

**A user guide for  
ClimGen: a flexible tool for generating  
monthly climate data sets and scenarios**

Timothy J. Osborn

*Climatic Research Unit  
School of Environmental Sciences  
University of East Anglia  
Norwich NR4 7TJ, UK*

*t.osborn@uea.ac.uk*

**ClimGen version 1-02**

**<http://www.cru.uea.ac.uk/~timo/climgen/>**

2<sup>nd</sup> February 2009

## Contents

<b>INTRODUCTION AND ACKNOWLEDGEMENTS</b>	<b>3</b>
<b>KEY STEPS FOR USING CLIMGEN</b>	<b>4</b>
<b>CLIMGEN INPUTS THAT ARE MODIFIABLE</b>	<b>6</b>
<b>scen . spec</b> – specifies the emissions scenario and GCM climates to use for generating scenarios.	7
<b>slice . spec</b> – specifies the time periods to generate the scenarios for, including whether they are time series or time means.	10
<b>season . spec</b> – specifies the “seasons” to output (not necessarily standard seasons, since they could be individual months, averages of various months, or even annual means).	11
<b>var . spec</b> – specifies the climate variables for which scenarios will be generated.	12
<b>space . spec</b> – specifies the regions for which output is required, which could be single grid boxes, all individual grid boxes, or regional averages over specified grid boxes.	13
<b>CLIMGEN SOURCE CODE FILES AND SUBROUTINES</b>	<b>16</b>
<b>REFERENCES</b>	<b>17</b>

## Introduction and acknowledgements

ClimGen was developed by Tim Osborn and Tim Mitchell at the School of Environmental Sciences (ENV), University of East Anglia (UEA), in Norwich, UK. A number of others are acknowledged for various types of help, advice, testing and data processing, including Craig Wallace, Ian Harris and Nigel Arnell. A description of the data sets and the scientific basis for the algorithms used in ClimGen is given in *Osborn et al.* (2009), building on earlier work reported by Mitchell *et al.* (2004) and Goodess *et al.* (2003). This document is restricted to describing the usage, inputs and outputs of the ClimGen software.

ClimGen can be used as a stand-alone application that takes its inputs from a fixed set of files and produces a set of output files. That is the version that is described in this user guide. ClimGen can also be coupled to other modules, perhaps providing alternative inputs and/or passing the outputs to climate impacts modules. One example is that ClimGen can be used as the climate scenario generator in the Tyndall Centre's modular Community Integrated Assessment System (CIAS; *Warren et al.*, 2008). In order to couple ClimGen into the CIAS or to other modules, modifications to the ClimGen inputs and outputs might be required – these are not described in this user guide.

This version of the user guide relates specifically to version 1-02 of ClimGen. This version has been developed for use in the Global Scale Impacts (GSI) project funded by the QUEST programme of the UK Natural Environment Research Council (NERC). The QUEST–GSI project is coordinated by Nigel Arnell at the University of Reading's Walker Institute.

ClimGen version 1-02 incorporates climate change information diagnosed from the ocean-atmosphere general circulation model (OAGCM) simulations that were undertaken for the World Climate Research Programme's (WCRP's) Coupled Model Intercomparison Project phase 3 (CMIP3; *Meehl et al.*, 2007), with the results being widely reported in the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC, 2007). We acknowledge the modelling groups, the Program for Climate Model Diagnosis and Intercomparison (PCMDI; [www.pcmdi.llnl.gov](http://www.pcmdi.llnl.gov)) and the WCRP's Working Group on Coupled Modelling (WGCM) for their roles in making available the WCRP CMIP3 multi-model dataset. Support of this dataset is provided by the Office of Science, U.S. Department of Energy.

## Key steps for using ClimGen

### 1. Installation

- (i) Select a location (i.e., directory) to install ClimGen. For example, in my installation it is `/home/user/appl/climgen/v1-02`. It is a good idea to include the version number, so that you can install later versions without losing your existing data. Throughout this user guide, the directory that ClimGen is installed in is given by `~`.
- (ii) Copy or download all files, including Fortran source code, input data files, and the specification files used to select options, into this directory `~`, maintaining the same subdirectory structure.
- (iii) Edit the text file `~/f90/climgen.properties` to contain the full path of directory `~`. For example, in my installation, the file `~/f90/climgen.properties` contains `/home/user/appl/climgen/v1-02/` on its first line (all subsequent lines are ignored). Note the trailing `/'` which must be included.

### 2. Compilation

ClimGen is written in Fortran90. It has been successfully compiled using the systems/compilers given in Table 1. The compilation commands used in each case are stored in `~/f90/batch_compile.job`. Other Fortran95 and Fortran90 compilers will probably work. Fortran77 compilers will definitely not work. Lines that may need to be altered to compile on different systems or with different compilers are given in Table 2.

**Table 1. Compilers that have successfully compiled ClimGen.**

System	Compiler
Compaq Unix	f90 (i.e., the built-in Compaq compiler)
OpenSuse Linux	gfortran

**Table 2. Source code lines that may be system and/or compiler dependent.**

Source file	Lines	Statements / Comments
out.f90	185	<pre>call system("mv " // trim(File) // " " // trim(FileOld),systemstat)</pre> <p>This line may be compiler dependent and system dependent. Its purpose is to rename an existing output file to avoid it being overwritten by the new run of ClimGen if the new run is producing an output file with the same name as an existing file. If it is not possible to make this line work on your system, then this section of the code can be commented out and ClimGen will still work, though you will receive no warning if existing output files are being overwritten.</p> <p>The compiler dependent part is the <code>system</code> subroutine, which calls an operating system command. The system dependent part is the operating system command being used to rename the existing file, which is here the UNIX/LINUX command <code>mv</code>.</p>

### 3. Specify options to generate each set of scenarios

Each time ClimGen is used, it reads a number of specification files to determine what it should do. These specify the variables, the climate models, the observations, the scenarios and the time periods required. These are text files that can be modified to indicate particular requirements that the user has. Most of the remaining sections of this user guide contain a detailed explanation of these options. The specification files are stored in `~/f90/spec/*.spec`. I also keep subdirectories containing frequently used specification files, and then I make a logical link (similar to an alias) from an existing specification file to the ones that ClimGen actually reads. An alternative would be to copy

the existing specification files that you choose into the ones that ClimGen actually reads. Specification files can be edited by hand or created using `~/f90/makespec`.

**For the QUEST-GSI project, all specification files that have been used to generate the priority scenarios have filenames beginning with `quest-` (except for the variables specification files, which are not specific to the QUEST-GSI scenarios).**

#### 4. Execute ClimGen

Change to the ClimGen source code directory `~/f90/` and execute the compiled `climgen` command. While executing, ClimGen displays various diagnostics to indicate its progress. In particular, all files that are read from or written to during execution are listed, and warning or error messages may be given if problems arise. Inspect these carefully to ensure that execution was successful.

#### 5. Inspect the ClimGen output files

ClimGen output files are text files written to `~/out/main/`. Most have unique filenames that reflect the various options selected in the specification files. Where filenames are not unique, differences are mostly indicated by different header lines at the start of the file. When filenames are not unique and a file with the same name already exists, ClimGen attempts to rename the existing file by adding a trailing `.nnn` unique number, to avoid overwriting the existing file. If it is unable to do this, a warning message is displayed and the existing file is overwritten.

The file naming convention and the content and format of the files are described in separate documents, both available here: <http://www.cru.uea.ac.uk/~timo/climgen/#refs>.

## **CLIMGEN inputs that are modifiable**

The inputs described here are those that should be modified prior to running `climgen` to produce the required scenario output. Other inputs are the observational and simulated data sets from which the scenarios are generated, and these are listed as “fixed” – even though they can, of course, be changed/updated if necessary.

The scenarios that `climgen` produces are determined by options given in the five specification files. The structure and content of these files are described below. Though they can be created or edited manually via a text editor, it is also possible to use `makespec` to create new specification files by selecting from a list of various options.

The five specification files are read from `~/spec/`. To allow various alternative specifications to be kept, subdirectories `scen`, `slice`, `season`, `var` and `space` are provided where many alternatives can be kept (with unique, and preferably descriptive, names). Choice of specifications for a particular run of `climgen` can then be made by using logical links/shortcuts to match the input files to the required specification files.

The following sections provide information about the structure (including Fortran format statements) and meaning of the specification files.

**scen.spec** – specifies the emissions scenario and GCM climates to use for generating scenarios.

	NScen	Method			(2i6)
	ObsArchive	StartYear	EndYear		(a11,2i5)
NScen *	{	ModArchive	Pattern	Forcing	InitialYear (a5,x,a24,x,a8,x,i4) }

NScen determines the number of scenarios to create.

Method determines the scenario generation method:

1. Returns **observations**. When Method=1, the only other input that is read from scen.spec is ObsArchive; if there are any other items in the file they are ignored.
2. Makes future scenario using an **additive** method (sometimes called “absolute changes” or the “ATEAM” method because this method was used to generate the climate scenarios used in the ATEAM project – Mitchell et al., 2004; Schröter et al., 2005). The scenarios are generated by adding GCM-derived changes in mean climate to the observed record, thus keeping interannual variability unchanged from the present-day climate.
3. Same as 2 for all variables except cloud cover, precipitation and wet-day frequency. Changes in **cloud cover** are also applied additively (the same as for Method=2), thus keeping interannual variability of cloud cover unchanged from the present-day climate. However, the changes are not a linear function of the global-mean temperature, but are instead based on a logistic function that allows the changes to approach zero or full cloud cover gradually. Most results are very similar to those obtained using Method=2; the only exceptions are cases where the cloud cover approaches 0% or 100%, and here the changes will be reduced to approach 0% or 100% gradually. Changes in mean **precipitation** are applied multiplicatively (the method is sometimes called “relative changes”). The observations are multiplied by the GCM-derived changes in mean precipitation. Because this multiplication is applied to the observed time series including its interannual variability, the variability is also modified in such a way that the coefficient of variation (ratio standard deviation to mean) is roughly constant. Another difference is that the magnitude of the mean precipitation change is an exponential function of global-mean temperature change rather than a linear function. This avoids obtaining zero precipitation in regions of decreased mean precipitation, because the rate of change decelerates as temperature increases, but it results in accelerating changes in regions of increased mean precipitation. Changes in **wet-day frequency** are calculated in the same way as 2, but because this is simply a non-linear function of the monthly precipitation time series, which will themselves be different for method 3 than for method 2, the resulting wet-day frequency time series will also be different.
4. Same as 3 for all variables except precipitation and wet-day frequency. Changes in mean **precipitation** are obtained in the same way as 3, by multiplying the observations by the GCM-derived changes in mean climate. But the interannual variability is also modified according to GCM-derived changes in precipitation probability distributions (parameterised via the shape parameter of the gamma distribution), using the so-called “Osborn” method. The changes in mean precipitation and precipitation distribution shape are linear functions of global-mean temperature in regions where they increase, and exponential functions of global-mean temperature in regions where they decrease. As in 3, the **wet-day frequencies** depend on the monthly precipitation time series, and thus will be different from those obtained using other methods.
5. Obsolete: do not use!

Further details regarding these scenario generation methods are provided in Osborn et al. (2009).

ObsArchive selects the observational archive to be used, either as direct output (if observed data are requested by Method=1) or for providing the interannual variability that is combined with a climate change signal to obtain a scenario (if scenario data are requested by Method>1). Currently available options are:

- cru\_ts\_2\_10 Observations from CRU TS 2.10 for 1901–2002 for tmp, tmn, tmx, dtr, pre, wet, cld and vap
- cru\_ts\_3\_00 Observations from CRU TS 3.00 for 1901–2005 for tmp, tmn, tmx, dtr, pre, wet and vap (note: not for cld!)
- hadisst\_1 Observations from HadISST1 for 1901–2007 for sst

StartYear and EndYear specify the range of observed anomalies to use when combining observed variability with a climate change signal to generate a climate scenario. If observed data are requested (Method=1), then these are ignored (and do not need to be present) because the range of years used is determined by the time periods requested in slice.spec. For scenarios, however, the range of observed anomalies specified by StartYear and EndYear is used, repeating the sequence if the range of output years is longer than the range of observed anomalies. For example, if StartYear=1961 and EndYear=1990 and a scenario 2010 to 2100 is to be generated, then the observed sequence from 1961 to 1990 might be used for 2010 to 2029 and then repeated for 2030 to 2059 and again for 2060 to 2099, and then finally the 1961 anomaly will be repeated once more for scenario year 2100. Note that the starting year in the sequence can be controlled by InitialYear (see below) for each scenario individually. By setting StartYear=EndYear, it is possible to generate a climate scenario which has no interannual variability and therefore reveals the climate change signal only (avoid using Method=4 for mean precipitation or wet-day frequency, because for this method the scenario will contain a fixed, but non-zero, anomaly when StartYear=EndYear).

ModArchive selects the climate model archive to be used to provide the climate change patterns. Currently available options are:

- LINK Climate change patterns from the Climate Impacts LINK Project archive for ukmo\_hadcm3
- DDC Climate change patterns from the IPCC Data Distribution Centre (DDC) archive for cccma\_cgcm2, csiro\_mk2, mpi\_echam4opyc and ncar\_pcm
- CMIP3 Climate change patterns from the CMIP3 archive (as used extensively in the Working Group 1 fourth assessment report of IPCC, published in 2007) for 22 GCMs

Note that it is strongly recommended to use CMIP3 because (a) these models are more recent; (b) the patterns are diagnosed using a consistent and documented method; and (c) in the LINK and DDC archives, some cloud cover change patterns may contain errors.

Pattern determines which GCM is used to provide the pattern (and possibly time series) of climate change. The identifier must combine the exact GCM name used by ClimGen (which combines the institution information and the model name) with an indication of the source of the pattern (which is usually an indication of which simulation(s) were used to diagnose the patterns). Note that the indicator of which simulation(s) were used to diagnose the patterns must not be confused with the climate forcing for which the scenario is to be generated – i.e., it is possible to use the pattern diagnosed from the A1B simulation to generate a scenario under any other scenario. Also note that there are some patterns averaged across all GCMs within the archive, labelled multimodelmean or MMMCMIP3 (multi-model-mean from CMIP3), which have been used for GCMs that do not have diurnal temperature range data output. For CMIP3 the options currently available are:

- bccr\_bcm20\_\_\_\_\_A2A1B\_\_\_\_\_
- cccma\_cgcm31\_\_\_\_A2A1B\_\_\_\_\_ or cccma\_cgcm31\_\_\_\_MMMCMIP3\_\_\_\_
- cccma\_cgcm31t63A1B\_\_\_\_\_ or cccma\_cgcm31t63MMMCMIP3\_\_\_\_



```

ccsr_miroc32hi_A1B_____
ccsr_miroc32medA2A1B_____
cnrm_cm3_____A2A1B_____ Or cnrm_cm3_____MMMCMIP3_
csiro_mk30_____A2A1B_____
csiro_mk35_____A2A1B_____
gfdl_cm20_____A2A1B_____ Or gfdl_cm20_____MMMCMIP3_
gfdl_cm21_____A2A1B_____ Or gfdl_cm21_____MMMCMIP3_
giss_aom_____A1B_____
giss_modeleh_____A1B_____ Or giss_modeleh_____MMMCMIP3_
giss_modeler_____A2A1B_____ Or giss_modeler_____MMMCMIP3_
iap_fgoals10g_____A1B_____ Or iap_fgoals10g_____MMMCMIP3_
inm_cm30_____A2A1B_____
ipsl_cm4_____A2A1B_____ Or ipsl_cm4_____MMMCMIP3_
mpi_echam5_____A2A1B_____ Or mpi_echam5_____MMMCMIP3_
mri_cgcm232a_____A2A1B_____ Or mri_cgcm232a_____MMMCMIP3_
ncar_ccsm30_____A2A1B_____
ncar_pcm1_____A2A1B_____ Or ncar_pcm1_____MMMCMIP3_
ukmo_hadcm3_____A2A1B_____ Or ukmo_hadcm3_____A2_____
ukmo_hadgem1_____A2A1B_____ Or ukmo_hadgem1_____MMMCMIP3_

```

**Forcing** determines the time evolution of climate according to an emissions scenario or other indicator of climate change magnitude. The first character of **Forcing** determines the type of climate change scenario:

- f** a fixed global-mean temperature change for then entire scenario period, whose magnitude (relative to 1961–1990) is given by the subsequent characters (e.g., **f1.0** is a scenario with a fixed global-mean warming of 1.0°C, **f3.5** is a scenario with a fixed global-mean warming of 3.5°C, etc.).
- s** a transient scenario that is independent of the choice of GCM pattern (e.g., **sA1B\_25** will generate a scenario under the global-mean temperature change given in file `~/globalt/specialseries__sA1B_25.ann`, regardless of which GCM is selected by **Pattern**).
- anything else a transient scenario that is dependent on the choice of GCM pattern (e.g., **A1B\_MG1** is the MAGICC simulation of global-mean temperature under the A1B scenario with MAGICC parameters tuned to emulate the GCM behaviour for the GCM identified by **Pattern**). So, if **Pattern=ukmo\_hadcm3\_\_\_\_\_A2A1B\_\_\_\_\_** and **Forcing=A1FIMG1**, the scenario will be generated using the climate change pattern diagnosed from the pooled A2 and A1B simulations with HadCM3, together with the HadCM3-tuned version of MAGICC simulating global-mean warming under the A1FI scenario.

**InitialYear** indicates the year whose observed anomaly is used for the first output year in each scenario. **InitialYear** must lie within the range of **StartYear** and **EndYear** (see above). For example, if **StartYear=1961**, **EndYear=1990**, **InitialYear=1987**, and a scenario 2006 to 2100 is to be generated, then the observed sequence from 1987 to 1990 will be used for 2006 to 2009, the 1961 to 1990 sequence is used for 2010 to 2029 and then repeated for 2030 to 2059 and again for 2060 to 2099, and then finally the 1961 anomaly will be repeated once more for scenario year 2100.

---

---

**slice.spec** – specifies the time periods to generate the scenarios for, including whether they are time series or time means.

---

Descriptor	(a3)
NSlice	(i6)
NSlice * {	BeginYear EndYear (2i5) }

---

`Descriptor` is a 3-character label used for generating the ClimGen output filename to distinguish between cases with different time averaging. The user is free to use any label of their choice, but suggestions are `ann` for output with individual values in each year, `y10` or `dec` for output of decadal means, `y30` for output of 30-year means, etc.

`NSlice` determines the number of time slices to create. If you want a time series with values every year (whether those values be monthly, seasonal or annual means) from, say, 2000 to 2029, then set `NSlice=30`. If you want the decadal means from 2000–2009, 2010–2019 and 2020–2029, then set `NSlice=3`. If you want the 30-year means from 2000–2029 and 2070–2099, then `NSlice=2`.

`BeginYear` and `EndYear` determine the start and end year of each time slice. If `BeginYear=EndYear` then the time slice will be a single year.

---

---

---

**season.spec** – specifies the “seasons” to output (not necessarily standard seasons, since they could be individual months, averages of various months, or even annual means).

---

Descriptor	(a7)
NSeason	(i6)
NSeason * { J F M A M J J A S O N D BeginMonth (13i3) }	

---

Descriptor is a 7-character label used for generating the ClimGen output filename to distinguish between cases with different seasonal averaging. The user is free to use any label of their choice, but suggestions are `annmean` for output of annual means, `seasons` for output of standard climatological seasonal means (MAM, JJA, SON, DJF), and `monthly` for output of all 12 months.

NSeason determines the number of seasons to output. If you want every single month individually, then set `NSeason=12`. If you want the four standard seasons and the annual mean, then set `NSeason=5`.

J, F, M, A, M, J, J, A, S, O, N and D determine which months of the year to include in each seasonal mean, set to `-1` to include a month and `0` to exclude it. Included months do not need to be adjacent, so that any “seasonal” mean can be constructed. Note that for precipitation and wet-day frequency, the values are **totals** for each output “season”, not **averages**.

BeginMonth specifies the start month of the season, to allow seasons that span the change of year. For example, to obtain a boreal winter season December–February, set `BeginMonth=12` so that the season is calculated from the December value of one year and finishes with the January and February values of the following year. Note that the final output value will be set to the missing code whenever the seasonal mean spans two years, because the January (and subsequent) values from the following year will not be available when the end of the time period is reached.

---

---

**var.spec** – specifies the climate variables for which scenarios will be generated.

---

NVar (i6)

NVar \* { Variable (a4) }

---

NVar determines the number of variables to create scenarios for.

Variable identifies the variables to be used, from the following options:

.tmp	Near-surface temperature (°C)
.tmn	Near-surface minimum temperature (°C)
.tmx	Near-surface maximum temperature (°C)
.dtr	Near-surface diurnal temperature range (°C)
.pre	Precipitation (mm)
.wet	Wet-day frequency (days)
.cld	Cloud cover (%)
.vap	Vapour pressure (hPa)
.sst	Sea surface temperature (°C)

For output of observations only (Method=1) from the CRU TS 2.10 archive (ObsArchive=cru\_ts\_2\_10) only, two further variables are available:

.rhm	Relative humidity (%)
.spc	Sunshine (%)

Note first that the '.' should be included in the name. Note second that for precipitation and wet-day frequency, the values are **totals** for each output "season", not **averages**.

---

**space.spec** – specifies the regions for which output is required, which could be single grid boxes, all individual grid boxes, or regional averages over specified grid boxes.

---

	Descriptor	(a16)
	NX            NY        NRegion WholeReg	(4i6)
	MeshY        MeshX BoundN    BoundE    BoundS BoundW	(6f8.2)
NY *	{ AreaWt	(f10.4) }
NX*NY *	{ RegionNo	(i10) }
NRegion *	{ RegionName	(a20) }

---

**Descriptor** is a 16-character label used for generating the ClimGen output filename to distinguish between cases with different spatial averaging or spatial extraction. The user is free to use any label of their choice, but suggestions are `countrymeans` for output of national means, `landboxes` for output of all land grid boxes individually, `seaboxes` for output of all ocean grid boxes individually, etc.

**NX** and **NY** specify the dimensions of the gridded data sets to use. There are currently two options, though it would be possible to use other grid dimensions if the observed and climate change pattern data sets were provided on other grids.

72	36	for the observed temperature and precipitation on the 5° resolution grid
720	360	for the observed and scenario variables on the 0.5° resolution grid

**NRegion** specifies the number of regional averages to calculate and output. If output from every grid box is required, then each one is a separate region and `NRegion=NX*NY`. If a global-mean of all grid boxes is required, then `NRegion=1`.

**WholeReg** determines what to do if data for some grid box values within a chosen region are missing. If a region is defined to include grid boxes that have either no observations (e.g., ocean-only boxes for variables available only over land) or no climate change pattern (e.g., for precipitation variability changes in regions that are too dry to define the variability), then the regional average will be set to missing if `WholeReg=-1` or will be calculated from the available non-missing grid box values if `WholeReg=0`.

**MeshY** and **MeshX** are the grid box sizes in the latitudinal and longitudinal directions, respectively. There are currently two options, though it would be possible to use other grid sizes or resolutions if the observed and climate change pattern data sets were provided on other grids.

5.00	5.00	for the observed temperature and precipitation on the 5° resolution grid
0.50	0.50	for the observed and scenario variables on the 0.5° resolution grid

**BoundN**, **BoundE**, **BoundS** and **BoundW** are the boundaries of the gridded fields to be used. They are in degrees of latitude or longitude. Positive latitudes indicate north of the equator and negative latitudes indicate south of the equator. Positive longitudes indicate east of the Greenwich meridian, and negative longitudes indicate west of the meridian, though a grid that spans the 180°E/W should be specified using positive longitudes (<180 for `BoundE` and >180 for `BoundW`). The boundaries

must be specified as the *outer edge* of the edge grid boxes, not the centres of the boxes. There are currently two options, though it would be possible to use other grid domains if the observed and climate change pattern data sets were provided on other grids, and new grids do not need to be global in domain. For example, `climgen` should work with patterns of climate change derived from regional climate model (RCM) simulations, though the observed data sets would need to be generated for the same domain and resolution as the RCM data (no interpolation between grids is done within `climgen` itself).

90.00 180.00 -90.00 -180.00 for both the 5° and 0.5° resolution data sets

`AreaWt` is used to specify the area weighting to be used for boxes in each row of the climate grid. It is usually set to the cosine of the latitude of each row in the gridded field. There is no requirement for the weights to sum to 1, because they are used as relative rather than absolute weights. The weights for the entire global field must be supplied even if the regions defined in `space.spec` use only part of the field.

`RegionNo` is used to specify which of the regions each grid box belongs to. `RegionNo` must be specified for every grid box in the gridded field, even those that are not part of any region (in which case their `RegionNo` should be set to -999). They are listed in order from the south-west box, moving northwards in columns, and then east-wards, until the final value specifies the region to which the north-east box in the grid should contribute. Regions are numbered from 1 to `NRegion`, and it is this region number that must be specified for each box's `RegionNo`. Note that no grid box can contribute to more than one region in a given `space.spec` file. If overlapping regions are required, `climgen` must be run multiple times with different `space.spec` files.

`RegionName` specifies the name to be given to each region. They do not need to be unique, though this can be useful for identifying the climate scenario output that is produced for each region (the regions are output in the same order that they are defined in `space.spec`, but a unique name can still be useful).

**Creating new `space.spec` files is probably the most time consuming aspect to configuring `climgen` for a particular application, because the size of the specification arrays prohibits editing them by hand.** While `makespec` can be used to automate the generation of `space.spec` files that are suitable in some instances, particularly where individual grid boxes are required, or rectangular area averages are required (which can be limited to the land grid boxes only, within a rectangular region of land and sea), more intricate regional shapes cannot be created using `makespec`. Three such `space.spec` files for the 0.5° resolution data have been supplied (in `~/spec/space/halfdeg/`) that may be suitable for some purposes:

`countrymeans.spec` defines regional averages for every country in the world. Inspect this file to check its suitability: for example, most, but perhaps not all, "overseas territories" are treated as separate regions from their sovereign nations; also some current or previous ocean weather ship locations are included as "pseudo-land" single grid boxes with data and are allocated to an appropriate country in this regional country file (and indeed in the `cnt.spec` and `kop.spec` files too).

`cnt.spec` defines regional averages for a set of nine continental or sub-continental regions of the world. Inspect this file before using to check it is suitable for use: for example, Europe includes Greenland (which will therefore influence the European-mean climate considerably!) and also many smaller islands – it is not "mainland Europe".

`kop.spec` defines regional averages for 24 different regions categorised according to the Koppen climate classification. Note that there are some slight variations on the Koppen classes (though the allocation of each region to one of 24 Koppen classes of course depends on the climate data being used and the period of analysis, and this is not

documented for the `kop.spec` file, so the regional definitions used in `kop.spec` should be inspected before using them).

**Note that the most informative aspect of a `space.spec` file is typically the list of `RegionName` values at the end of the file, and thus when trying to determine the purpose of an undocumented `space.spec` file it is worth examining the last `NRegion` lines of the file, as well as plotting a map of the regional definitions themselves.**

---

## CLIMGEN source code files and subroutines

All Fortran90 source code is in subdirectory ~/f90/. Unless otherwise stated, all code was originally written by Tim Mitchell (Tyndall Centre, ENV, UEA, UK) and Tim Osborn (Climatic Research Unit, ENV, UEA, UK), based on the scientific methods developed by Tim Mitchell and Tim Osborn. The source code is split into the following files:

makespec.f90	engine.f90
<ul style="list-style-type: none"><li>• program MakeSpec</li></ul>	<ul style="list-style-type: none"><li>• subroutine MethodOsbornGamma</li><li>• subroutine MethodATEAM</li><li>• subroutine PreToWet</li><li>• subroutine TruncateRange</li></ul>
climgen.f90	
<ul style="list-style-type: none"><li>• program GetClim</li></ul>	
climgen.properties	agg.f90
<ul style="list-style-type: none"><li>• <b>Must be edited to contain the full path of the ClimGen software and data sets</b></li></ul>	<ul style="list-style-type: none"><li>• subroutine AggMonths</li><li>• subroutine AggYears</li><li>• subroutine AggBoxes</li></ul>
spec.f90	out.f90
<ul style="list-style-type: none"><li>• subroutine EnsureSpec</li><li>• subroutine ReviewSpec</li><li>• subroutine SpecDim</li><li>• subroutine SpecScen</li><li>• subroutine SpecVar</li><li>• subroutine SpecSeason</li><li>• subroutine SpecSlice</li><li>• subroutine SpecSpace</li><li>• subroutine MakeSpace</li><li>• subroutine MakeVar</li><li>• subroutine MakeSlice</li><li>• subroutine MakeSeason</li><li>• subroutine MakeScen</li><li>• subroutine IntToStr</li></ul>	<ul style="list-style-type: none"><li>• subroutine DumpGrim</li><li>• subroutine DumpByRegion</li></ul>
space.f90	vari.f90
<ul style="list-style-type: none"><li>• subroutine Rectangles</li></ul>	<ul style="list-style-type: none"><li>• subroutine CheckVariSuffix</li></ul>
load.f90	time.f90
<ul style="list-style-type: none"><li>• subroutine UnZipper</li><li>• subroutine LoadGlobalT</li><li>• subroutine LoadObs</li><li>• subroutine LoadGrid</li><li>• subroutine LoadGrim</li></ul>	<ul style="list-style-type: none"><li>• subroutine CommonVecPer</li><li>• subroutine GetMonthLengths</li></ul>
	inc_gam.f90
	<ul style="list-style-type: none"><li>• A library routine written by Alfred H Morris, Jr at Naval Surface Warfare Center, Dahlgren, Virginia, USA</li><li>• Subroutine GamInv [calculates the inverse incomplete gamma ratio function]</li><li>• Also contains various other routines required by GamInv</li></ul>
	con_gam.f90
	<ul style="list-style-type: none"><li>• Contains various routines used to provide constants to GamInv</li></ul>



## References

Goodess CM, Osborn TJ and Hulme M (2003) The identification and evaluation of suitable scenario development methods for the estimation of future probabilities of extreme weather events. Tyndall Centre Technical Report 4, Tyndall Centre for Climate Change Research, University of East Anglia, Norwich, 69pp.

IPCC (2007) Climate change 2007: the physical science basis; contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (eds. Solomon S, Qin D, Manning M, Chen Z, Marquis M, Averyt KB, Tignor M and Miller HL). Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 996 pp.

Meehl GA, Covey C, Delworth T, Latif M, McAvaney B, Mitchell JFB, Stouffer RJ and Taylor KE (2007) The WCRP CMIP3 multi-model dataset: a new era in climate change research. *Bulletin of the American Meteorological Society* **88**, 1383–1394.

Mitchell TD, Carter TR, Jones PD, Hulme M and New M (2004) A comprehensive set of high-resolution grids of monthly climate for Europe and the globe: the observed record (1901-2000) and 16 scenarios (2001-2100). Tyndall Centre Working Paper 55, Tyndall Centre for Climate Change Research, University of East Anglia, Norwich, 30pp.

Osborn TJ, Wallace CJ, Harris IC and Mitchell TD (2009) ClimGen: a flexible tool to generate monthly climate scenarios for global-scale impact assessment. In preparation.

Schröter D, Cramer W, Leemans R, Prentice IC, Araújo MB, Arnell NW, Bondeau A, Bugmann H, Carter TR, Garcia CA, de la Vega-Leinert AC, Erhard M, Ewert F, Glendining M, House JI, Kankaanpää S, Klein RJT, Lavorel S, Lindner M, Metzger MJ, Meyer J, Mitchell TD, Reginster I, Rounsevell M, Sabaté S, Sitch S, Smith B, Smith J, Smith P, Sykes MT, Thonicke K, Thuiller W, Tuck G, Zaehle S, Zierl B (2005) Ecosystem service supply and vulnerability to global change in Europe. *Science* **310**, 1333-1337 (doi: 10.1126/science.1115233).

Warren R, de la Nava Santos S, Arnell NW, Bane M, Barker T, Barton C, Ford R, Fuessel H-M, Hankin RKS, Klein R, Linstead C, Kohler J, Mitchell TD, Osborn TJ, Pan H, Raper SCB, Riley G, Schellnhuber HJ, Winne S and Anderson D (2008) Development and illustrative outputs of the Community Integrated Assessment System (CIAS), a multi-institutional modular integrated assessment approach for modelling climate change. *Environmental Modelling and Software* **23**, 592-610 (doi: 10.1016/j.envsoft.2007.09.002).