat             -60.0
```

```
Max_long              180.0
Min_long             -180.0
Resolution          0.166666667
Index               yes
Index_name          World2_10m.idx
Index_rec_size          57
```

The next file is the MetGrid data file, extension .mtg. This is a binary file of compressed climate data records. It is unreadable without the functions in MetGrid_handler, or ones constructed on similar lines. The record definition is:

```
type metgrid_record                              ! Total record length = 104 bytes
       real*4           :   lat, long, phase
       integer*2        :   elev
       integer*2        :   r (9)          ! mm
       integer*2        :   t (9)          ! degree C times 30 + 2100 offset
       integer*2        :   d (9)          ! degree C times 100
       integer*2        :   s (9)          ! MJ times 110
       integer*2        :   rd (9)         ! raindays times 120
end type metgrid_record
```

The nine by 2-byte data fields are the twelve monthly values compressed into 12-bit values and scaled as indicated. They are stored as shown below. They do not conform to full word boundaries.

```
0123456701234567012345670123456701234567012345670123456701234567     - bit numbering
1       2       3       4       5       6       7                    - bytes
1               2               3               4                    - half word boundaries
aaaaaaaaaaaabbbbbbbbbbbbccccccccccccdddddddddddd etc.  12 bit words a, b, c, d as
packed.
```

The use of 12-bit words in modern computing is unusual, but the type and quality of the data in this case lend themselves to it, considerably reducing record size and hence file size. The extra computation involved can be offset against the reduced seek and read times from the file which will only have to be produced once. The overall record *does* conform to standard 32-bit word boundaries, and so it is read efficiently.

The variates are:

Lat, long	latitude and longitude
Phase	The angle through which the data have been rotated (see the section on data rotation below)
Elev	The modal elevation for the pixel taken from GTOPO30
R	rainfall per day mm
T	mean temperature per month °C
D	mean diurnal temperature range per month °C
S	mean solar radiation per day Mjm^{-2}
Rd	rain days per day. Calculated from the internal functions of MarkSim v1.0

Indexing

The index file (extension .idx) is a binary file that can efficiently relate a latitude and longitude to a single record in the direct access .mtg file without involving multiple reads and searches. It is essentially a run length encoded file by lines of latitude. The run length encoded values are simply presence or absence of a land based pixel and the identity of the target record within the .mtg file can be returned with the absolute minimum of computation. Each .idx is specific to the resolution of the .mtg file and so must never be separated from it.

Rotation

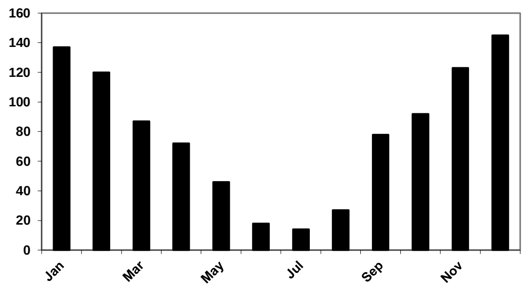
The use of MarkSim and the MetGrid files requires an understanding of climate rotation. The following section was taken mainly from the original manuals of FloraMap, MarkSim and Homologue, where it is crucial in the operation of these applications.

The climatic events that occur through the year, such as summer/winter and start/finish of the rainy season, are of prime importance when comparing one climate with another. Unfortunately, they occur at different dates in many climate types. The most obvious case is where we compare climates between points in the Northern and Southern Hemispheres, but more subtle differences occur in climate event timing throughout the tropics. What we need is a method of eliminating these differences to allow us to make comparisons free of these annual timing effects.

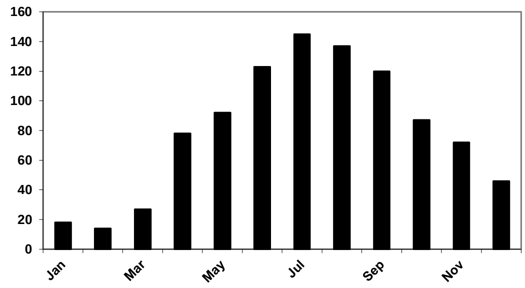
Let us look at two hypothetical climate stations. They are in a typical Mediterranean climate—warm wet winters, hot dry summers. Northville could be somewhere in California, and Southville might be in Chile. The August rainfall in Southville happens in January in Northville (Figure 3.3). If we plot these rainfalls in polar coordinates, we can readily see that to compare them we need to rotate them to a standard time.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Northville	137	120	87	72	46	18	14	27	78	92	123	145
Southville	18	14	27	78	92	123	145	137	120	87	72	46

Table 1 Typical rainfalls for imagined Mediterranean climates



Northville monthly



Southville monthly

Fig 2 Monthly rainfalls for Northville and Southville.

How do we do this automatically? The answer is the 12-point Fourier transform. This is fortunately the simplest of all the possible Fourier transform algorithms. It is highly computationally efficient and fast. It takes the 12 monthly values and converts them to a series of sine and cosine functions. The one used in MarkSim has a modification to make it conserve the monthly total values (Jones, 1987). The equation produced is:

$$r = a_0 + \sum_{i=1}^{6} a_i \sin(ix) + b_i(ix) \qquad (5)$$

We can write this as a series of frequency vectors, each with an amplitude α_I, and a phase angle, θ_i:

$$\alpha_i = \sqrt{(a_i^2 + b_i^2)} \qquad \theta_i = \sin\left(\frac{b_i}{\alpha_i}\right) = \cos\left(\frac{a_i}{\alpha_i}\right) \qquad (6)$$

If we subtract the first phase angle from all the other vectors in the set, we have produced a rigid rotation of the vectors; this is the rotation we want, it puts the maximum of the first frequency at a phase angle of zero and places the rest in positions equivalent to their angular separation in the original data. We then use the first phase angle for rainfall to rotate the data for temperature and diurnal temperature range, which we rotate through the same angle.

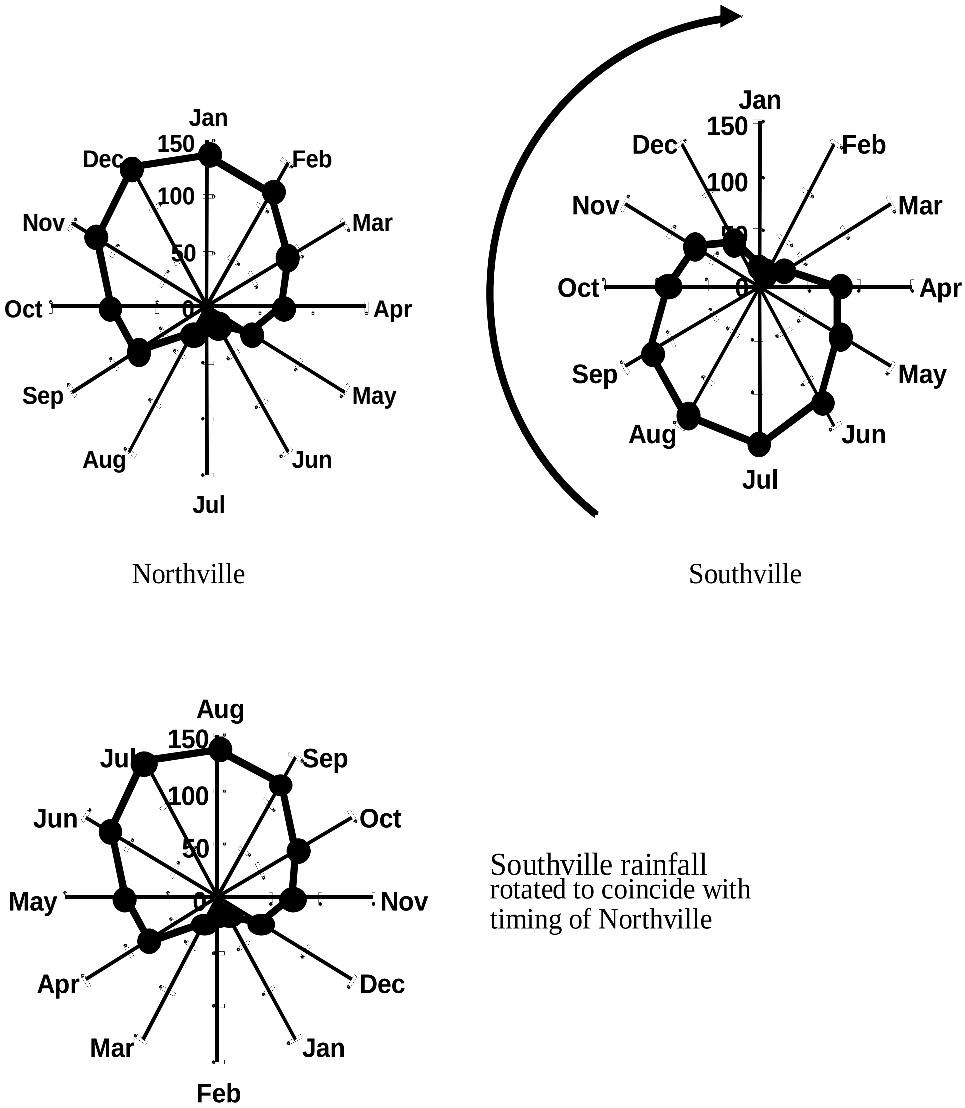


Fig 3 Illustration of the rigid rotation of a MetGrid record

This explanation works well for the tropics. There was a small chance of the procedure going off the rails if the rainfall record did not have a seasonal peak. This was the case in some records from tropical desert regions, in these cases the rotation was ambiguous and sometimes resulted in pixels allocated to the wrong cluster.

The beta release of MarkSim went out with this type of rotation algorithm, as did the first release of FloraMap. When the climate grids of the latter were extended to Europe, the case arose where annual climate pattern was dominated by temperature and not rainfall.

We therefore have the possibility of rotating on rainfall or temperature, but when to decide which is the dominant? We tried many combinations of rules, but unfortunately came to the conclusion that none were acceptable. They all resulted in a hard line across the map at some point where the rotation basis changed. This led to climates that should have been grading imperceptibly from one type to another suddenly jumping at a discontinuity. This would have given the users serious problems when fitting models in these areas.

The best solution found is to use BOTH the rainfall and the temperature in calculating the rotation phase angle. Thus:

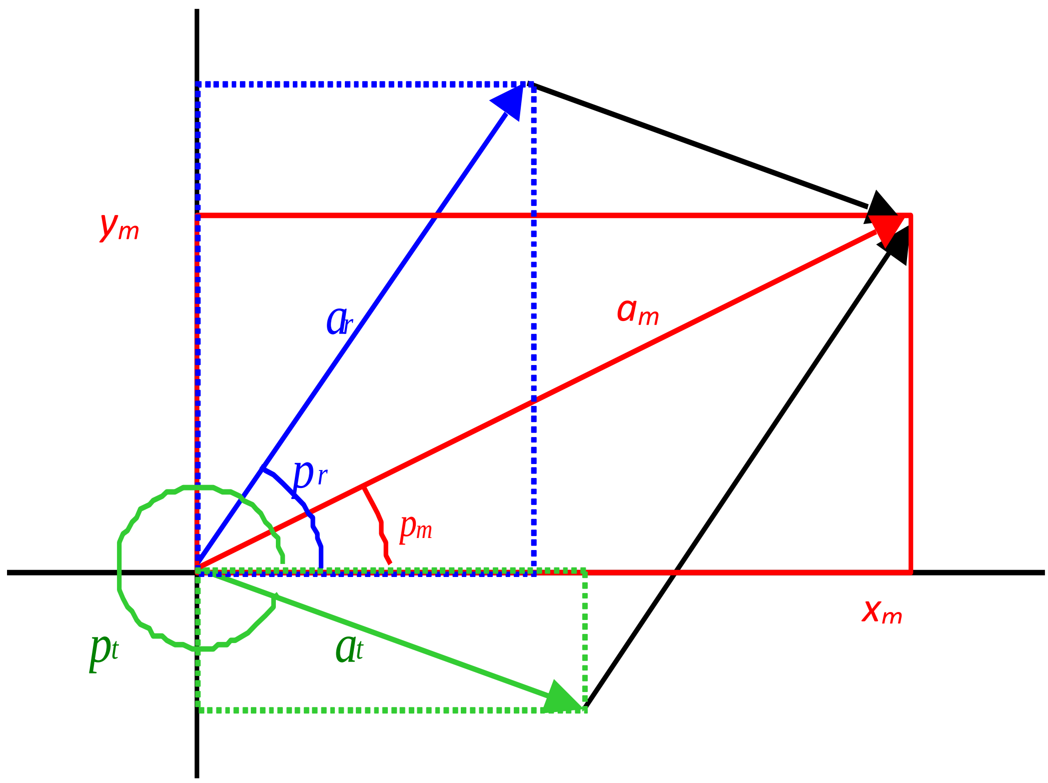


Fig 4 Vector diagram of the first phases of rainfall (ar) and temperature (at) with the resultant vector (am)

The resultant phase angle and amplitude are then:

$$y_m = a_r \cos p_r + a_t \cos p_t$$

$$x_m = a_r \sin p_r + a_t \sin p_t$$

$$a_m = \sqrt{y_m^2 + x_m^2}$$

$$p_m = angle\left({}^{x_m}\!/_{a_m}, {}^{y_m}\!/_{a_m} \right)$$

Unfortunately, this does not completely solve the problem of fitting a model to climates with different weather determinants. However, the vast majority of climates in the world are either:

(1) Rainfall determined where temperature is not an important seasonal effect (large areas of the tropics and subtropics);
(2) Temperature determined where rainfall is even throughout the year (most of the rest of the tropics and some temperate climates); or
(3) Rainfall and temperature determined when the two variates are highly correlated (summer rains - most of the rest of the world).

The Odd Man Out is:

(4) Winter rains and hot dry summers (almost only Mediterranean climates).

Luckily, the Mediterranean climates are at moderately high latitudes and we can afford to have the rotation dominated by temperature without losing generality in the rotations and comparisons. We therefore need to increase the weighting for the temperature vector smoothly as we approach the Mediterranean climates (in order to avoid a sudden swing). I found the following weightings to work well: $\mathbf{p} = \text{rainfall mm}$ and $\mathbf{t} = \text{temperature} \times 2 \times \text{abs(latitude)}$.

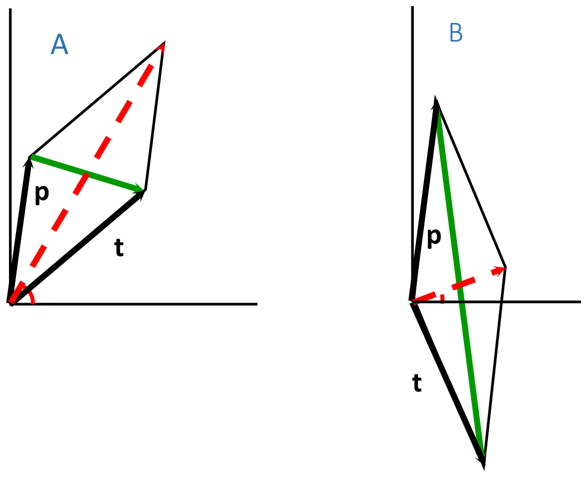


Fig 5 Illustration of concordant and discordant weighted rotation vectors

In A the two vectors of **p** and **t** (the bold black lines) are pointing in the same direction, in B they are pointing in disparate directions. The dashed red line shows the sum of the vectors and the green line shows their difference. From these I can calculate an index of reliability of the rotation as arc tan (sum/difference)

The index marked OK at 0.79 indicates (see Fig 7) an angle of 90 degrees between the two vectors if they are of equal length that denotes a reasonably stable rotation angle. It also corresponds to the case where one vector dominates the phase angle, which is also acceptable.

The perfect confidence is found in the subtropics there the two vectors point in same direction.

The equatorial regions and some Mediterranean regions have vectors pointing to different degrees in opposite direction. Luckily, none of these areas reaches the level where the phase angle is completely indefinite. The highest index noted is about 1.1, which is equivalent to an angle of about 120 degrees between two vectors of the same length. While this is somewhat indeterminate, there is still enough purchase to get a unique phase angle for rotation

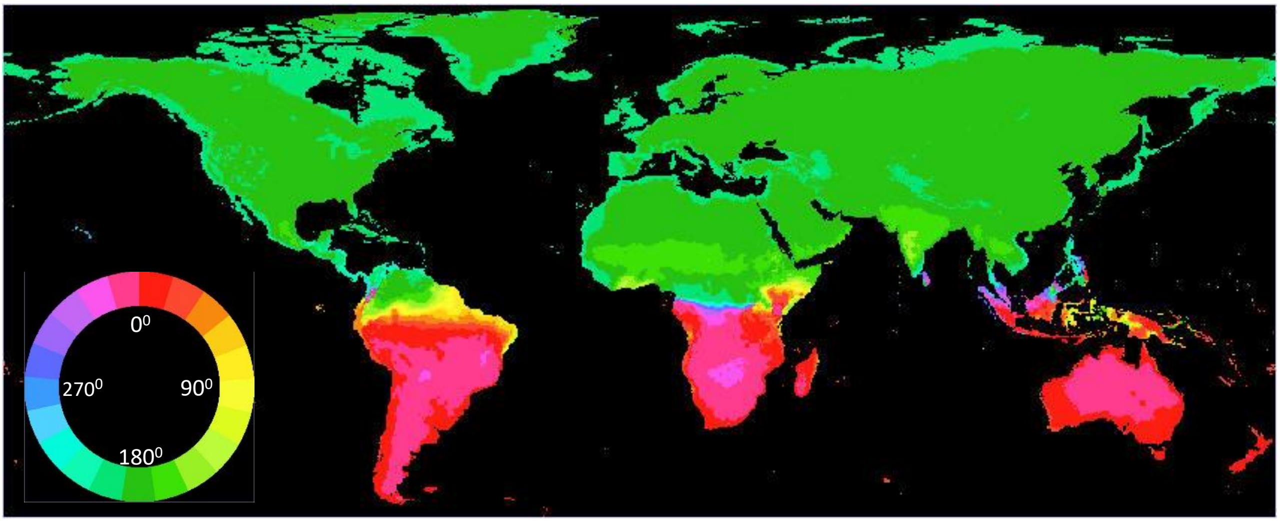


Fig 6 The rotation angle of world climates.

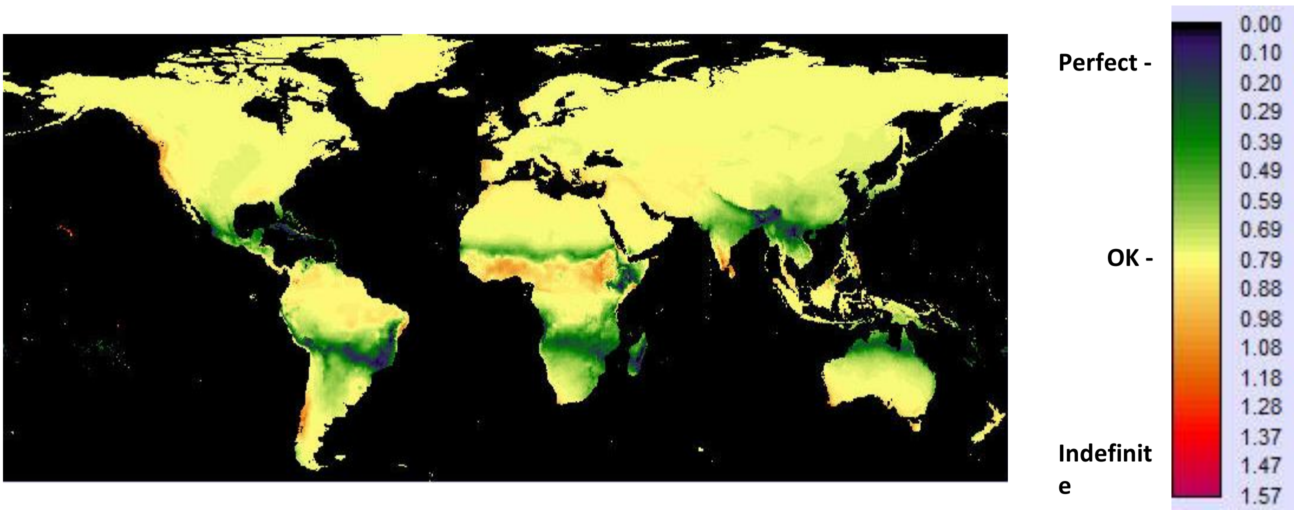


Fig 7 Confidence in the rotation angle of the world climates.

MetGrid_Handler module

Accessing MetGrid files from Fortran

A Fortran module contains data structure definitions that specify defined type variables that allow the grouping of various types of data into structured data units, and sets of functions or subroutines that use these units or operate on them. The module is invoked from within a Fortran program with the *use* command thus: use MetGrid_Handler

Module Description

Variable structures

The structure most likely to be interpreted by the user, in that the internal variables will be used in a Fortran program is the climate_structure. Most of the others are specific to the internal workings of the module and so are only minimally described. This structure normally only exists during a run unit of the program.

```
type climate_structure
        real*4        ::    lat,        &    ! decimal degrees latitude
                            long,       &    ! decimal degrees longitude
                            phase,      &    ! radians rotation angle
                            confidence       ! unused when constructed from MetGrid record
        real*4        ::    rain(12)         ! mm per day
        real*4        ::    temp(12)         ! mean temperature degrees centigrade
        real*4        ::    diurn(12)        ! diurnal temperature range degrees centigrade
        real*4        ::    srad(12)         ! solar radiation Mj m-2 day-1
        real*4        ::    raindays(12)     ! dimensionless 0 to 1
        integer*4     ::    elev             ! metres above sea level
        logical*1     ::    rotated          ! climate structure may be rotated or not
end type climate_structure
```

type cli_record	A structure to hold the data for a CLI record
type Fourier	The Fourier coefficients of a 12 monthly climate array
type metgrid_record	The compressed form of the climate_structure
type index_element	A 3 byte element of the index file holding a count of land or sea pixels
type metgrid_hdr	A structure to hold the information contained in a MetGrid header file
type pane_descriptor	Used only for the production of compressed panes of CLI files

Public variables

Some variables are used within the module, but are defined as public as they may be of use in a program using the module. These include halfpi, pi and twopi as real*4, eof, year, month and day as integer*4 and month_code as an array of 12 three byte character representations of the calendar month. If the MetGrid header has been read then it is held in the variable h. Lastly, two logical*4 variables header_loaded and index_loaded indicate if the respective loading has been done.

The index arrays are defined as private and are only accessed within the module; module routines to access MetGrid records use the loaded index arrays.

Function descriptions

As with the variable structures, not all module functions are necessarily called by the end user. Some have to be included because they are called by end-user functions; others are there because they are of use in the administration of the files.

Grid functions

G01 integer*4 function grid_col (long)
G02 integer*4 function grid_row (lat, error)
G03 real*4 function grid_lat (row)
G04 real*4 function grid_long (col)

Calendar functions

C01 pure real*4 function month_days (month, year)
C02 pure integer*4 function rotate_month (m, phase)
C03 pure logical function leap (year)

Input functions

I01 type (metgrid_record) function find_met_record (lat, long, unit, error)
I02 type (metgrid_record) function nearest_met_record (lat, long, max_distance, unit)
I03 type (cli_record) function read_cli (filename)
I04 type (metgrid_hdr) function read_metgrid_header (path, version)
I05 subroutine load_metgrid_index (path, version)
I06 subroutine open_metgrid_files (path, version, unit)
I07 subroutine close_metgrid_files (unit)

Output functions

```
!O01
   subroutine print_climate(c)
!O02
   subroutine print_metgrid_record(m)
!O03
   subroutine write_climate(c,unit)
!O04
   subroutine write_cli(cli,version,path)
!O05
   subroutine write_pane_descriptor(p,path)
!B01
   type(climate_structure) function  make_climate_structure(m) result(c)
!B02
   type(climate_structure) function null_climate()result(c)
!B03
   type(cli_record) function make_CLI_from_climate_structure(c)result(cli)
!B04
         type (climate_structure) function rotate_climate(ctc,phase)result(c)
!B05
         type(metgrid_record) function compress_met_record(c) result(m)
!B06
   type(metgrid_record) function null_metgrid_record()result(m)
!R01
```

```
   type(climate_structure) function new_rotate(c)result(r)
!R02
   subroutine neg_correct(v)
!R03
   subroutine rotate(a,thru)
!R04
   subroutine decode(q,v)
!R05
   subroutine encode(v,q)
!R06
   subroutine freq1(q,f)
!R07
   subroutine frqinv1(f,q)
!R08
   real*4 function angle(sin,cos)result(a)
!R09
   subroutine frrota(q,thru)
!M01
   subroutine pack12(v,n,p)
!M02
   subroutine unpack12(p,n,v)
!M03
   type (pane_descriptor) function pane(row,col,version)result(p)
!M04
   integer*4 function version_index(ver)result(ind)
```

Examples

CLI files

Reason

Structure

These are basic input files for the DSSAT weather generators and are used as input to the executable version of MarkSim (MarkSim_Standalone) which allows researchers without detailed Fortran knowledge to run multiple geographic points for interface with DSSAT. This is particularly useful for mapping crop responses over geographic areas.

The CLI files are compressed with WinRAR into panes of files with differing number of files depending on the existence of land based climate data within each pane. The size of the pane is adjusted for the resolution of the MetGrid so that a maximum of 14400 files can be present in each pane. At a resolution of 10 arc minutes, each pane is 20 degrees on a side, at five arc minutes each pane is 10 degrees and at 2.5 arc minutes, 5 degrees on a side.

At 30 arc seconds, the pane would have been 1 degree on a side, but the processing was estimated to take well over 30 days and so it was not attempted at this resolution.

Structure

User Access

List of figures

Fig 1 Illustration of the different data structures of MetGrid and WorldClim

Fig 2 Monthly rainfalls for Northville and Southville.

Fig 3 Illustration of the rigid rotation of a MetGrid record

Fig 4 Vector diagram of the first phases of rainfall (a_r) and temperature (a_t) with the resultant vector (a_m)

Fig 5 Illustration of concordant and discordant weighted rotation vectors

Fig 6 The rotation angle of world climates.

List of tables

Table 1 Typical rainfalls for imagined Mediterranean climates

References

Fick S.E and Hijmans R.J. 2017 WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. International Journal of Climatology, 2017 - Wiley Online Library

Hijmans, R.J., Cameron, S., Parra, J., Jones, P. G. and Jarvis, A. (2005) WorldClim, a very high resolution global terrestrial climate surfaces for monthly temperature and precipitation. International Journal of Climatology **25**: 1965–1978

Jones, P. G. and Thornton, P. K. 1993. A rainfall generator for agricultural applications in the tropics. Agric. For. Meteor.63:1-19.

Jones, P. G. and Thornton, P. K. 1997. Spatial and temporal variability of rainfall related to a third order Markov model. Agric. Forest Meteor. 86:127-138.

Jones, P. G. and Thornton, P. K. 1999. Fitting a third order Markov rainfall model to interpolated climatic surfaces. Agric. Forest Met.97:213-231

Jones P. G. and Thornton P. K. 2000. MarkSim: Software to generate daily weather data for Latin America and Africa. Agron. J 93:445-453.

Jones, P. G., Thornton, P. K., Diaz, W. and Wilkens, P. W. (2002) MarkSim, Version 1. A computer tool that generates simulated weather data for crop modelling and risk assessment. CIAT CD-ROM series, Cali, Colombia. Centro Internacional de Agricultura tropical. CD-ROM + Guide (87p.)

Jones P. G. and Thornton P. K. 2013. Generating downscaled weather data from a suite of climate models for agricultural modelling applications. Agricultural Systems 114 (2013) 1–5, Elsevier

Köppen, W., (1884) Die wärmezonen der erde, nach der dauer der heissen, gemässigten und kalten zeit und nach der wirkung der wärme auf die organische welt betrachtet. – Meteorol. Z.1, 215–226.