Tropical Cyclone Risk Model User Guide

GEOSCIENCE AUSTRALIA
RECORD 2011/73005

by

Nicholas Summons[1] and Craig Arthur[1]

Australian Government
Geoscience Australia

1. Coastal Marine and Climate Change Group, Environmental Geoscience Division

Licence

USER MANUAL

SOFTWARE

Acknowledgements

PEOPLE

- The Tropical Cyclone Risk Model (TCRM) was developed by Craig Arthur, Nicholas Summons, Nariman Habili, Geoff Xu and Xun Guo Lin. The software is currently being maintained and developed by Nicholas Summons and Craig Arthur.
- The software was reviewed and tested by Shane Martin and Nariman Habili before release.
- The TCRM icon and the front cover of this user manual were created by Adrian Yee.

THIRD PARTY LIBRARIES AND DATA

- L-moments package was developed by J. R. M. HOSKING, IBM RESEARCH DIVISION, T. J. WATSON RESEARCH CENTER, YORKTOWN HEIGHTS, NEW YORK 10598, U.
- KPDF.c was developed by Jon Saenz, Jesus Fernandez and Juan Zubillaga. Departamento de Fisica Aplicada II. Facultad de Ciencias. Universidad del Pais Vasco. Spain.
- NCEP-DOE Reanalysis 2 MSLP data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, from their web site at http://www.esrl.noaa.gov/psd/.
- International Best Track Archive for Climate Stewardship (IBTrACS) (http://www.ncdc.noaa.gov/oa/ibtracs/).

Contents

Introduction

PURPOSE

The Tropical Cyclone Risk Model (TCRM) is a computational tool developed by Geoscience Australia (GA) for estimating the wind hazard from tropical cyclones. TCRM uses statistical and parametric models to simulate tropical cyclone behaviour and effects; a statistical model to generate thousands of year's worth of storms that are similar to the input track data and a parametric wind field and boundary layer model to simulate the storm winds. An extreme value distribution is then fitted to the maximum winds at each location to estimate the wind hazard.

TCRM is free and open-source and licensed under the GNU General Public License version 3 and above (refer to the second page of this user guide for further details). The software is written in the Python programming language (http://www.python.org/) with the exception of a small amount of C code used to optimise the code. It can be run on either Windows or Linux machines and includes a graphical user interface to simplify the process of preparing a simulation.

Due to the flexible and modular design of the code, TCRM can also be employed for a wide range of applications. For example, it has been utilised:

- to inform emergency services of the possible impact of an approaching storm;
- to refine wind loadings maps to inform building design standards;
- as a component of full risk assessments; and
- for climate change studies.

AUDIENCE

This tool is intended for meteorologists, climatologists and natural hazard researchers with limited or no programming experience.

BACKGROUND

TCRM was born out of a need to assess cyclonic wind hazard in the Australian region. Though there are existing tropical cyclone risk models, the majority are proprietary models used by insurance and reinsurance firms to assess their portfolio risk. These models are also restrictive in the parameter space, implementing only one type of radial wind field or boundary layer model. Following an internal review of several proprietary tropical cyclone hazard models by GA, it was decided to proceed with the development of an open-source software package capable of providing a range of radial profiles and boundary layer models.

GA has also developed two other open-source hazard modelling packages - ANUGA (http://sourceforge.net/projects/anuga/) and EQRM (http://sourceforge.net/projects/eqrm/).

CAVEATS AND LIMITATIONS

TCRM has been designed primarily as a tool for assessing long term wind hazard. As such, it should not be applied as a forecasting tool. The software is also limited to assessing only wind hazard; other hazards associated with tropical cyclones such as flooding, landslides and storm surge inundation are not currently represented.

To estimate the long term wind hazard from a short historical record, TCRM generates several thousand years' worth of storms that are statistically similar to the input dataset. Whilst this process greatly reduces the sampling error, it can inadvertently introduce track and intensity biases.

Simulating several thousand years' worth of events is also computationally demanding, so TCRM must rely upon simplified parametric wind fields and linear boundary layer models rather than more realistic dynamical models.

The wind hazard generated by TCRM represents the regional three-second gust wind speed at the 10-meter height. It does not take into account local influences on the wind caused by topography, terrain and shielding for example. Whilst GA has methods to account for these effects, they are not currently incorporated as part of TCRM.

Getting Started

This chapter provides an overview of the steps required to install TCRM, prepare a configuration file and run a hazard simulation.

INSTALLING TCRM

To run TCRM, a PC running Windows (XP, Vista or 7) or Linux operating system with at least 1GB of RAM and several gigabytes of free disk space is required. The software is written mostly in the Python programming language, with the exception of a small amount of C code to improve computational speed. The current version of TCRM requires Python 2.5, 2.6 or 2.7 (32-bit) as well as five Python packages: NumPy, SciPy, Matplotlib, Basemap and wxPython. The installation instructions are listed below for both Windows and Linux systems.

Windows Installation

1. Install Python 2.7 (32-bit version) by running the self installer package found on the website: www.python.org.

2. Next, install each of the five python packages (NumPy, SciPy, Matplotlib, Basemap and wxPython) by running the installation files found on the websites listed below. Select the most recent stable releases (i.e. avoiding any with 'rc', 'b' or 'beta' file tags) for 32-bit windows with a python version corresponding to the version you have installed.

   NumPy
   http://sourceforge.net/projects/numpy/files/NumPy/

   SciPy (version 0.9.0 or above)
   http://sourceforge.net/projects/scipy/files/scipy/

   Matplotlib
   http://sourceforge.net/projects/matplotlib/files/matplotlib

   Basemap
   http://sourceforge.net/projects/matplotlib/files/matplotlib-toolkits/

   wxPython
   www.wxpython.org/

3. A 'gcc' C compiler is also required. This can be obtained by installing MinGW using the automatic downloader/installer found on the website: http://www.mingw.org.

4. Add the python and MinGW paths to the Windows path environment variable. For example, if you installed the software on drive C using default path names, then the following directories should be added to the path variable: 'C:\Python25' (for Python 2.5) and 'C:\MinGW\bin'. If you are unsure how to modify Windows environment variables, then please refer to the section "Setting Windows Path Environment Variables" located in the appendix.

5. Create a TCRM directory (i.e. C:\TCRM) and copy the compressed TCRM file to that location before extracting the contents.

6. Compile the C-code by running 'compile.cmd', located in the TCRM directory that was just created.

Linux Installation

The following instructions apply to Ubuntu 10.04 and above. We currently do not support other Linux distributions.

Note: It is not necessary to install Python or the C-compiler 'gcc' as these should have already been installed with Ubuntu.

1. Ensure your computer has an active internet connection.

2. The Basemap toolkit for matplotlib is not included as part of the standard software list for Ubuntu. This toolkit can be obtained by adding PyGeode (https://launchpad.net/~pygeode) to the list of software sources by following the commands listed below. Warning: this will allow Ubuntu to download third party software and must be done so at your own risk.

   Steps for adding PyGeode to the software sources list:
   - Open a terminal window
   - Type the following commands:

     sudo add-apt-repository ppa:pygeode/ppa
     sudo apt-get update

3. Open the Synaptic Package Manager found under System > Administration. Then, follow the instructions listed below:
   - Search for and install the package python-#.#-dev (where #.# should match the version of python installed). This package is required for compiling the C code accompanying TCRM.
   - Install python-scipy. This will also install python-numpy at the same time. Next, install python-matplotlib-dbg, python-wxgtk and python-matplotlib-basemap.
   - Install python-scientific package. This is required as the Ubuntu software repository has an older version of SciPy that does not support writing to NetCDF files.
   - Create a directory for TCRM and copy the file 'tcrm-v1.0.tar.gz' from the installation CD to this location. Then in a command terminal, change to the TCRM directory you just created and type 'tar –zxvf tcrm-v1.0.tar.gz' to extract the files.
   - Compile the C-code by running the command 'python Utilities/compile.py' from the TCRM root directory.

TESTING THE INSTALLATION

Before using TCRM, the unit test suite should be run to verify that TCRM is properly installed and functioning correctly. First ensure that the C code is compiled by following the instructions listed above. Then follow the instructions below to run the unit test suite:
- On Windows, this can be performed by running the file 'run_test_all.cmd', which can be found in the TCRM root directory.
- On Linux, this can be performed by opening a terminal window, changing directory to the TCRM root directory and running the command 'python ./unittests/test_all.py'

The test suite will take several minutes to run and will then display ‘OK’ on the terminal window if all components are working correctly. If a component fails, ‘FAILED’ will be displayed along with an error traceback for the failed test.

PREPARING A HAZARD CONFIGURATION FILE

Before performing a wind hazard simulation, you must first prepare a configuration file. This is a text file (with ‘.ini’ file extension) that contains all the settings required for the simulation. The first few lines from a configuration file are shown below to give you an idea of the format:

```
[Actions]
; TCRM modules to execute
DataProcess=True
ExecuteStat=True
ExecuteTrackGenerator=True
ExecuteWindfield=True
ExecuteHazard=True

[DataProcess]
InputFile=C:\tcrm\input\Allstorms.ibtracs_wmo.v03r02.csv
Source=IBTRACS

[Region]
; Domain for windfield and hazard calculation
gridLimit={'xMin':160.0,'xMax':170.0,'yMin':-30.0,'yMax':-20.0}
```

Rather than editing these files directly, a graphical user interface (the TCRM Configuration Editor) has been developed to simply the process of setting up a hazard simulation.

Running TCRM Configuration Editor

The configuration editor ‘configeditor.pyw’ is located in the TCRM root directory. On Windows, it can be launched by double clicking on this file using Windows Explorer. On Linux systems, it can be launched by opening a terminal window, changing directory to the TCRM directory and typing the command: ‘python configeditor.pyw’

Using the TCRM Configuration Editor

Once the configuration editor has been launched, you will be presented with a graphical user interface that will look similar to the screen capture shown in Figure 1.

The GUI window is divided into three sections. The top section allows the user to create or open a configuration file, the middle section allows the user to edit configuration settings and the bottom section provides user hints and allows any modified settings to be saved.

Before editing the configuration settings, you must either create a new configuration file by clicking on ‘Create New’ or open a pre-existing configuration file by clicking on ‘Browse’ in the top section of the GUI. Alternatively, ‘New’ and ‘Open’ in the File menu perform the same respective functions.

If this is your first time using TCRM, then you will probably want to create a new configuration file by pressing ‘Create New’. Then select ‘Save’ from the File menu and choose an appropriate name for your configuration file (i.e. ‘testhazard.ini’).

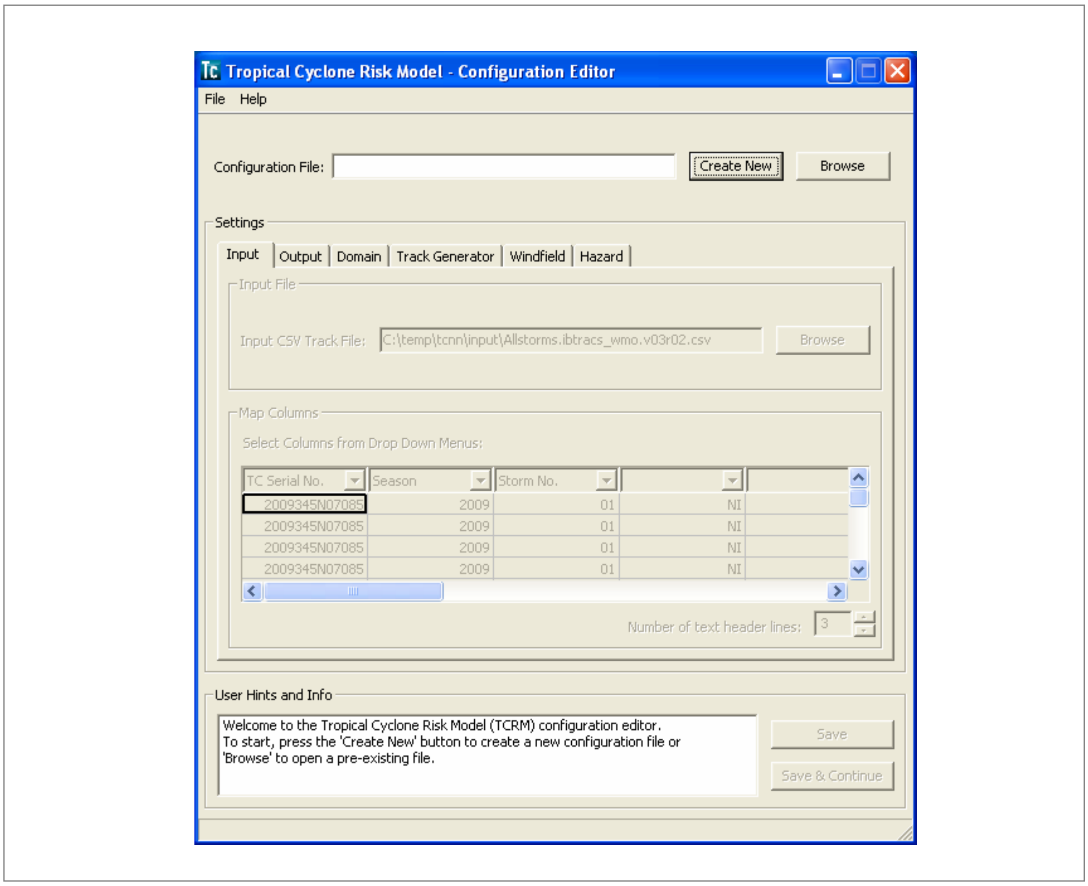
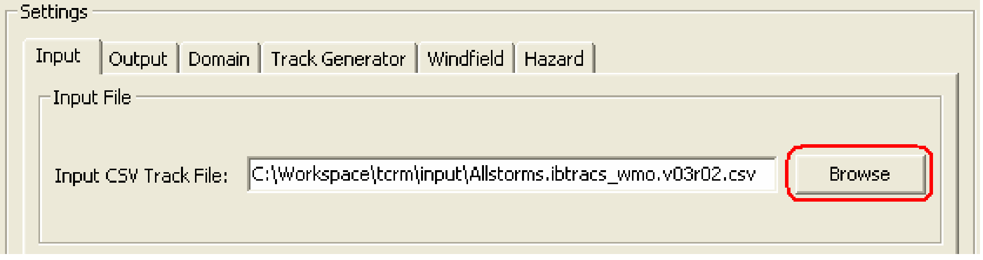


Figure 1: *TCRM Configuration Editor*

Input Settings

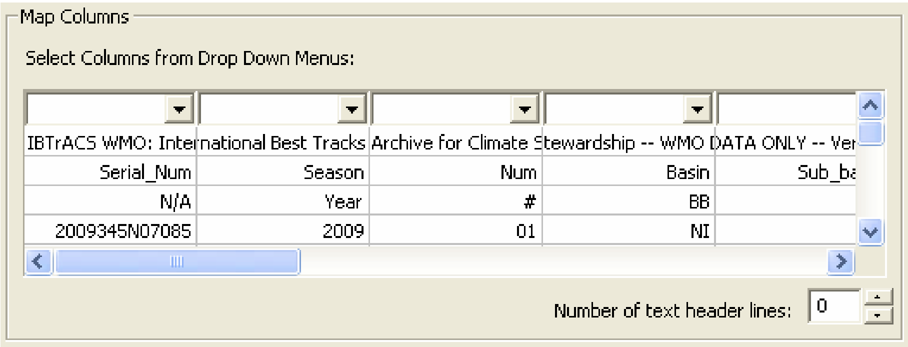
Once a pre-existing configuration file has been loaded or a new file has been created, the middle section of the GUI will become activated. This section allows for the configuration settings to be modified. The settings are organised into tabbed pages which can be navigated by clicking on the various tabs.

The first page displayed is the ‘Input’ page, which contains the settings for the cyclone track file to be used as input data by TCRM. To select a track file, click on the Browse button as shown in the figure below:



When the Browse button is pressed, a dialog window will be displayed allowing the user to select a track file. This must be a CSV (comma delimited) file containing columns that specify the time, longitude, latitude and central pressure values for all cyclone track points. A detailed explanation of the column requirements is provided further down.

TCRM should come pre-packaged with a recent version of the IBTrACS global database of tropical cyclones (www.ncdc.noaa.gov/oa/ibtracs/). IBTrACS is available in a range of data formats but the IBTrACS-WMO CSV version is utilised since the format is compatible with TCRM. Cyclones predating the 1981 cyclone season have been removed from the file, due to data homogeneity concerns. By clicking on 'Browse', a file selection box will open displaying the files in the 'input' folder under the TCRM root directory. Unless you wish to use your own input track file, select the IBTrACS file ('Allstorms.ibtracs_wmo.v03r02' or similar) and press enter. Sample data from this file will then automatically populated the columns under the 'Map Columns' heading as shown below:



A drop-down selection list will be found above each data column. This allows the user to specify what each data column represents. Only columns that are required by TCRM need column mappings; all the other columns can be left blank. Use the scroll bars to view the rest of the columns and sample data.

If you are using the IBTrACS dataset, then the following column settings are required:

Table 1: Column mapping required for IBTrACS input track file

COLUMN NUMBER	COLUMN MAP
1	TC Serial No.
2	Season
3	Storm No.
7	Date
9	Latitude
10	Longitude
12	Central Pressure

Some columns (Date, Central Pressure, RMW) require additional information and will result in pop-up dialogs being displayed. This will request the user to specify the date format or unit type. For the IBTrACS file, simply click okay to accept the default date format.

If an alternative track file is utilised, you must ensure that it meets the following column requirements:

1. Longitude, Latitude and Central Pressure columns must be defined

2. TCRM needs a date and time coordinate for each track point. This requirement can be met either with separate Year, Month, Day, Hour (and optional Minute) columns, or alternatively, with a customised date string column (i.e. '1997-12-30 22:30:00'). If a customised date string is to be used, then select the column name 'Date' and a pop-up window will appear asking for the required date format. The window also includes a help panel explaining how to define customised date formats.
3. TCRM needs to be able to combine individual track points into separate storm tracks. To achieve this, TCRM requires one of the following columns: Index, TC Serial No or Storm No. An Index column is simply a row of ones and zeros, where ones signify the start of a new storm. The TC Serial No is a unique identifier for each storm such as that utilised in the IBTrACS track file. Lastly, the Storm No. labels each storm with an integer counting up from one. If this storm number denotes the storm count for each season, then an additional Season column must also be defined.
4. Optionally, if the data set also contains reliable estimates of the Radius of Maximum Wind (RMW), then the column 'RMax' can also be defined. When this is not defined, TCRM reverts to a lognormal distribution to create RMW distributions.
5. Be careful to quality control the input track file. Non-physical values or incorrect signs can produce unexpected results and TCRM may terminate unexpectedly.

Once the columns have been mapped, edit the setting 'Number of text header lines' found below the sample data. If you are using the IBTrACS file, then this setting should be set to '3' since the file contains three header lines.

Once the Input page is complete, click on the 'Save & Continue' button in the bottom right corner of the GUI. This will save all modifications and navigate to the next configuration page.

Output Settings

The Output page allows the user to select which directory TCRM will use for outputting data and results. By default, the folder named 'output' under the TCRM root directory will be selected. If you wish to use a different directory, click the Browse button on this page to open a directory selection box. Once you are ready to continue, click 'Save & Continue'.

Domain Settings

The Domain page allows the user to select the location for estimating the wind hazard. TCRM defines the locations based on WMO weather station latitude/longitude coordinates. To choose a location, select the country from the first drop down menu and a weather station closest to your region of interest from the second drop down menu. The domain limits will then be automatically determined.

Track Generator Settings

This page allows the user to select the number of years' worth to simulate, where each simulation corresponds to one year. The higher the number of simulations selected, the more robust the wind hazard estimate will be since more samples will be available to fit the extreme value distributions. The improved robustness, however, comes at the expense of longer computational time required to run the simulations so a good compromise is to enter around 5000 simulations.

Hazard Settings

The Hazard page contains a list of return periods for estimating the wind hazard. The values must be a list of integers separated by commas. This page also includes a pull down selection box for specifying the speed units to be used for plotting the wind hazard. This is the last settings page; once you have finished editing the settings, click the Save button and exit the Configuration Editor. You are now ready to run your hazard simulation.

RUNNING TCRM

Windows Instructions:

1) Open Windows Explorer and go to the TCRM root directory.

2) Double click 'tcrm.cmd'. A window will open asking the user to enter a configuration filename.

3) Enter the name of the configuration file: i.e. "hazard.ini" and press enter. If the file is not located in the TCRM directory, then you will need to include the full file path: i.e. "C:\mydir\hazard.ini".

4) The message "---- Starting Hazard Simulation ----" will appear on the screen. The simulation should take several hours to run if default configuration settings were selected. Running TCRM at higher resolution or over a large domain can take considerably longer (i.e. several days or weeks to simulate).

5) The window will update periodically to notify you of the simulation progress. Do not close the window whilst TCRM is running as this will stop the simulation. However, you can minimise the window.

6) Once the simulation is complete, the message "TCRM has finished running" will appear on the screen.

Linux Instructions:

1) Open a terminal window and change directory to where TCRM is installed. To change directory, use the "cd" linux command. For example, if TCRM is located at "/home/username/tcrm" then type "cd /home/username/tcrm" to change to that directory.

2) Type the command "python main.py {configuration filename}" to run the hazard simulation. For example, you should type "python main.py hazard.ini" if the configuration file is "hazard.ini".

3) The simulation should take several hours to run if default configuration settings were selected. Running TCRM at higher resolution or over a large domain can take considerably longer (i.e. several days or weeks to simulate).

4) The terminal window will update periodically to notify the user of the simulation progress. Once the simulation is complete, the message "TCRM has finished running" will appear on the screen.

LOCATING THE OUTPUT FILES

All the output files created during the hazard simulation will be located in the output directory that was selected from the TCRM Configuration Editor. The files in this directory are organised in folders as described in Table 2.

Table 2: Output Folders and Contents

OUTPUT DIRECTORY FOLDERS	CONTENTS
Hazard	A NetCDF file named 'hazard.nc'. This file contains gridded maps of the cyclonic wind hazard for each return period as well as the shape parameters defining the Generalised Extreme Value Distributions at every grid point. It will also contain the 05^{th} and 95^{th} confidence intervals if they were calculated.
log	A text file containing a log of the simulation progress.
plots	All plots produced by TCRM. The files are organised into two subfolders 'hazard' and stats'.
plots \ hazard	Plots of the wind hazard results. This folder contains both spatial maps of the wind hazard for each return period as well as plots of the wind speed as a function of return period for each station found in the domain.
plots \ stats	Plots for statistical verification of the results.
process	Temporary files generated by TCRM.
tracks	Comma-delimited text files containing the storm tracks produced for each simulation. By default, each file will contain one years' worth of simulated events.
windfield	NetCDF files containing gridded maps of the maximum wind speed recorded at every grid point for each simulation.

VIEWING THE RESULTS

TCRM automatically generates plots of the wind hazard once a simulation has completed. These files can be found in the "plots\hazard" folder and are in 'PNG' format viewable by most image viewing software. The folder contains both spatial maps of the wind hazard for each return period as well as return period wind hazard curves for each weather station identified in the domain (See the plots shown in the next chapter for an example).

The rest of the output files generated by TCRM are stored in CSV (comma-delimeted) and NetCDF format as outlined in Table 2. The track files are stored in CSV format and can be viewed in either a spreadsheet program such as Microsoft Excel or a standard text editor. The windfield and hazard files are stored in NetCDF format. There are many options for reading and viewing NetCDF files, but a good free cross-platform viewer is Panoply that can be downloaded from the following website:

http://www.giss.nasa.gov/tools/panoply/

TROUBLE SHOOTING

If a simulation terminates unexpectedly before the hazard files were created, then an error has occurred. This can be caused by a range of reasons, such as: poor quality control of the input track file, insufficient disk space, missing Python packages, uncompiled C files or program bugs. To identity the cause of the error, you should start by reading the error traceback found in the log file (located in the 'log' folder of the simulation output directory). The log file can be viewed by opening it in any standard text file editor such as 'Notepad' in Windows or 'gedit' in Linux. If an error has occurred, then the error traceback will be listed at the end of the log file and will look similar to the highlighted section of this log file excerpt shown below:

```
Completed Data Processing
Running StatInterface
Traceback (most recent call last):
File "main.py", line 242, in <module>
main(configFile)
File "main.py", line 138, in main
from StatInterface import StatInterface
File "C:\tcrm\StatInterface\StatInterface.py", line 76, in <module>
import KDEOrigin
File "C:\tcrm\StatInterface\KDEOrigin.py", line 75, in <module>
import Utilities.KPDF as KPDF
ImportError: No module named KPDF
```

The last line displays the error event that caused TCRM to stop running; in this case the module named ‘KPDF’ could not be imported. This error was caused by not compiling the C files required by TCRM, which include “KPDF.c”. Sometimes it can be difficult to identify the root cause of an error. If this is the case, then follow the steps below:

1) Run the unit test suite by following the instructions given near the start of this chapter. This will identify any TCRM files that are not operating correctly.
2) If no errors were identified, run the example hazard simulation following the instructions provided in the next chapter.
3) If the test suite and example hazard simulation both ran correctly, then it is likely that the error was caused by a specific problem with your configuration settings or the input track file chosen. If a different track file was used, then try running the simulation again with the IBTrACS file found in the TCRM input folder. If this runs successfully, then the error was probably caused by a problem with your input file so check the values and formatting for mistakes.
4) More detailed log file information can also help identify the cause of the error. Open your configuration file in a text file editor and look for the “Loglevel” setting under the “[Logging]” heading. Change this setting from “Loglevel=INFO” to “Loglevel=DEBUG”, then save the file and then rerun the simulation. If the cause of the error is still not clear from the log file, then send an email explaining the problem encountered along with the log and configuration files as attachments to the contacts listed on the http://code.google.com/p/tcrm/ home page and we will endeavour to help you resolve the problem.
5) [Advanced Users]: If you have programming experience, then you can use the python debugger module ‘pdb’ to track down the cause of the error. Refer to http://docs.python.org/library/pdb.html for more information.

Example Hazard Run

ASSESSING THE CYCLONIC WIND HAZARD FOR PORT HEDLAND, WESTERN AUSTRALIA

This chapter provides a demonstration of how TCRM can be applied to assess the cyclonic wind hazard for a given location. The location chosen for the assessment is Port Hedland, a town situated in the Pilbara region of northern Western Australia. The town experiences an average of approximately one tropical cyclone every two years, with the cyclone season extending from mid December to April and peaking in February. The most significant cyclone that affected this town in recent decades was TC Joan which occurred in December 1975. This storm caused damage to 85% of the town's houses, destroyed the local hospital and produced a maximum recorded wind gust of 208 km/h (http://www.bom.gov.au/wa/cyclone/about/pthed/index.shtml).

Figure 2: *Map showing location of Port Hedland.*

RUNNING THE EXAMPLE HAZARD SIMULATION

Windows Instructions:

1) Open Windows Explorer and go to the TCRM directory.

2) Double click on 'tcrm.cmd'. A window will open asking the user to enter a configuration filename.

3) Type ".\example\port_hedland.ini" followed by the enter key (See Fig #).
This provides TCRM with the name of the configuration file as well as the relative path, indicating that the file 'port_hedland.ini' is located in the 'example' folder under the TCRM directory.

4) A message will appear on the screen "---- Starting Hazard Simulation ----". The simulation should take between 1 and 2 hours depending on the speed of the computer. The screen will update periodically to notify you of the simulation progress.

5) Remember: do not close the window whilst the simulation is in progress. However, you can minimise the window.

6) Once the simulation is complete, the message "TCRM has finished running" will appear on the screen. The next section will describe how to view the results.

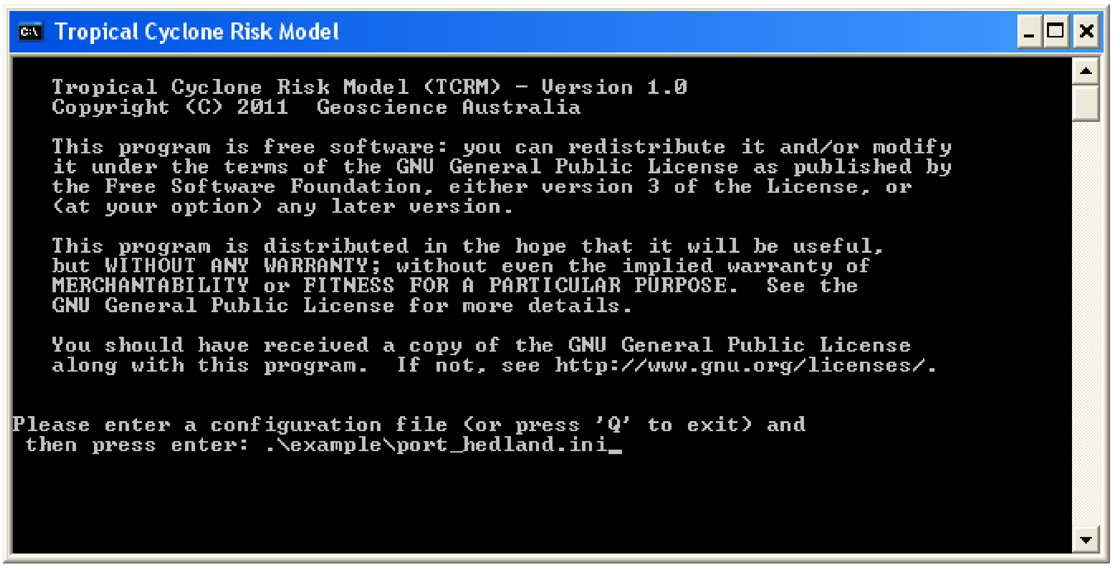


Figure 3: *Entering the configuration file for the Port Hedland simulation.*

Linux Instructions:

1) Open a terminal window and change directory to where TCRM is installed. To change directory, use the “cd” linux command. For example, if TCRM is located at “/home/username/tcrm” then type “cd /home/username/tcrm” to change to that directory.

2) Type “python main.py ./example/port_hedland.ini” to start the simulation.

3) The message will appear in the terminal “---- Starting Hazard Simulation ----”. The simulation should take between 1 and 2 hours depending on the speed of the computer. The screen will update periodically to notify you of the simulation progress.

4) Once the simulation is complete, the message “TCRM has finished running” will appear on the screen. The next section will describe how to view the results.

VIEWING THE SIMULATION RESULTS

Locating the output files:
To locate the output files, open a file browser and go to the TCRM directory. Then open the “output” folder and inside this folder will be another folder named “port_hedland” that contains all the files generated by TCRM during the simulation.

Viewing the hazard results:
Plots of the wind hazard can be found under the ‘plots\hazard’ folder. This folder contains both spatial maps of the wind hazard for each return period as well as return period wind hazard curves for each weather station identified in the domain (See Figure 4 & 5 for examples).

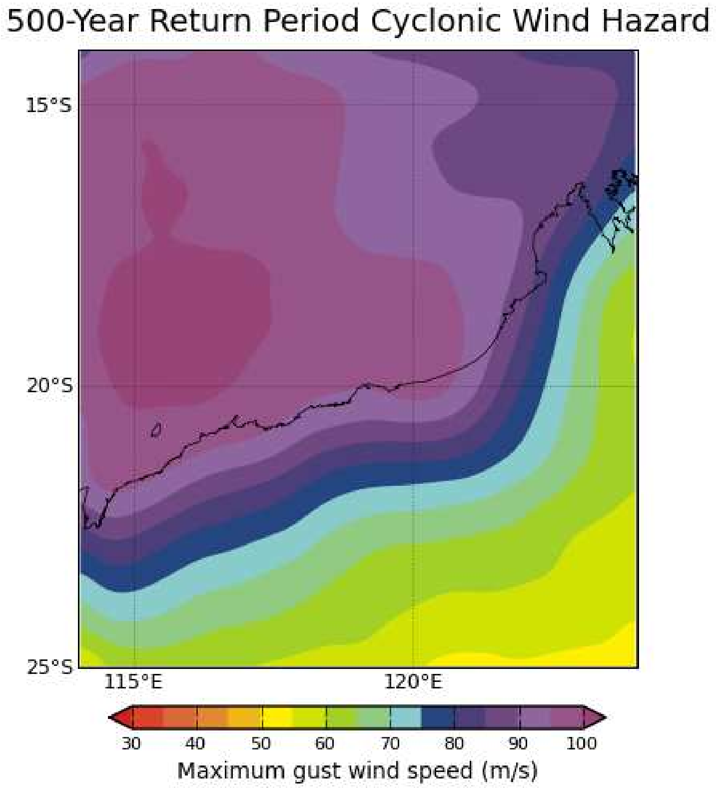


Figure 4: *500-year return period cyclonic wind hazard for north-west Australia*

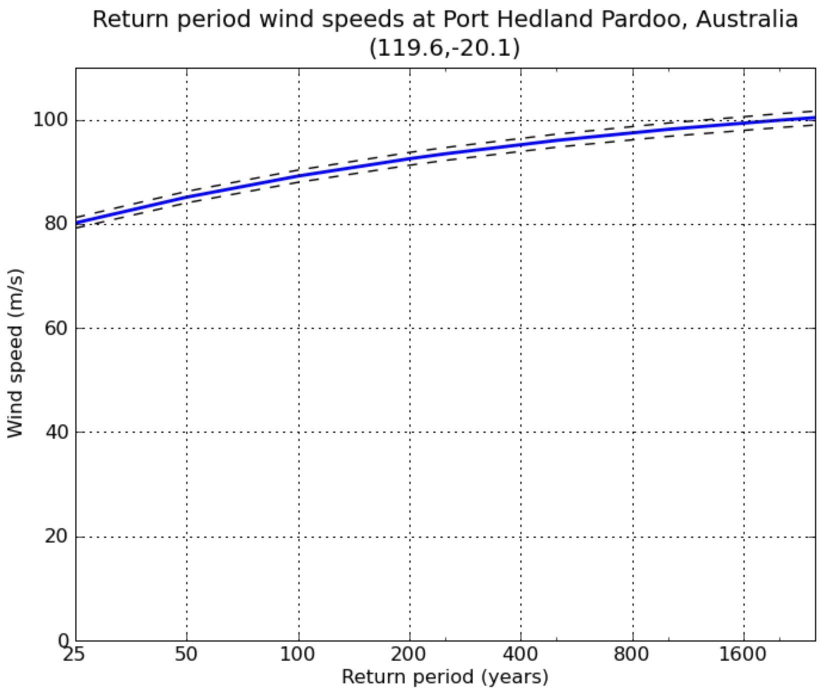


Figure 5: *Cyclonic wind hazard for Port Hedland, WA, Australia. The dotted lines indicate the 5th and 95th confidence intervals.*

Description of TCRM

SOFTWARE ARCHITECTURE

TCRM is modular code (See Fig. 1) written mostly in Python, with a small amount of C code used to increase computational speed. Since it is modular, TCRM can be flexibly applied to a wide range of applications. These include: creating regional wind hazard maps, estimating damage from an approaching storm and providing input wind fields for a storm surge model for example. The model can also be used to assess potential changes to cyclonic wind hazard with climate change by using tracks extracted from global climate models as input data. TCRM provides a range of adjustable settings and also offers a selection of wind profile and boundary layer models. This allows the user to adjust the wind fields to better represent the storm characteristics of a particular region.

TCRM can be run on either Windows or Linux operating systems. It can also be run with no software cost to the user since both TCRM and Python are open-source software.

TCRM is comprised of five modules (See Figure 6) that are run sequentially such that the output from one module forms the input for the next. See below for a brief description of each of these modules.

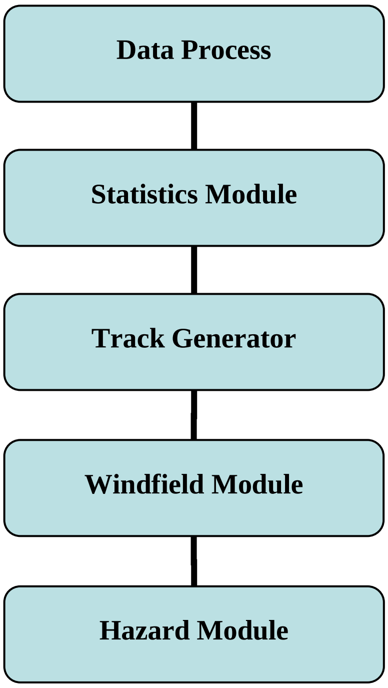


Fig. 6: TCRM Modules

DATA PROCESS MODULE

This module processes the input track file, calculating basic track properties such as bearing and speed, and then outputs this information as a series of comma-delimited text files ready for input into the statistics module.

STATISTICS MODULE

The statistics module fits probability distributions to the track properties of the input track record (i.e. speed, bearing, pressure and storm genesis) for each location by applying kernel density estimation.

TRACK GENERATOR MODULE

This module generates synthetic tracks that have similar statistical properties to the input track record using a method based on Hall & Jewson (2007). The tracks are generated via a Monte-Carlo process whereby the track properties are obtained by sampling from the statical distributions produced with the Statistics Module. The algorithm also makes use of the auto-correlation to account for track memory.

WINDFIELD MODULE

To estimate the swath of destructive winds around each of the synthetic cyclones, TCRM applies a parametric wind field comprised of a radial wind profile and a boundary layer model. Using parametric wind fields permits a large number of synthetic events to be modelled using modest computational resources.

Radial wind profiles

The wind field around each tropical cyclone is calculated at high spatial resolution (up to 0.01°) to ensure the peak wind speeds near the eye is accurately captured. TCRM first uses a radial profile to estimate the gradient level wind associated with the circulation. To allow users to explore the range of uncertainty associated with choosing different radial profiles, we have implemented a number of profiles in TCRM including:

- Holland (1980);
- Schloemer (1954);
- Willoughby and Rahn (2004);
- Powell et al (2005);
- Jelesnianski (1966);
- McConochie et al (2004); and
- A Rankine vortex profile.

The Willoughby, Schloemer and Powell et al profiles are all variants of the Holland profile – the difference being the definition of the β parameter.

Boundary layer model

In addition to the range of radial profiles, users can also select one of three boundary layer models. These boundary layer models relate the winds at the gradient level to those near the surface, taking into account the asymmetry induced by the forward motion of the tropical cyclone and the surface friction effects. Most commonly in parametric tropical cyclone models, this is achieved by vector addition of the forward motion and the gradient winds together with a surface wind reduction factor. Examples of this type include McConochie et al's (2004) model, which varies the inflow angle as a function of radial distance, or the Hubbert model (1991). Alternatively, a linear analytic model (Kepert, 2001) can be used, which provides a more realistic representation of the boundary layer flow with minimal computational cost.

HAZARD MODULE

The hazard module estimates the maximum gust wind speed for a given return period. This is achieved by fitting a Generalised Extreme Value Distribution (GEVD) to the aggregated set of maximum gust wind speeds from all simulations for a given grid point. The method of L-moments is used for fitting the GEVDs (Hosking, 1980).

References

Holland, G. J., 1980: An Analytic Model of the Wind and Pressure Profiles in Hurricanes. *Monthly Weather Review*, **108,** 1212-1218.

Hubbert, G. D., G. J. Holland, L. M. Leslie, and M. J. Manton, 1991: A Real-Time System for Forecasting Tropical Cyclone Storm Surges. *Weather and Forecasting*, **6,** 86-97.

Jelesnianski, C. P., 1966: Numerical Computations of Storm Surges without Bottom Stress. *Monthly Weather Review,* **94,** 379-394.

Kepert, J. D., 2001: The Dynamics of Boundary Layer Jets within the Tropical Cyclone Core. Part I: Linear Theory. *Journal of Atmospheric Sciences*, **58,** 2469-2484.

McConochie, J. D., T. A. Hardy, and L. B. Mason, 2004: Modelling tropical cyclone over-water wind and pressure fields. *Ocean Engineering*, **31,** 1757-1782.

Powell, M., and Coauthors, 2005: State of Florida hurricane loss projection model: Atmospheric science component. *Journal of Wind Engineering and Industrial Aerodynamics*, **93,** 651-674.

Schloemer, R. W., 1954: Analysis and synthesis of hurricane wind patterns over Lake Okeechobee NOAA Hydrometeorology Report 31.

Willoughby, H. E., and M. E. Rahn, 2004: Parametric Representation of the Primary Hurricane Vortex. Part I: Observations and Evaluation of the Holland (1980) Model. *Monthly Weather Review*, **132,** 3033-3048.

Appendix

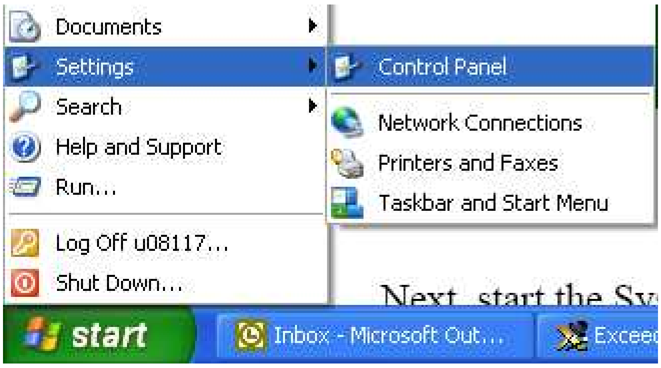
GLOSSARY

ACRONYM	FULL NAME	DEFINITION
GEVD	Generalised Extreme Value Distribution	A probability density function used to describe extreme value events. The function is a generalisation of the Gumbel, Fréchet and Weibull extreme value distributions and can be defined by three parameters: location, scale and shape.
HURDAT	North Atlantic hurricane database	A database of tropical cyclones in the Atlantic Ocean, Gulf of Mexico and Caribbean Sea.
IBTrACS	International Best Track Archive for Climate Stewardship	A global best-track tropical cyclone database, collected from agencies in every ocean basin.
MSLP	Mean Sea Level Pressure	The atmospheric pressure at mean sea level. If over land, the pressure is reduced via a formula to the mean sea level height.
NetCDF	Network Common Data Form	NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
RMW	Radius of Maximum Wind	Distance between the centre of the cyclone and its band of strongest winds.

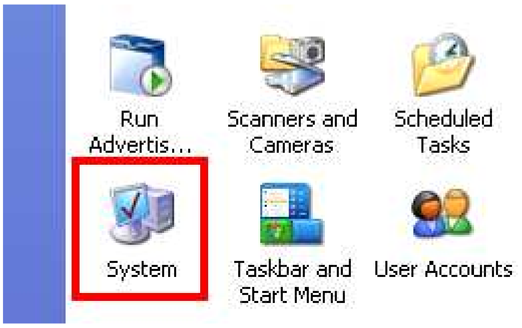
SETTING WINDOWS PATH ENVIRONMENT VARIABLES

The following steps demonstrate how to set the PATH environment variable for Windows XP systems. The procedure for Windows Vista and 7 should be similar.

First, open the Control Panel:

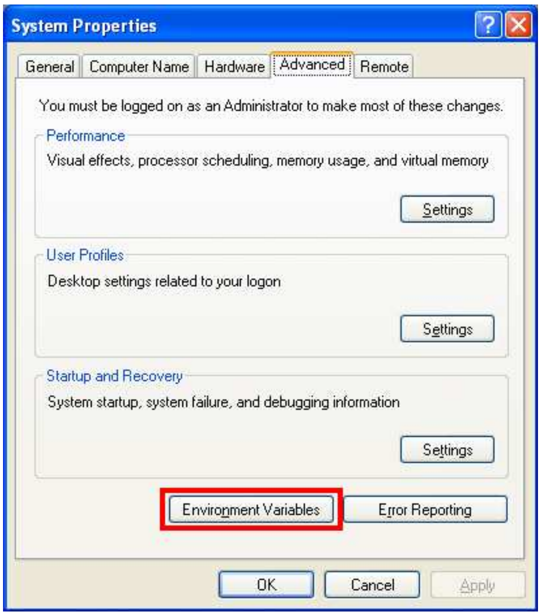


Next, start the System applet:

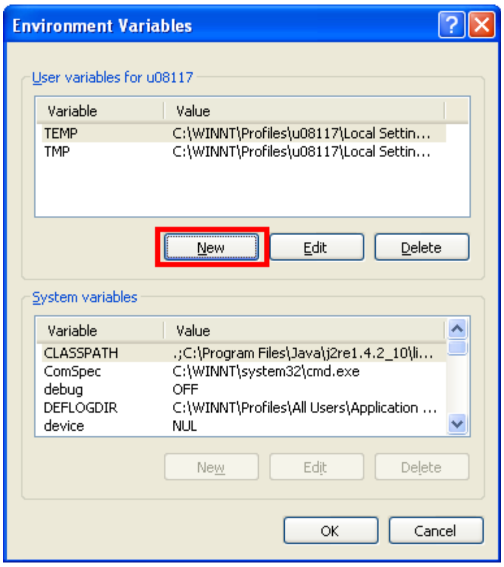


Select the Advanced tab in the System Properties window:

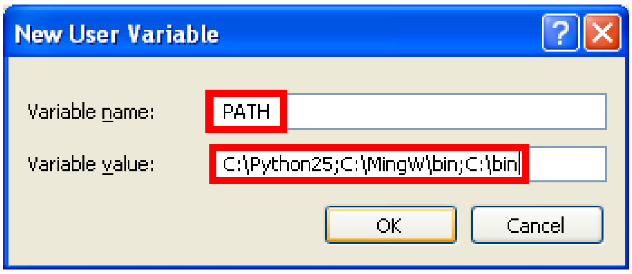
Press the Environment Variables button in the Advanced tab:



If the PATH variable is not defined in the User variables or System variables windows, press the New button in either of the two windows (for a personal machine, choose the System variables window). If PATH already exists in the User variable or System variables window, select the row with the PATH variable name in the appropriate window and press the Edit button next to the New button in that window:



You will be shown the editor window whichever button you pressed in the above step. If the Variable name box is empty, type in the name PATH. In the Variable value box type the value you want the PATH variable to have. If there is already some text in the box, place your additional value at the front of the existing value, not forgetting to terminate your additional string with the ';' character. The final value string must be a series of directory names separated by ';' characters:



When you are finished, press the OK button and exit from the applet.