CLIMADA MANUAL

Version Jan 2014

Uncertainty and risk of climate change: from probabilistic damage calculation to the economics of climate adaptation – shaping climate resilient development.

David Bresch, *david.bresch@gmail.com*
Lea Mueller, *muellele@gmail.com*

Instead of an Introduction

1.1. Preamble

Climate adaptation is an urgent priority for the custodians of national and local economies, such as finance ministers and mayors. Such decision-makers ask:

- What is the potential climate-related damage to our economies and societies over the coming decades?
- How much of that damage can we avert, with what measures?
- What investment will be required to fund those measures – and will the benefits of that investment outweigh the costs?

The economics of climate adaptation methodology as implemented in climada provides decision-makers with a fact base to answering these questions in a systematic way. It enables them to understand the impact of climate on their economies – and identify actions to minimize that impact at the lowest cost to society. Hence it allows decision-makers to integrate adaptation with economic development and sustainable growth. In essence, we provide a methodology to pro-actively manage total climate risk. Using state-of-the-art probabilistic modeling, we estimate the expected economic damage as a measure of risk today, the incremental increase from economic growth and the further incremental increase due to climate change. We then build a portfolio of adaptation measures, assessing the damage aversion potential and cost-benefit ratio for each measure. The adaptation cost curve illustrates that a balanced portfolio of prevention, intervention and insurance measures allows to pro-actively manage total climate risk.

1.2. A visual primer

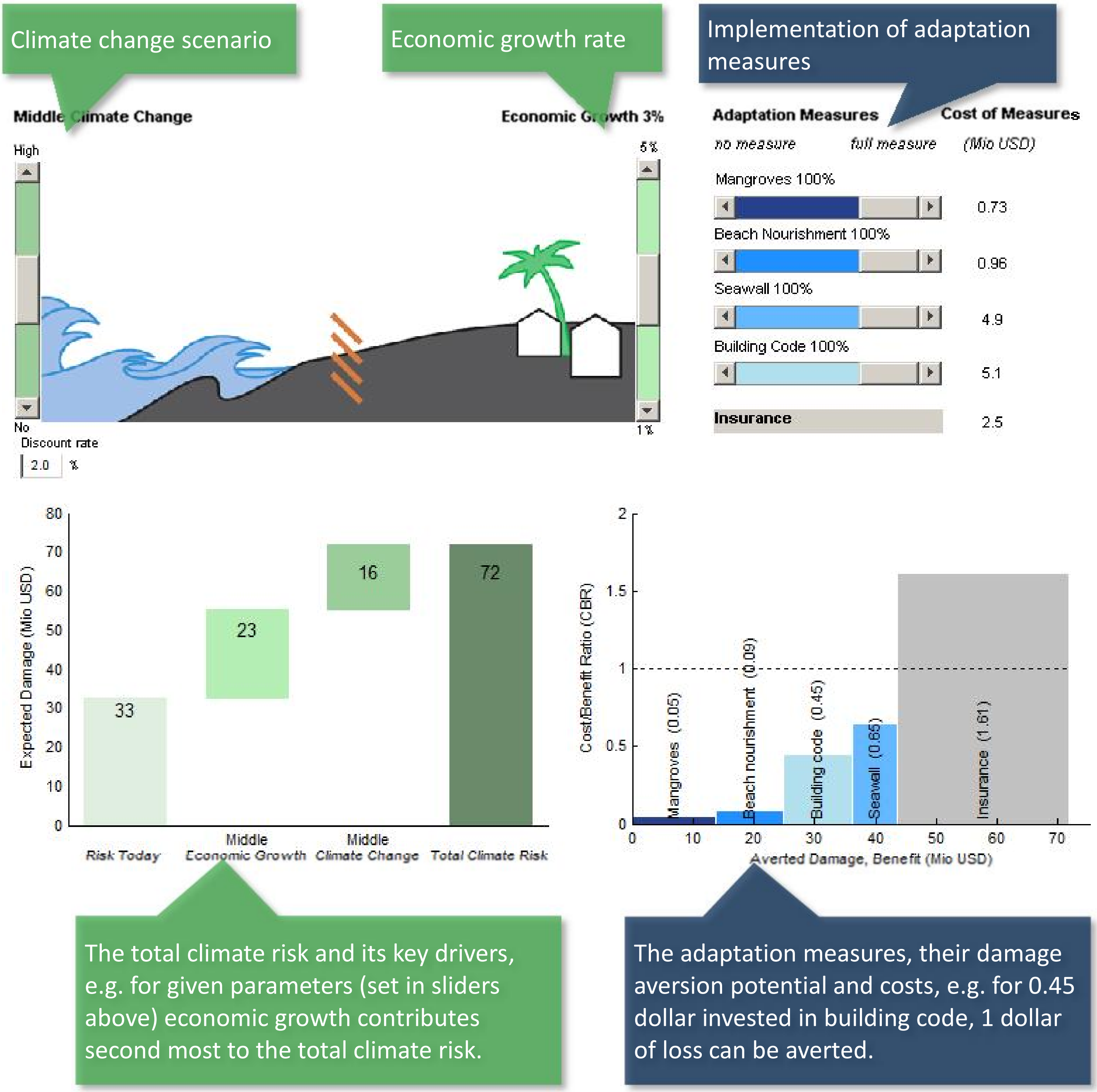


Figure 1: The demonstration code `climada_demo`[1] *implements the concept of total climate risk and cost-effective adaptation in an interactive way: The user can experiment with all relevant factors (sliders, top) and instantly observe the effect – both on risk (measured by expected damage, graph on the left) and the basket of adaptation measures (shown as adaptation cost curve, graph on the right).*

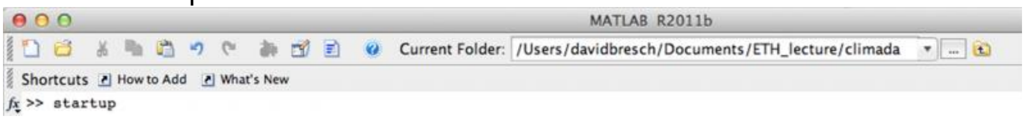
[1] See just a few lines below about how to get this (and climada) started.

1.3. A few introductory remarks

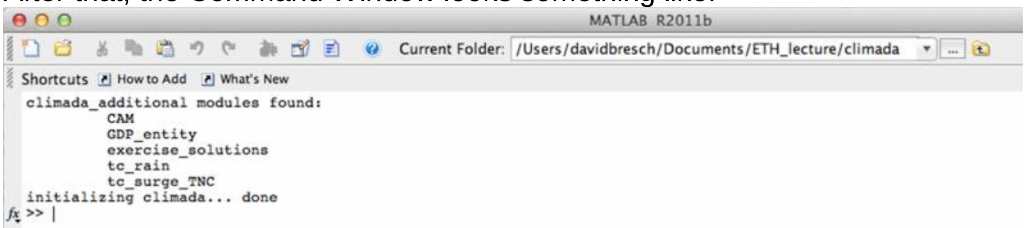
Get started

- Set the MATLAB Current Folder to climada[2] (use the button to browse), e.g.:

- Enter `startup` in the MATLAB Command Window:



  and press Enter (or Return). This initializes climada, sets some variables (e.g. the location of the data folder[3]) and detects any climada_additional modules[4]. After that, the Command Window looks something like:



  It's ok if there are no further modules shown, as long as … `done` appears.
- Start by just invoking the climada demonstration by entering `climada_demo` in the MATLAB Command Window[5], which is also the best way to test whether climada works properly – you should see something as shown above as a visual primer (Figure 1) and be able to play with the sliders.

If unfamiliar with the basic concepts of damage calculation, climate impact and adaptation measures assessment, please refer to the slides as handed out in the lecture[6] - or play at least a bit with `climada_demo` ;-)

To cut the whole story short, climada produces an adaptation cost curve, as shown in the lower right part of the visual primer (and many more nice things). According to the modules hazard, assets, damage functions and adaptation measures, the following steps are required in order to come up with a climate adaptation cost curve

1. Generate a <u>hazard</u> event set
   a. Generate the events for today's climate
   b. Store intensities at centroids

2. Import a list of <u>assets</u> and with corresponding <u>damage functions</u>[7] (the so-called entity)

---

[2] Usually the folder you extracted climada.zip to
[3] The global variable `climada_global` (a struct) contains all these variables. See the code climada_init_vars.m which sets all these variables. Make sure you never issue a `clear all` command, as this would also delete `climada_global` and hence climada would not find it's stuff any more.
[4] A climada_additional module extends the functionality of climada and allows users to further develop climada without risking to change the core code. See further below for some examples of modules.
[5] From now on, just type any command in `Courier` in the MATAB Command Window, as we will not state this each time again.
[6] www.iac.ethz.ch/education/master/climate_risk

a. Read the list of today's assets
b. Encode to centroids
c. Repeat above steps for future assets (e.g. 2030)

3. Import the list of adaptation measures
   a. Read the data from Excel

4. Calculate the damages and benefits of measures
   a. Calculate the damages for the list of today's assets, today's hazard event set and the list of measures
   b. Repeat the previous step for future assets but still today's hazard and the list of measures
   c. Finally, repeat the first step (a.) again now for future assets, the climate change scenarios and the list of measures. Note that for this step, you need to create the hazard event set for the climate change scenarios (e.g. 2030)

5. Display the results – e.g. in the form of an adaptation cost curve.

[7] Sometimes also referred to as ‚vulnerability curves' of just ‚vulnerabilities'. See lecture material for proper definitions.

Contents

Part 1: Wind from Tropical Cyclones

Generate a Tropical Cyclone (TC) Hazard Event Set

Here, we describe how to generate the hazard event set for today's climate.

2. Download the Raw Data

From http://weather.unisys.com/hurricane

North Atlantic (atl)	http://weather.unisys.com/hurricane/atlantic/tracks.atl
East Pacific (epa)	http://weather.unisys.com/hurricane/e_pacific/tracks.epa
West Pacific (bwp)	http://weather.unisys.com/hurricane/w_pacific/tracks.bwp
South Indian ocean (bsh)	http://weather.unisys.com/hurricane/s_indian/tracks.bsh
North Indian ocean (bio)	http://weather.unisys.com/hurricane/n_indian/tracks.bio

Upon download, it is worth adding the extension .txt, so the files can be read with any editor (they are ASCII), such that the North Atlantic file becomes e.g. tracks.atl.txt

In order of testing routines and understanding the methodology in a faster way, a shortened version of the North Atlantic dataset, `TEST_tracks.atl.txt` is often used, referred to as *TEST atl* data (just comprising the last few years of data). You find this dataset at ...\climada\data\tc_tracks\TEST_tracks.atl.txt

Please consult Appendix B for details of the raw data file structure.

3. Read the Raw Data, Append Season and Category and Check Data Manually

3.1. Read the Raw Data

Select the test dataset `TEST_tracks.atl.txt` the first time you use this routine. In case you read southern hemisphere data make sure that the dataset filename contains 'she' ('she' for southern hemisphere) for a correct format conversion of the latitude with the code `convert_latitude`.

stdout for *TEST atl* data:

reading raw data from C:\Documents and Settings\All Users\Documents\catMos\climada\data\tc_tracks\TEST_tracks.atl.txt ...
WARNING: line 565 empty
writing binary file C:\Documents and Settings\All Users\Documents\catMos\climada\data\tc_tracks\TEST_tracks.atl_bin.mat
filtering raw data (4926 nodes) ...
WARNING: 440 nodes w/o pressure, 440 nodes w/o wind, 440 without neither (but no records deleted)
Processing 142 tracks ...

142 tracks read, 142 tracks chosen, 0 tracks not chosen

MATLAB call:

tc_track = **climada_tc_read_unisys_database** (unisys_file);

1 x 448 storms, per storm:

.Timestep
.lon
.lat
.MatSustainedWind
.CentralPressure vector
.yyyy (1 x No. of nodes)
.mm
.dd
.hh
.nodetime_mat
.ID_no
.orig_event_flat
.extratrop array
.name

stdout for atl data (the full historical track data):

```
reading raw data from C:\Documents and Settings\All Users\Documents\catMos\climada\data\tc_tracks\tracks.atl.txt ...
WARNING: line 2477 empty
WARNING: line 7598 empty
WARNING: line 12719 empty
writing binary file C:\Documents and Settings\All Users\Documents\catMos\climada\data\tc_tracks\tracks.atl_bin.mat
filtering raw data (43390 nodes) ...
WARNING: 29417 nodes w/o pressure, 3174 nodes w/o wind, 3174 without neither (but no records deleted)
Processing 1410 tracks ...
WARNING: 1851 ID 1185102, less than 3 nodes, skipped
WARNING: 1851 ID 1185103, less than 3 nodes, skipped
WARNING: 1853 ID 1185301, less than 3 nodes, skipped
WARNING: 1853 ID 1185302, less than 3 nodes, skipped
WARNING: 1853 ID 1185305, less than 3 nodes, skipped
WARNING: 1853 ID 1185307, less than 3 nodes, skipped
WARNING: 1854 ID 1185402, less than 3 nodes, skipped
WARNING: 1855 ID 1185501, less than 3 nodes, skipped
WARNING: 1855 ID 1185503, less than 3 nodes, skipped
WARNING: 1856 ID 1185604, less than 3 nodes, skipped
WARNING: 1858 ID 1185801, less than 3 nodes, skipped
WARNING: 1858 ID 1185802, less than 3 nodes, skipped
WARNING: 1859 ID 1185901, less than 3 nodes, skipped
WARNING: 1860 ID 1186003, less than 3 nodes, skipped
WARNING: 1861 ID 1186104, less than 3 nodes, skipped
WARNING: 1861 ID 1186107, less than 3 nodes, skipped
WARNING: 1862 ID 1186204, less than 3 nodes, skipped
WARNING: 1864 ID 1186402, less than 3 nodes, skipped
WARNING: 1865 ID 1186501, less than 3 nodes, skipped
WARNING: 1865 ID 1186502, less than 3 nodes, skipped
WARNING: 1865 ID 1186505, less than 3 nodes, skipped
WARNING: 1865 ID 1186506, less than 3 nodes, skipped
WARNING: 1866 ID 1186604, less than 3 nodes, skipped
WARNING: 1867 ID 1186703, less than 3 nodes, skipped
WARNING: 1867 ID 1186705, less than 3 nodes, skipped
WARNING: 1867 ID 1186708, less than 3 nodes, skipped
WARNING: 1869 ID 1186908, less than 3 nodes, skipped
WARNING: 1869 ID 1186909, less than 3 nodes, skipped
WARNING: 1870 ID 1187001, less than 3 nodes, skipped
WARNING: 1870 ID 1187007, less than 3 nodes, skipped
WARNING: 1870 ID 1187010, less than 3 nodes, skipped
1410 tracks read, 1379 tracks chosen, 31 tracks not chosen
```

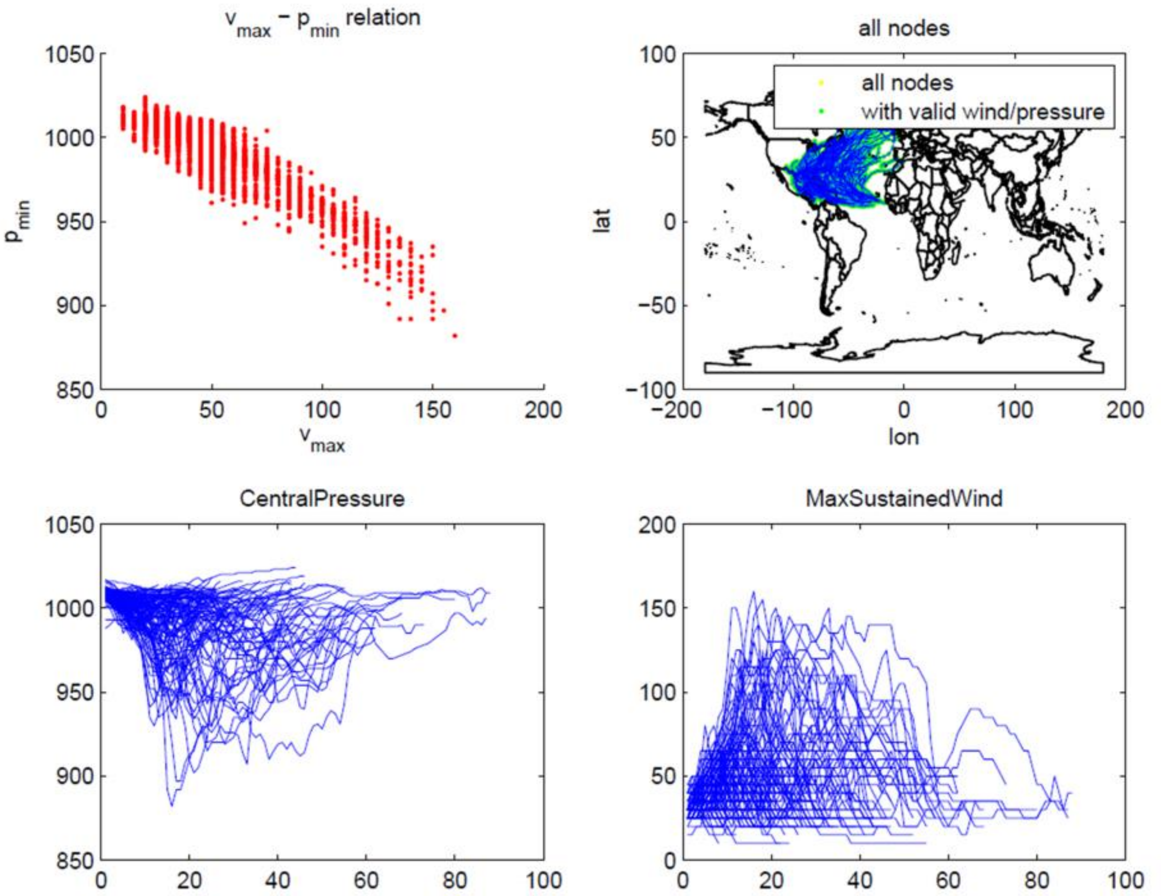


Figure 2: Graphical check plot for climada_tc_read_unisys_database for the TEST atl *data.*

3.2. Add Season and Storm Category

MATLAB call:

tc_track = **climada_add_tc_track_season** (tc_track);

tc_track = **climada_add_tc_track_stormcategory** (tc_track);

1 x 448 storms, per storm:

.season
.category
} array

North Atlantic cyclone season lasts from 1 June to 31 May. In the South Indian Ocean cyclone season starts only 1 July and ends 30 June.

Add storm category according to Saffir-Simpson Hurricane Scale to every track (Table 1).

Table 1 : Saffir-Simpson Hurricane Scale.

Category	Maximum Sustained Wind Speed*				Storm Surge
	knots	mph	km/h	m/s	m
Tropical Depression	< 34	< 39	< 63	< 18	≈ 0
Tropical Storm	< 64	< 73	< 118	< 33	0,1−1,1
Hurrican Category 1	< 83	< 95	< 153	< 43	1,2−1,6
Hurrican Category 2	< 96	<110	< 177	< 49	1,7−2,5
Hurrican Category 3	< 113	<130	< 209	< 58	2,6−3,8
Hurrican Category 4	< 135	<155	< 249	< 69	3,9−5,5
Hurrican Category 5	> 135	> 155	> 250	> 70	> 5,5

* 1-minute period, at 10 m (33 ft) above the surface
Source:
http://en.wikipedia.org/wiki/Saffir%E2%80%93Simpson_Hurricane_Scale

3.3. Check Data

Recalculate `nodetime_mat` and `timestep` between each node records.
Check dates of measurement of each tc track (e.g. correct year after turn of the year)
If dates of all nodes checked and cleaned, interpolate records to 6 hours interval
Add categories and seasons to tc tracks

Recalculate nodetime_mat

```
for track_i = 1:length(tc_track)
        tc_track(track_i).nodetime_mat = datenum(tc_track(track_i).yyyy,...
                                tc_track(track_i).mm,...
                                tc_track(track_i).dd,...
                                tc_track(track_i).hh, 0,0);
end
```

Recalculate timestep

```
for track_i = 1:length(tc_track)
        timestep       = diff(tc_track(track_i).nodetime_mat)*24;
        timestep(end+1) = timestep(end);
        tc_track(track_i).TimeStep = timestep;
end
```

4. Technical Details

4.1. Create Figure

Create figure with specific height and width as percentage of screen height.
Output is the handle of the figure. Default values for width are 0.7 and height 0.8.

MATLAB call:
fig = **climada_figuresize** (height, width)

climada_figuresize (0.8, 0.8)

4.2. Technical Details: Plot World Borders and Shade Mozambique and Madagascar

Plots world borders from ASCI file. One or multiple countries can be shaded in light yellow if requested with `check_country` .Default axis limits are Longitude from -200 to 200 and Latitude from -100 to 100 (European map view).

MATLAB call:
h = **climada_plot_world_borders** (linewidth, check_country, map_border_file, keep_boundary, country_color)

climada_plot_world_borders (0.8, 'United States (USA)')
climada_plot_world_borders (0.8, {'Canada' 'Germany'})

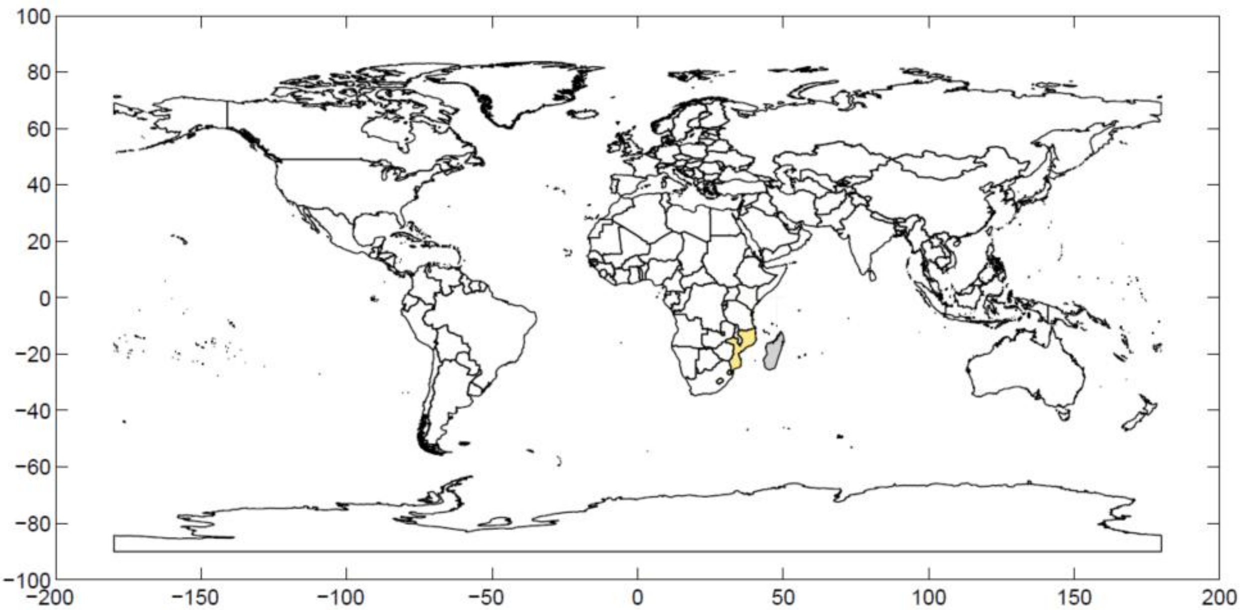


Figure 3: World borders from ASCI file and shaded countries Mozambique(yellow) and Madagascar (grey).

5. Visual analysis of historical tracks

5.1. Plot TC track with colour coding according to Saffir-Simpson Hurricane Scale

Plot single tc track with color coding, if `check_legend` = 1 display legend for Saffir-Simpson Hurricane Scale if requested in north-eastern corner, change markersize, default 8. Returns handle of all points and lines. h: vector with length of nodes.

MATLAB call:

h = **climada_plot_tc_track_stormcategory** (tc_track, markersize, check_legend)

climada_plot_tc_track_stormcategory (0, [], 1)
used to plot only legend without tc track (tc_track = 0).

5.2. Plot all TC tracks during one specific season (year)

Choose one season (year) for all plots during this period to be displayed. Figure can be saved in

...\results\tc_tracks_2000.pdf

MATLAB call:

climada_plot_tc_track_season
(tc_track, season, markersize, check_printplot, invisible)

climada_plot_tc_track_season

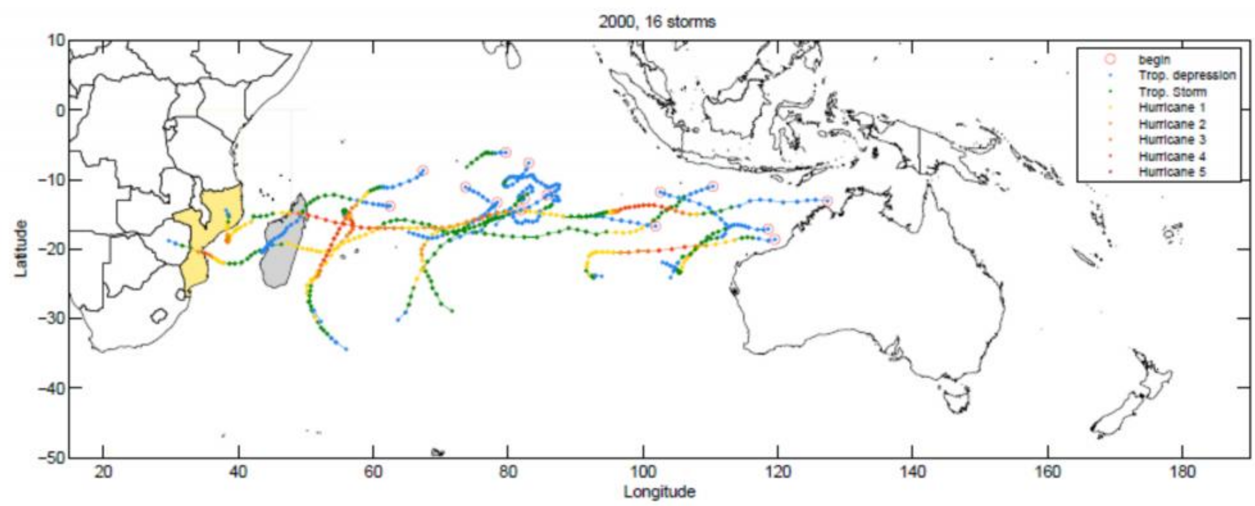


Figure 4: TC tracks recorded in the year 2000 displayed with colours according to the Saffir-Simpson Hurricane Scale

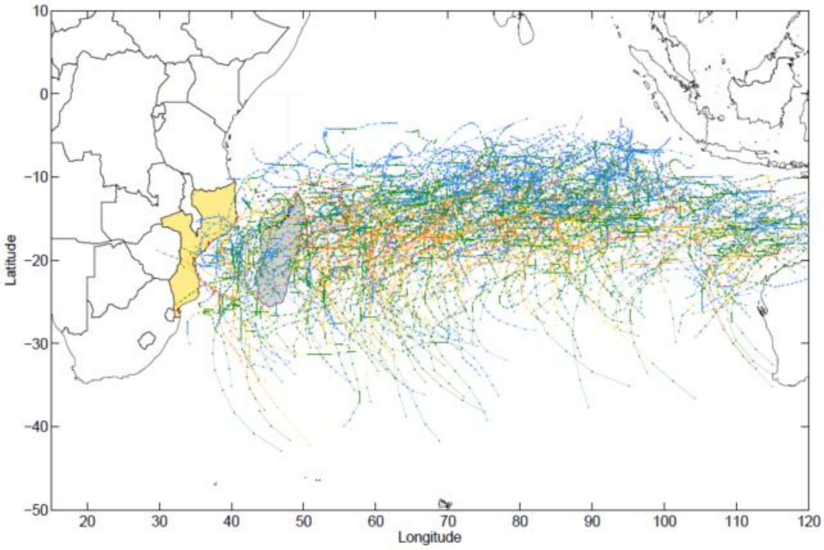


Figure 5: All historical storms in the south westerns Indian Ocean from 1978 to 2011.

6. Analyse Distribution of Initial Wind Speed and Change in Wind Speed (for Random Walk Method to Generate Probabilistic Wind Speeds)

Plots distribution of initial wind speed and change in wind speed from node to node based on historical tracks. The function fits a normal distribution to both and returns mu and sigma of the normal distribution either in knots or in m/s. Figure is always displayed in m/s, can be saved in results folder.

MATLAB call:

[mu, sigma, A] = **climada_distribution_v0_vi** (tc_name, output_unit, check_visible, check_printplot)

climada_distribution_v0_vi (tc_track, 'm/s', 1, 1)
climada_distribution_v0_vi (tc_track, 'kn' , 1, 1)

Function needs specific functions:

- subaxis
- parseArgs
- mygaussfit

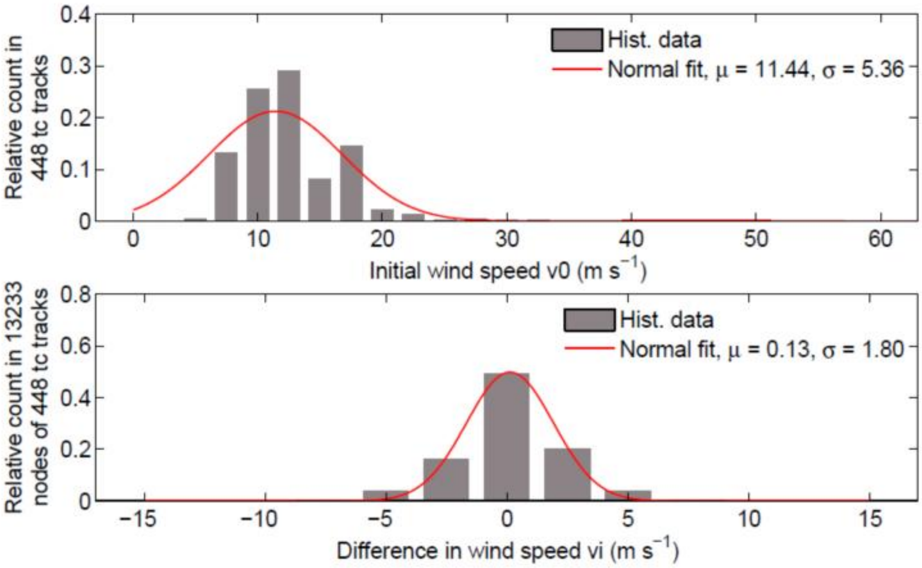


Figure 6: Distribution of initial wind speed v0 (top) and difference in wind speed vi (bottom) for the tracks recorded in the south western Indian Ocean during 1978 and 2011.

7. Generate the Probabilistic Set

In order to generate the probabilistic set, each historical (original) track is varied, using a directed random walk process (wind speed, longitude, latitude)

Probabilistic tc tracks saved in ...\data\tc_tracks\tc_track_save.mat

MATLAB call:

tc_track_prob = **climada_tc_random_walk_position_windspeed** (tc_track, tc_track_save, ens_size, ens_amp, Maxangle, check_plot, check_printplot);

climada_tc_random_walk_position_windspeed (tc_track, 'tc_track_prob', 9, 0.35, pi/7);

Please check the header of climada_tc_random_walk_position_windspeed.m for more parameters, or simply type

help climada_tc_random_walk_position_windspeed

in the MATLAB command window.

stdout for atl data (the full historical track data):

season to every tc track added
mu and sigma in KNOTS
adding 9 derived tracks to 1379 original storms
generating 13790 derived storms took 5.922000 sec (0.658000 sec/track
saving probabilistic tc track set as tc_track_XXXX.mat

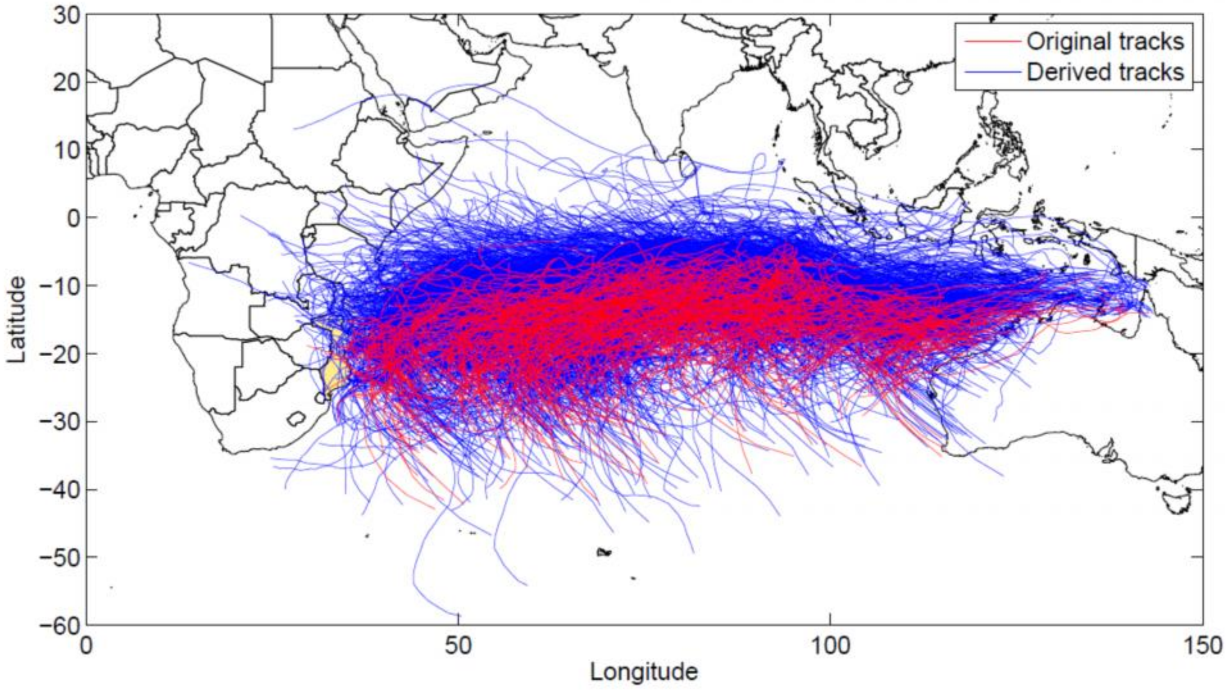


Figure 7: Graphical check for climada_tc_random_walk_position_windspeed for TEST atl with default parameters.

Methods

a. Wind speed:

1. Derive mu and sigma of initial wind speed and change in wind speed (see chapter 6)

2. Generate probabilistic change in wind speed for every node
   Vi = mu(2) + sigma(2) .* randn(ens_size,nodes_count);

3. Add up probabilistic changes in wind speed
   vi_cum = cumsum(vi(ens_i,:));

4. Sum up initial wind speed and following changes in wind speed
   V = tc_track(track_i).MaxSustainedWind + vi_cum;

5. Round to 5 kn
   v = round(v /5)*5;

b. Longitude, Latitude

1. Generate random starting points for ensemble members
   x0 = ens_amp0 *(rand(ens_size,1)-0.5);
   y0 = ens_amp0 *(rand(ens_size,1)-0.5);

2. Directed random walk (summed in dimension 2), how much differ the derived from the original
   X = cumsum(ens_amp ***sin**(cumsum(2*Maxangle*rand(ens_size,nodes_count) - Maxangle, 2)),2);

   y = cumsum(ens_amp ***sin**(cumsum(2*Maxangle*rand(ens_size,nodes_count) - Maxangle, 2)),2);

3. Add change in coordinates to the different starting points
   x_cum = x(ens_i,:)- x(ens_i,1) + x0(ens_i);
   y_cum = y(ens_i,:)- y(ens_i,1) + y0(ens_i);

4. Fill in the derived track: add dlon/dlat
   tc_track_prob(track_counter).lon = tc_track(track_i).lon + x_cum;
   tc_track_prob(track_counter).lat = tc_track(track_i).lat + y_cum;

8. Analyse Probabilistic Data Set

8.1. Accumulated Cyclone Energy ACE

The function plots histograms of accumulated cyclone energy ACE, No. of storms per seasons, No. of hurricanes and No. of major hurricanes per season of probabilistic tracks, with historical tracks indicated with dotted black lines

MATLAB call:

climada_plot_ACE (tc_track, name_tag, check_printplot)

climada_plot_ACE (tc_track_prob, '4480', 1)
climada_plot_ACE

Figure can be saved with check_printplot = 1 and specification of name_tag in
...\results\histogram_tracks_prob_'name_tag'.pdf

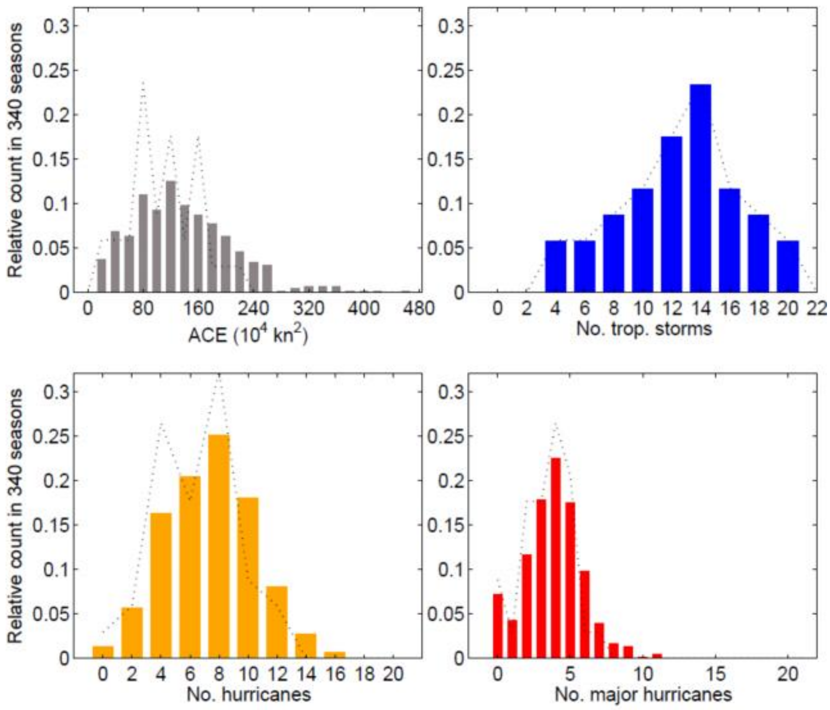


Figure 8: Histograms of available cyclone energy (ACE) (top left), number of tropical storms per season (top right), number of hurricanes per season (bottom left) and number of major hurricanes per season (bottom right). The histograms are based on 4480 probabilistic tracks. The dotted lines indicate the histograms based on the historical tracks.

8.2. Histogram of Position and Change in Position

The function plots histograms of start point of longitude and latitude and difference in longitude and latitude of probabilistic tracks, with historical tracks indicated with dotted black lines

MATLAB call:

climada_distribution_lon_lat (tc_track, check_printplot, check_printplot_2)

climada_distribution_lon_lat ([], 1, 1)

Needs function

- hist2d

Figure can be saved with check_printplot = 1 in
...\results\tc_tracks_histogram_tracks_prob_lon_lat.pdf

and with check_printplot2 = 1 in
...\results\tc_tracks_map_starting_points.pdf

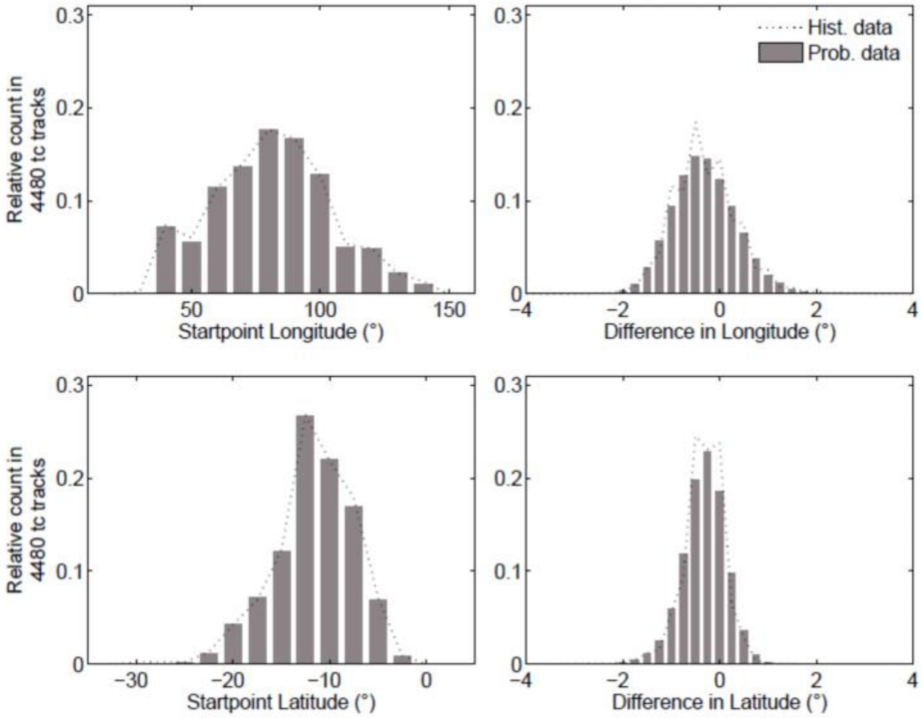


Figure 9: Histograms of starting points of longitude (top left) and latitude (bottom left) and difference in longitude (top right) and latitude (bottom right). The histograms are based on 4480 probabilistic tracks. The dotted lines indicate the histograms based on the historical tracks.

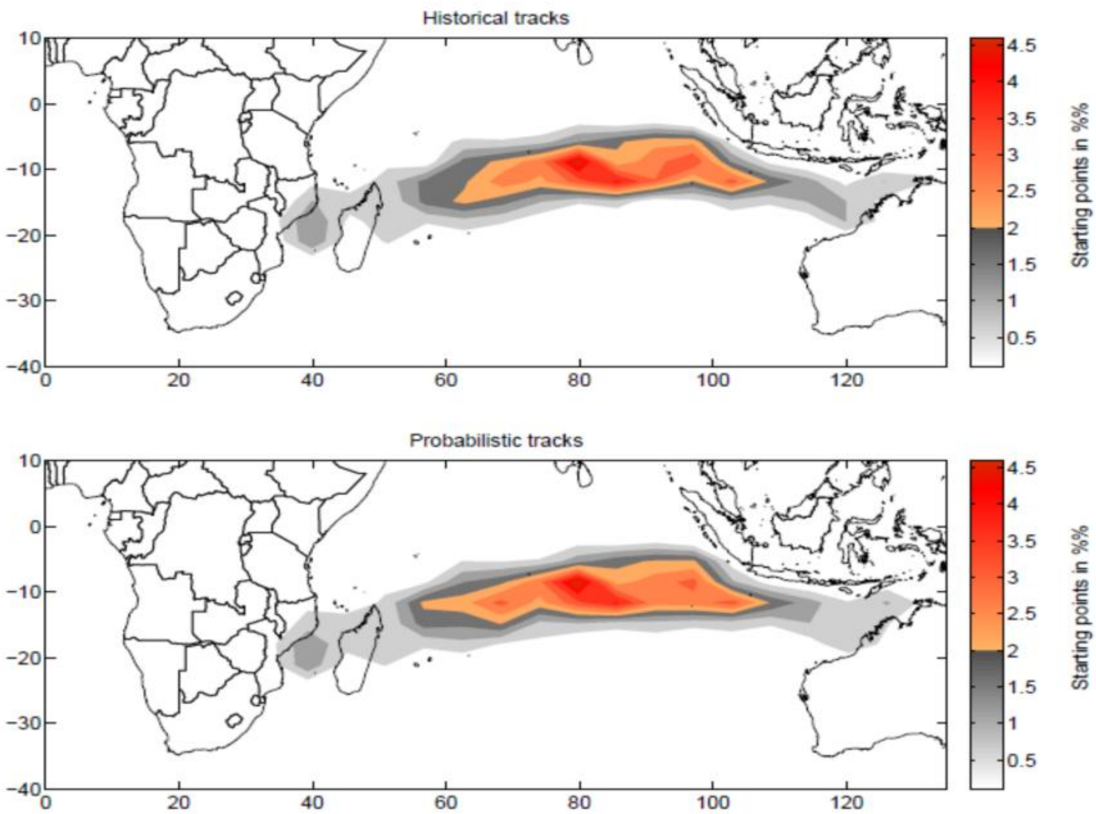


Figure 10: Starting points of probabilistic (bottom) and historical tracks (top).

8.3. Visual Analysis of Probabilistic Data Set

Plot historical tc track (Longitude, Latitude) in world map according to Saffir-Simpson Hurricane Scale. Add plot of probabilistic generated sister storms. Historical track has black lines around markers to identify as original track.

Prompt to press p for print, enter to go to next historical track or choose your own historical track by inserting the requested track number or press x to exit.

MATLAB call:

climada_plot_probabilistic_wind_speed_map (tc_track, track_req)

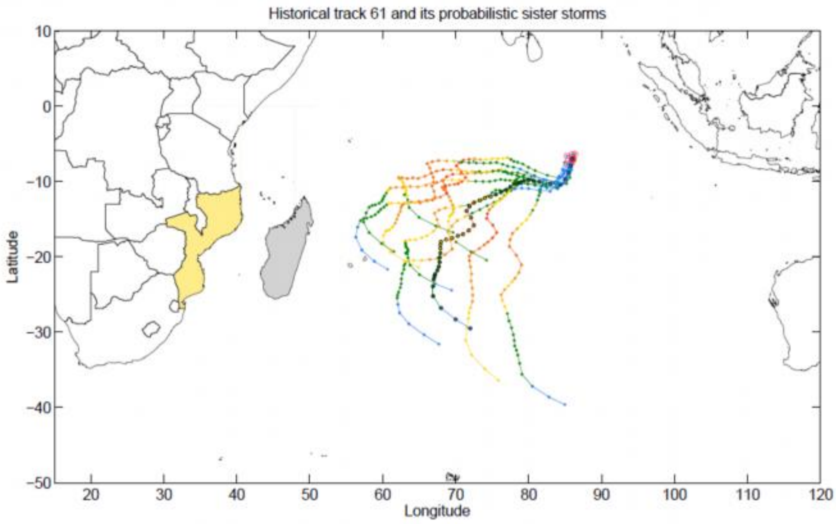
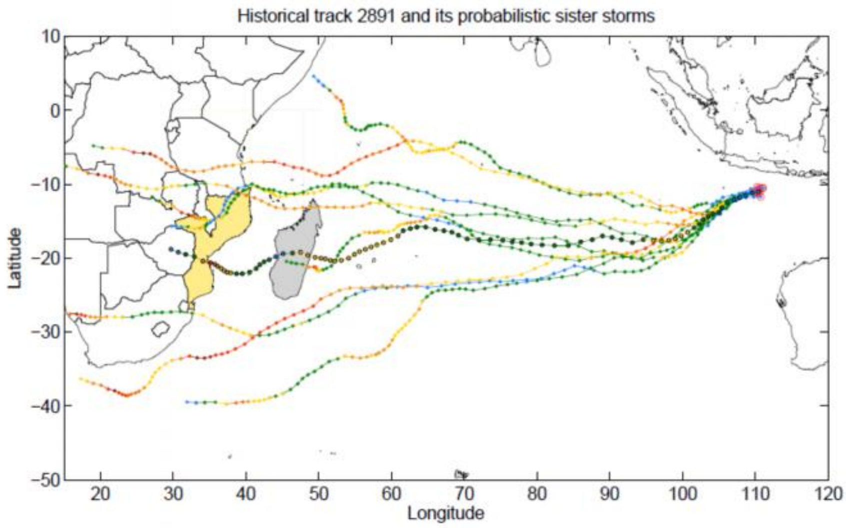


Figure 11: Visual check for historical track 61 and its 9 generated probabilistic storms.

Figures can be saved by pressing p in

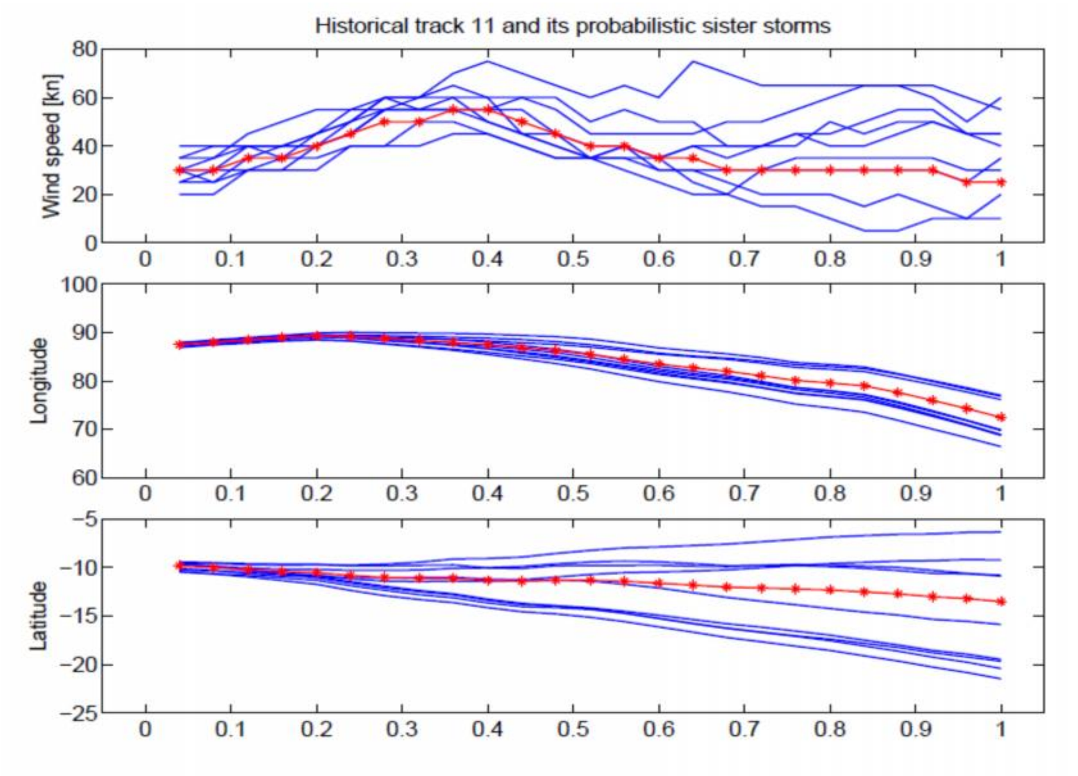
...\results\tc_track_61_prob.pdf



Plot historical tc track (wind speed, longitude, latitude) and add wind speed, longitude and latitude of probabilistic generated sister storms. Historical tracks are red, probabilistic blue.

Prompt to press p for print, enter to go to next historical track or choose your own historical track by inserting the requested track number or press x to exit.

MATLAB call:
climada_plot_probabilistic_wind_speed_lon_lat (tc_track, track_req)

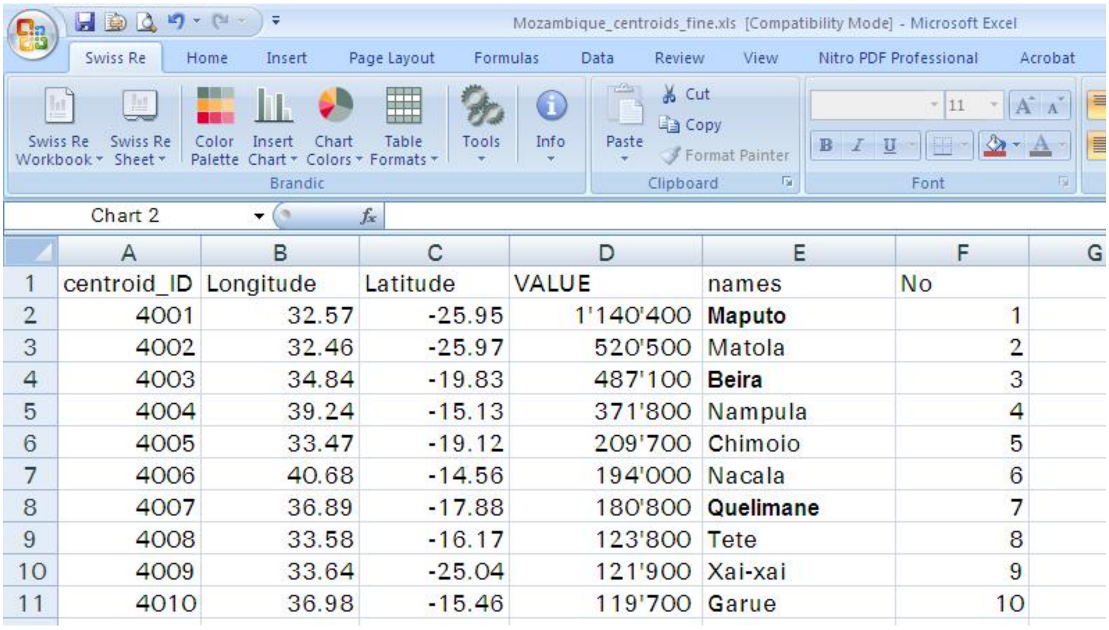


Figures can be saved by pressing p in
...\results\tc_track_11_v_lon_lat.pdf

9. Read, Create and Save Centroids

Read centroids from excel file, creates uniformly spaced grid with
grid_resolution = 1 °
grid_extent = 5°.

Append excel file with names and No (optional)



	A	B	C	D	E	F	G
1	centroid_ID	Longitude	Latitude	VALUE	names	No	
2	4001	32.57	-25.95	1'140'400	**Maputo**	1	
3	4002	32.46	-25.97	520'500	Matola	2	
4	4003	34.84	-19.83	487'100	**Beira**	3	
5	4004	39.24	-15.13	371'800	Nampula	4	
6	4005	33.47	-19.12	209'700	Chimoio	5	
7	4006	40.68	-14.56	194'000	Nacala	6	
8	4007	36.89	-17.88	180'800	**Quelimane**	7	
9	4008	33.58	-16.17	123'800	Tete	8	
10	4009	33.64	-25.04	121'900	Xai-xai	9	
11	4010	36.98	-15.46	119'700	Garue	10	

MATLAB call:

centroids = **climada_centroids_read** (centroids_filename, centroids_save, visualize, markersize)

climada_centroids_read ([], [], 1, 8);

1x 1 struct

.excel_file_name	array
.centroid_ID	
.Longitude	
.Latitude	vector
.VALUE	(No. of centroids x 1)
.names	
. city_ID	only for Mozambique

Centroids saved with centroids_save name or prompted in
...\climada\data\system.centroids_countryname.mat

Code:

```
for lon_i=minlon-grid_extent:grid_resolution:maxlon+grid_extent*3
   for lat_i=minlat-grid_extent:grid_resolution:maxlat+grid_extent
           ...
      end
end
```

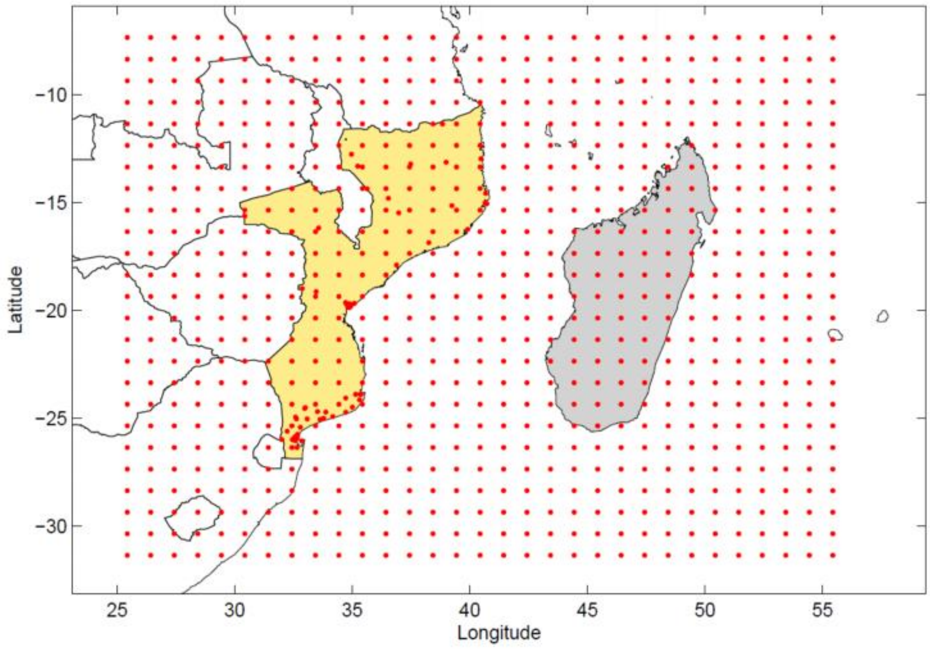


Figure 12: Extract of world map with centroids displayed in red dots.

10. Generate the Wind Footprint(s)

Generate wind field resulting from single track of tropical cyclone. The function converts tc_track.MaxSustainedWind in knots to res.gust in m/s.

Normally wind footprint calculation is tested on a single tc track prior to generation of the hazard event set of all the entire historical and probabilistic track set. The tc_windfield calculations are speeded up by only calculating for centroids within 750 km distance of min,max track lon/lat and by not assigning res.node_lon(centroid_i), res.node_lat(centroid_i).

Uses function

- climada_gridded_VALUE

MATLAB call:

res = **climada_tc_windfield**
(tc_track, centroids, equal_timestep, silent_mode, check_plot)

climada_tc_windfield (tc_track, centroids, 1,1,1);
climada_tc_windfield;

1x 1 struct

.gust (wind speed in m/s at all centroids)
.node_Azimuth
.node_lat
.node_lon
.ID
.lat
.lon

vector
(1 x No. of centroids)

Method:

Currently, the code implements the Holland windfield[8]. Given that the distance of the centroid (D) to the eye of the storm is smaller than its corresponding radius (R), the wind speed (S) is given by:

$$S = \begin{cases} \min\left(M, \quad M + 2\cdot T\cdot\dfrac{D}{R}\right) & D \le R \quad \textit{in the inner core} \\ \max\left(0, \quad \left((M - abs(T))\cdot\dfrac{R^{1.5}}{D^{1.5}}\cdot e^{1-\frac{R^{1.5}}{D^{1.5}}}\right) + T\right) & D < 10\cdot R \quad \textit{in the outer core} \\ 0 & D > R \quad \textit{out of radius} \end{cases}$$

[8] Holland, G. J., 1980: An analytic model of the wind and pressure profiles in hurricanes. Monthly Weather Review, 108, 1212-1218.

where M denotes the maximum sustained wind and T is the celerity (forward speed). In case where D is still ten times smaller than R, you find yourself in the outer core of the storm where the wind speed takes the form of the second line in the equation above. If none of these cases are true, the wind speed is set to zero.

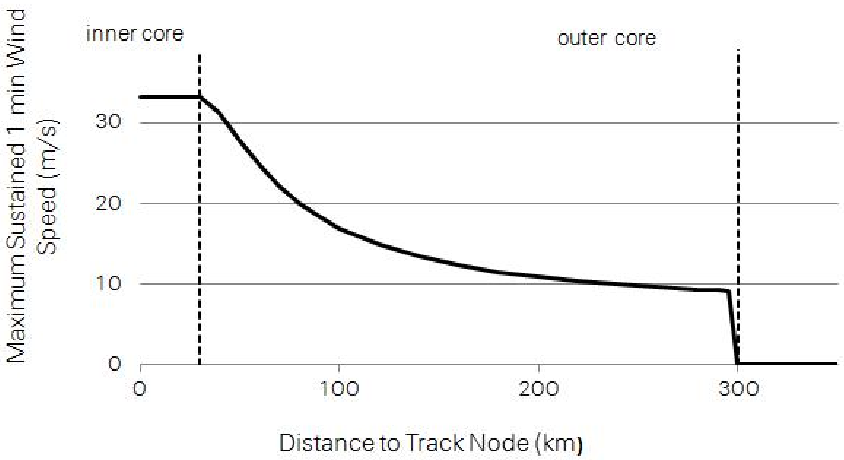


Figure 1: Maximum sustained 1 min wind speed in relation to the distance to the track node.

The radius of maximum wind (R, in km) depends on the latitude of the track node (L) as follows:

$$R = \begin{cases} 30 & L \leq 24° \\ 30 + 2.5 \cdot abs(L) - 24 & L > 24° \\ 75 & L > 42° \end{cases}$$

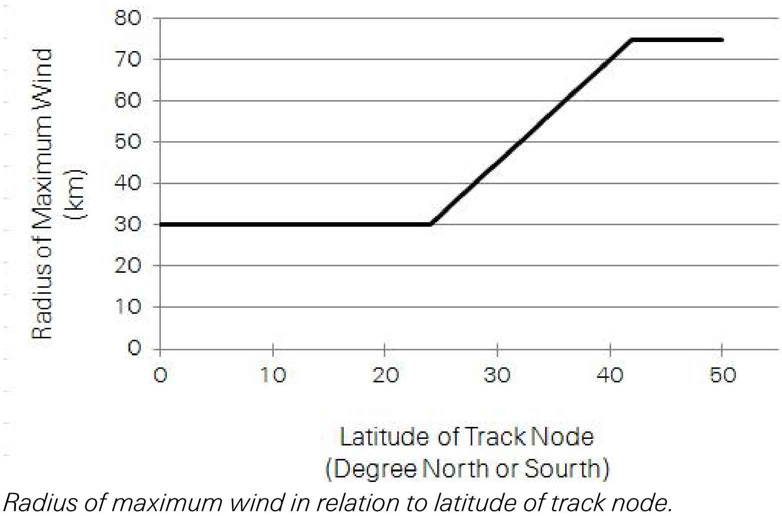


Figure 2: Radius of maximum wind in relation to latitude of track node.

Finally the wind speed (S) describes the maximum sustained 1 min wind speed. To get wind gusts that a few seconds (3-5 s) wind peaks are typically around 27% higher than a 1 min sustained wind in a hurricane environment.
http://www.prh.noaa.gov/cphc/pages/FAQ/Winds_and_Energy.php

Any other windfield parametrization can be implemented in a similar fashion (just implement in a copy of climada_tc_windfield , e.g. climada_tc_windfield2, see also the routine climada_tc_hazard_set to change the caller when generating the probabilistic set).

In order to test the wind field calculation, the following might help:
Use the tc_track structure (should still be in memory), but start with only one track, e.g. tc_track(84) for the 84th track[i]. Investigate tc_track.name to find a particular event. Use e.g. the following code to show a list of track number, year and name:

```
for i=1:length(tc_track)
   fprintf('%i %i %s\n',i,tc_track(i).yyyy(1),...
  char(tc_track(i).name));
end
```

Load the centroids using
centroids = climada_centroids_read('',1),
note that this call also plots the centroids (use the zoom function on the map).

See also the parameter check_plot in the PARAMETER section of the code or refer to the routine climada_color_plot.

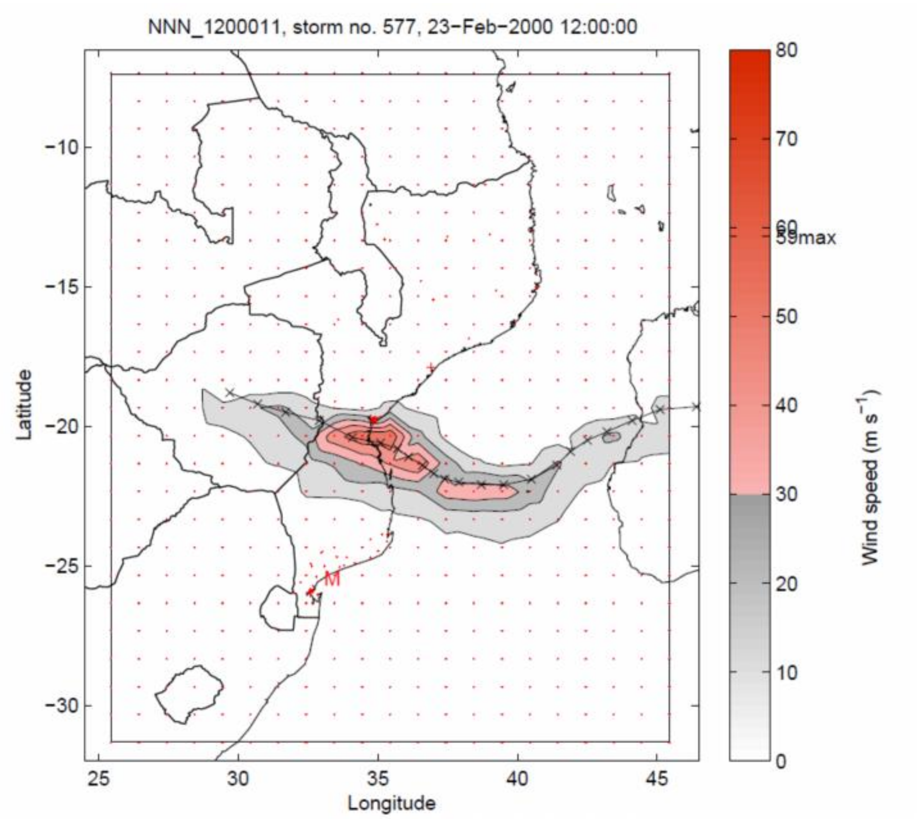


Figure 13: Wind field calculate based on track 577. The track resulting in the second highest wind speed in the city of Beira, Mozambique.

Figure saved in
...\results\footprint_NNN_1199506.pdf

11. Calculate the Wind Fields for a Single Track and Display as Animation

Refines tc track to 1hour resolution, calculates wind field for every time step of 1h. The function displays the wind fields for selected aggregated time steps, e.g. 3h, 6h, 24h. Aggregation default is 6h.

MATLAB call:

climada_tc_windfield_animation
(tc_track, centroids, aggregation, check_avi)

climada_tc_windfield (tc_track(34), centroids, 12);
climada_tc_windfield;

Uses function

- climada_gridded_VALUE
- climada_tc_windfield_timestep

Movie saved in
...\results\windfield_animation_trackname_24h.avi

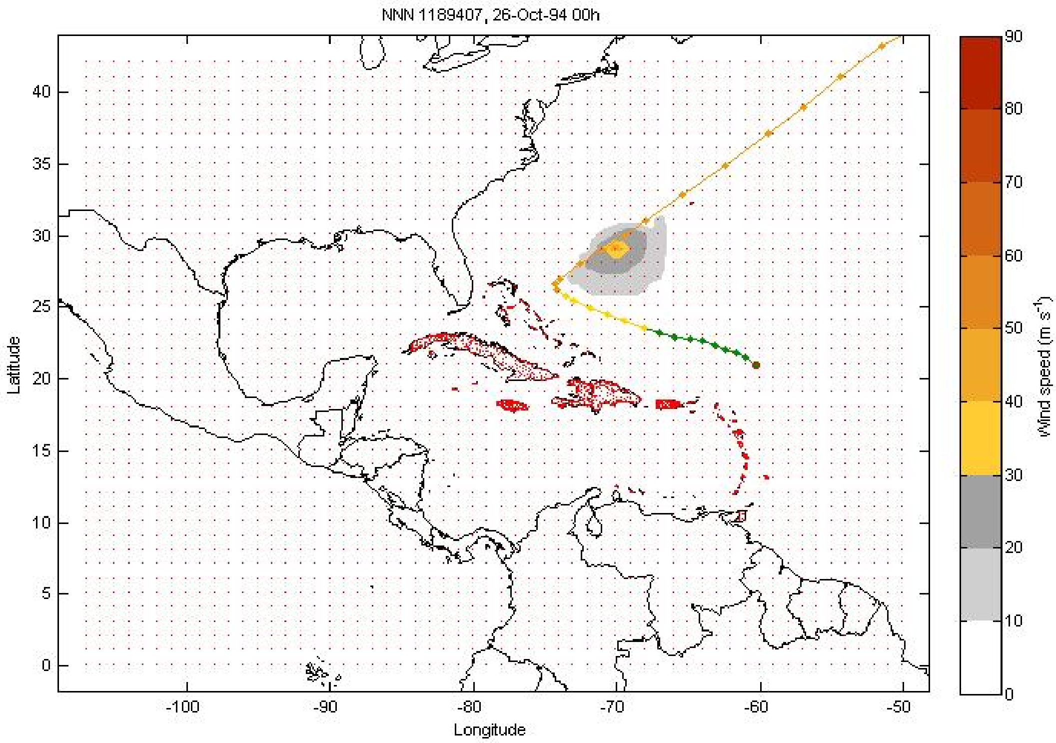


Figure 14: Wind field calculated for every time step.

12. Create the Hazard Set (i.e. all Footprints)

MATLAB call:
hazard = **climada_tc_hazard_set** (tc_track, hazard_set_file, centroids)

1x 1 struct

.reference_year
.peril_ID
.date — creation date of the set
.comment — free comment, normally containing the time the hazard event set has been generated
.windfield_comment
.filename — filename of the hazard event set (if passed as a struct, this is often useful)
.orig_years
.event_count
.orig_event_count

.lon
.lat
.centroid_ID — vector (No. of centroids x 1)

.matrix_density — density of the sparse array hazard.arr
.event_ID
.orig_event_flag
.frequency — vector (1 x No. of tc_tracks)
1 for original, 0 for probabilistic
.arr — sparse array, wind field (m/s) for all storms at each centroid

stdout for Mozambique data:

processing 4480 tracks (updating waitbar with estimation of time remaining every 100th track)
generating 4480 windfields took 256.157000 sec (0.057178 sec/event)
saving hazard set as C:\Documents and Settings\s3bxxw\Desktop\Lea_climada_local\climada\data\hazards\TC_MO

Code:

```
for track_i=1:length(tc_track)
	...
	res = climada_tc_windfield(tc_track(track_i),centroids,1,1);
	hazard.arr(track_i,:) = sparse(res.gust); % fill hazard array
	...
end
```

change of frequency when generating 9 probabilistic tracks of each historical one:
frequency = 1 / (orig_years * (ens_size + 1)) = 1 / ori_frequency / (ens_size+1)
frequency is diminished by factor 10

13. Create a Climate Change Scenario

13.1. Hazard independent and IPCC related code

Create climate change scenario based on today's hazard set. The intensity or frequency of storms, or particular categories of storms, can be changed accordingly to a specific climate change scenario, by the input structure *screw*.
The default values for screw are the climate change projections from IPCC special report on weather extremes (SREX, March 2012) with the given time horizon of 2100.
The time horizon can be requested for any given year and the climate change projections are linearly interpolated. The default values for *screw*:

screw.variable_to_change	'frequency'
screw.frequency	0.8
Screw.time_horizon	2100
screw.cat	[4 5]
screw(2).variable_to_change	'frequency'
screw(2).frequency	-0.28
screw(2).time_horizon	2100
screw(3).cat	[0 1 2 3]
screw(e).variable_to_change	'intensity'
screw(3).frequency	0.11
screw(3).time_horizon	2100
screw(3).cat	[0 1 2 3 4 5]

MATLAB call:

hazard = **climada_hazard_clim_scen** (hazard, tc_track, hazard_save_name, reference_year, screw)

climada_hazard_clim_scen

1x 1 struct

.reference_year	Is the requested time horizon, e.g. 2017, 2030...
.peril_ID	
.date	creation date of the set
.windfield_comment	
.filename	filename of the hazard event set (if passed as a struct, this is often useful) and enhanced with _cc_reference year
.orig_years	
.event_count	
.orig_event_count	
.lon .lat .centroid_ID	vector (No. of centroids x 1)
.matrix_density	density of the sparse array hazard.arr
.event_ID .orig_event_flag .frequency	vector (1 x No. of tc_tracks) 1 for original, 0 for probabilistic
.arr	sparse array, wind field (m/s) for all storms at

	each centroid, including **climate change scenario based on change in frequency and intensity**.
.comment	TCNA climate change scenario

stdout

```
Reference year for hazard_cc: 2017
***
frequency increased by      4.55% for category 4  5 for reference year 2017
***
frequency decreased by     -1.59% for category 0  1  2  3 for reference year 2017
***
intensity increased by      0.63% for category 0  1  2  3  4  5 for reference year 2017

***Climate change scenario ***
  saved in C:\Documents and saved in C:\Documents and Settings\s3bxxw\Desktop\Lea_climada_local\climada\data\hazards\hazard_clim.mat
```

Technical Hints

Remember: hazard contains the hazard event set, hazard.arr the sparse array with intensities, hazard.frequency the vector of event (occurrence) frequencies. Hence the following options to parametrize the climate scenario are easily be implemented (see `climada_hazard_clim_scen`)

- Increase wind speed
  hazard_cc.arr(tc_to_change,:) = hazard_cc.arr(tc_to_change,:)(1+frequency_screw_reference_year);*

- Increase frequency
  hazard_cc.frequency(tc_to_change) = hazard_cc.frequency(tc_to_change)(1+frequency_screw_reference_year);*

- Increase all wind speeds by 5 m/s,
  attention: hazard.arr is sparse, hence:
  ```
  nz_pos             = find(hazard.arr) %non-zeros
  hazard.arr(nz_pos)= hazard.arr(nz_pos)+5
  ```

- Increase only wind speeds > 45 m/s by 5 m/s
  ```
  pos45              = find(hazard.arr>45)
  hazard.arr(pos45) = hazard.arr(pos45)+5
  ```

13.2. Specific TC related code (initial version of code in 13.1)

Create climate change scenario based on today's hazard set. The intensity of storms is enhanced by a certain intensity_screw (default 1.05, 5% increase of wind speed). The frequency of all storms is increased by 10% default (frequency_screw = 1.10).

MATLAB call:

hazard = **climada_tc_hazard_clim_scen** (hazard, hazard_clim_file, frequency_screw, intensity_screw)

climada_tc_hazard_clim_scen ([], [], 1.1, 1.05)

1x 1 struct

.reference_year	
.peril_ID	
.date	creation date of the set
.windfield_comment	
.filename	filename of the hazard event set (if passed as a struct, this is often useful)
.orig_years	
.event_count	
.orig_event_count	
.lon .lat .centroid_ID	vector (No. of centroids x 1)
.matrix_density	density of the sparse array hazard.arr
.event_ID .orig_event_flag .frequency	vector (1 x No. of tc_tracks) 1 for original, 0 for probabilistic
.arr	sparse array, wind field (m/s) for all storms at each centroid, including **climate change scenario based on change in frequency and intensity**.
.comment	climate change scenario based on TCXXXX
.frequency_screw_applied	
.intensity_screw_applied	

stdout

```
***Climate change scenario ***
        intensity screw    =    1.05
        frequency_screw  =    1.10
saved in
...\climada\data\hazards\ TCXXXXX_hazard_clim.mat
```

Technical Hints

Remember: hazard contains the hazard event set, hazard.arr the sparse array with intensities, hazard.frequency the vector of event (occurrence) frequencies. Hence the following options to parametrize the climate scenario are easily be implemented (see `climada_tc_hazard_clim_scen`)

- Increase wind speed for all events by 5%
  ```
  hazard.arr = hazard.arr*1.05
  ```

- Increase frequency by 5%
  ```
  hazard.frequency =hazard.frequency*1.05
  ```

- Increase all wind speeds by 5 m/s,
  attention: hazard.arr is sparse, hence:
  ```
  nz_pos = find(hazard.arr) %non-zeros
  hazard.arr(nz_pos)= hazard.arr(nz_pos)+5
  ```

- Increase only wind speeds > 45 m/s by 5 m/s
  ```
  pos45 = find(hazard.arr>45)
  hazard.arr(pos45) = hazard.arr(pos45)+5
  ```

14. Analyze Statistics; Plot Wind Speed for Specific Return Periods at all Centroids for Historical Data Set, Probabilistic Data Set or Climate Change Scenario

Plot wind speed based historical, probabilistic or climate change data, for requested return periods at all centroids. If no return periods are specified, it takes return periods indicated in climada_global.DFC_return_periods.

MATLAB call:

hazard = **climada_hazard_stats** (hazard, return_periods, hazard_R_file, check_plot, centroids, rain, check_printplot)

climada_hazard_stats ([], 1)

.intensity_fit_ori
.R_fit_ori
.intensity_fit
.R_fit

Figures can be saved in
...\results\hazard_stats_historical.pdf
...\results\hazard_stats_probabilisitc.pdf
...\results\hazard_stats_climate.pdf

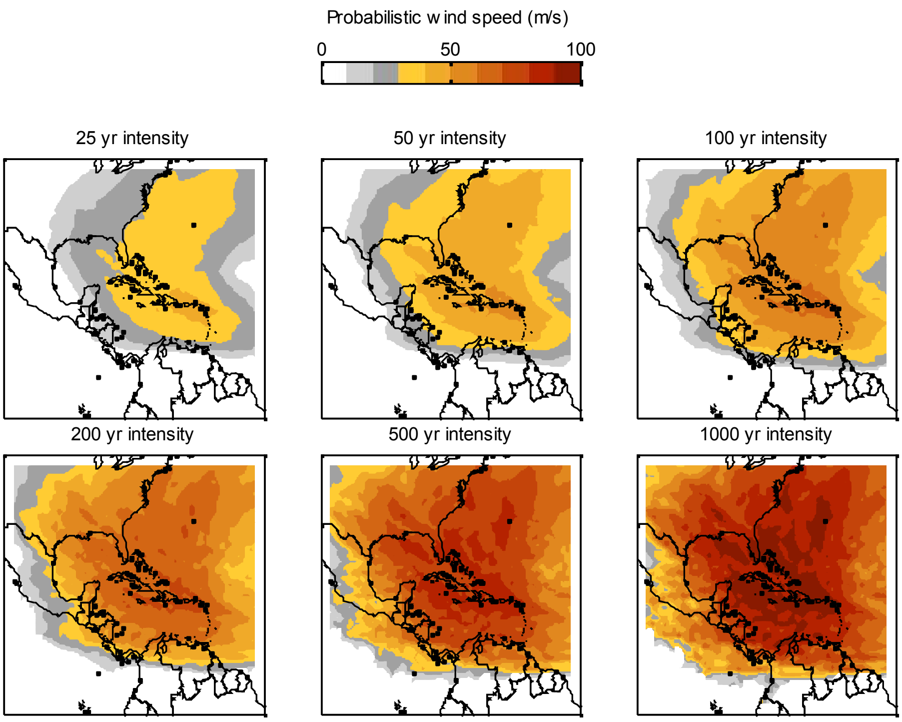


Figure 15: Wind speed maps for specific return periods.

15. Read Entity; Assets, Vulnerability, Deductibles and Covers

MATLAB call:

entity = **climada_entity_read** (entity_filename, hazard)
climada_entity_read

1x 1 struct

.assets
- .excel_file_name
- .Latitude
- .Longitude
- .Value
- .Deductible
- .Cover
- .VulnCurveID
- .centroid_index
- .Value_2030
- .hazard.comment

(.Latitude through .Value_2030: vector (1 x No. of centroids))

.vulnerability
- .exce_file_name
- .VulnCurveID (1, 2 and 3)
- .Intensity (wind speed in m/s)
- .MDD
- .PAA
- .MDR

(.VulnCurveID through .MDR: vector (1 x 30))

.measures
- .excel_file_name
- .name
- .color
- .cost
- .hazard_intensity_impact
- .hazard_high_frequency_cutoff
- .vuln_MDD_impact_a
- .vuln_MDD_impact_b
- .vuln_PAA_impact_a
- .vuln_PAA_impact_b
- .vuln_map
- .risk_transfer_attachement
- .risk_transfer_cover
- .color_RGB
- .vuln_mapping

.discount
- .excel_file_name
- .yield_ID
- .year
- .discount_rate

stdoutput

entity saved as mat-file in
...\climada\data\entities\entity_USFL_MiamiDadeBrowardPalmBeach2010.mat

16. Calculate the Event Damage Set (EDS)

Compute event damage for every storm and sum up for all centroids, based on asset value, MDD, PAA, deductibles and cover.

MATLAB call:

EDS = **climada_EDS_calc** (entity, hazard, annotation_name, EDS_save_file)

climada_EDS_calc

1x 1 struct

.hazard	.filename	
	.comment	
.assets	.filename	
.vulnerability	.filename	
.annotation_name		
.comment		
.event_ID		vector (No. of storms x 1)
.frequency		
.orig_event_flag		
.damage		
.reference_year		2010
.Value		total value of assets
.EL		losses for all storms summed up per centroid

.damage_sort
.damage_ori_sort
.R
.R_ori

Fitted wind speed for specific return periods
Linear interpolation between damage points
.damage_fit_ori
.damage_fit
.R_fit_ori
.R_fit

stdout

processing 189 assets and 1379 events, calculation took 0.188000 sec (0.000995 sec/event)
saving EDS as ...\climada\data\results\EDS_2030_clim.mat

Technical Hints

The variables have speaking names, note that the inner loop is vectorized. The key code within climada_EDS_calc (about line 155ff):

```
for asset_i=1:n_assets
     temp_damage = entity.assets.Value(asset_i)*MDD.*PAA
end % asset_i
```

with:

- temp_damage since it will be added in an 'outer loop' over asset_i
- entity.assets.Value(asset_i) is the Value of asset_i
  entity is a structure which contains all asset and vulnerability data
- MDD is here a vector of MDDs, PAA is the vector or PAAs
- .* is the element-wise (scalar) multiplication
- EDS stands for Event Damage Set, the set (or vector) of all event losses

In that calculation, the hazard intensity did not show up in the calculation, did we miss something? Well, the vulnerability is a function of the hazard intensity, hence:

MDD = f(hazard intensity) and PAA = f(hazard intensity)

where hazard intensity is the hazard intensity at asset_i for event_j, but event_j never shows up in the code, since the code is vectorized along the event dimension for performance reasons

And now, it gets technical (no way around this, sorry) – how to get the vector of MDDs

Remember: outer loop (explicit) over assets, inner loop (implicit) over events:

```
% approx line 115 in climada_EDS_calc.m
for asset_i=1:n_assets
    % the index of the centroid for given asset in the hazard
    % set
    asset_hazard_pos = entity.assets.centroid_index(asset_i);

     % find the vulnerability for the asset under consideration
    asset_vuln_pos = …find(entity.vulnerability.VulnCurveID ==
     entity.assets.VulnCurveID(asset_i));

     % convert hazard intensity into MDD: we need a trick to
     % apply
     % interp1 to the SPARSE hazard matrix: we evaluate only at
     % non-zero elements, therefore need a function handle (the @
     % below) to pass vulnerability to climada_sparse_interp:
     interp_x_table=
     entity.vulnerability.Intensity(asset_vuln_pos);

     interp_y_table = entity.vulnerability.MDD(asset_vuln_pos);

     % apply to non-zero elements only
     MDD = spfun (@climada_sparse_interp,
              hazard.arr(:,asset_hazard_pos));
```

```
 %   similarly,   convert   hazard   intensity   into   PAA
interp_y_table  =  entity.vulnerability.PAA(asset_vuln_pos);

 PAA = spfun (@climada_sparse_interp,
             hazard.arr(:,asset_hazard_pos));

%   calculate   the   from   ground   up   (fgu)   damage
temp_damage = entity.assets.Value(asset_i)*MDD.*PAA;
```

Remember: outer loop (explicit) over assets, inner loop (implicit) over events, now, we need to sum up over assets:

```
      % add to the EDS
      EDS.damage  = EDS.damage  + temp_damage';

      % add Value
      EDS.Value = EDS.Value + entity.assets.Value(asset_i);
  end % asset_i
```

A note on ' : for historical reasons the EDS.damage vector is transposed

Technical Hints on Insurance Conditions

Remember: outer loop (explicit) over assets, inner loop (implicit) over events

```
% approx line 115 in climada_EDS_calc.m
for asset_i=1:n_assets
    […]

    % calculate the from ground up (fgu) damage
    temp_damage = entity.assets.Value(asset_i)*MDD.*PAA;

    if entity.assets.Deductible(asset_i)>0 ||
             entity.assets.Cover(asset_i) <
             entity.assets.Value(asset_i)

        % apply Deductible and Cover
        temp_damage = min(max(temp_damage-...
            entity.assets.Deductible(asset_i)*PAA,0),...
            entity.assets.Cover(asset_i));
    end
    […]
end % asset_i
```

and similar for any conditions on the event damage set (EDS), always of the form min(max(damage-deductible,0) ,cover).

17. Plot Event Damage vs. Return Period

Plot occurrence damage exceedance frequency curve (DFC) for one or multiple event damage sets (EDS). If EDS not given, EDS is prompted for. Multiple EDS can be selected (press and hold shift to select multiple EDS:mat files).

Figure can be saved in
...\results\DFC_3.pdf (No. of DFCs displayed is attached)

MATLAB call:
climada_EDS_DFC
(EDS,EDS_comparison,Percentage_Of_Value_Flag)

climada_EDS_DFC

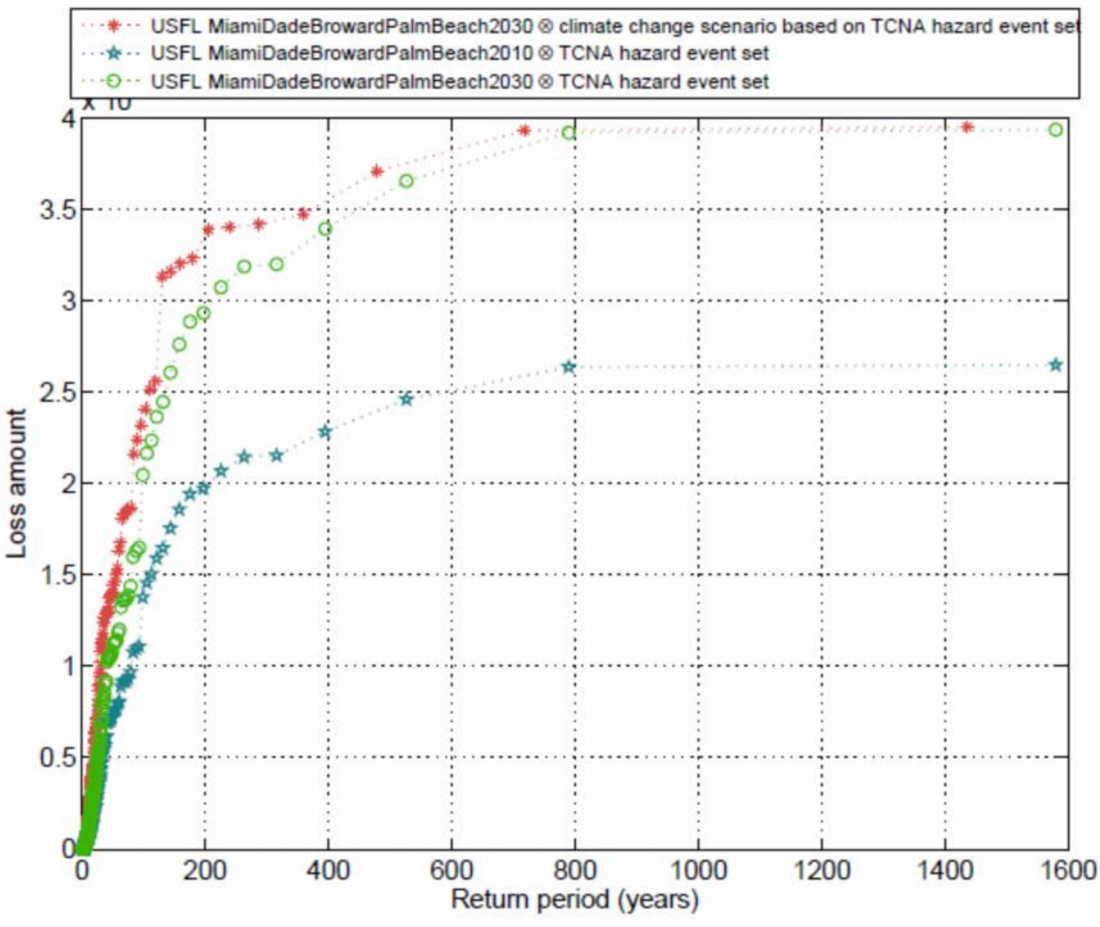


Figure 16: Damage frequency curve of today, economic growth and climate change.

18. Plot Event Damage vs. Specific Return Period

Calculates damage value for specified requested return period(s) from the damage exceedance frequency curve (DFC) for one event damage sets (EDS) and its historic EDS. If EDS not given, EDS is prompted for. Calculated damage values are stored in new variable. Figure can be saved in

...\results\DFC_SRP.pdf

MATLAB call:

climada_EDS_stats (EDS, EDS_save_file, return_period, check_plot)

climada_EDS_stats

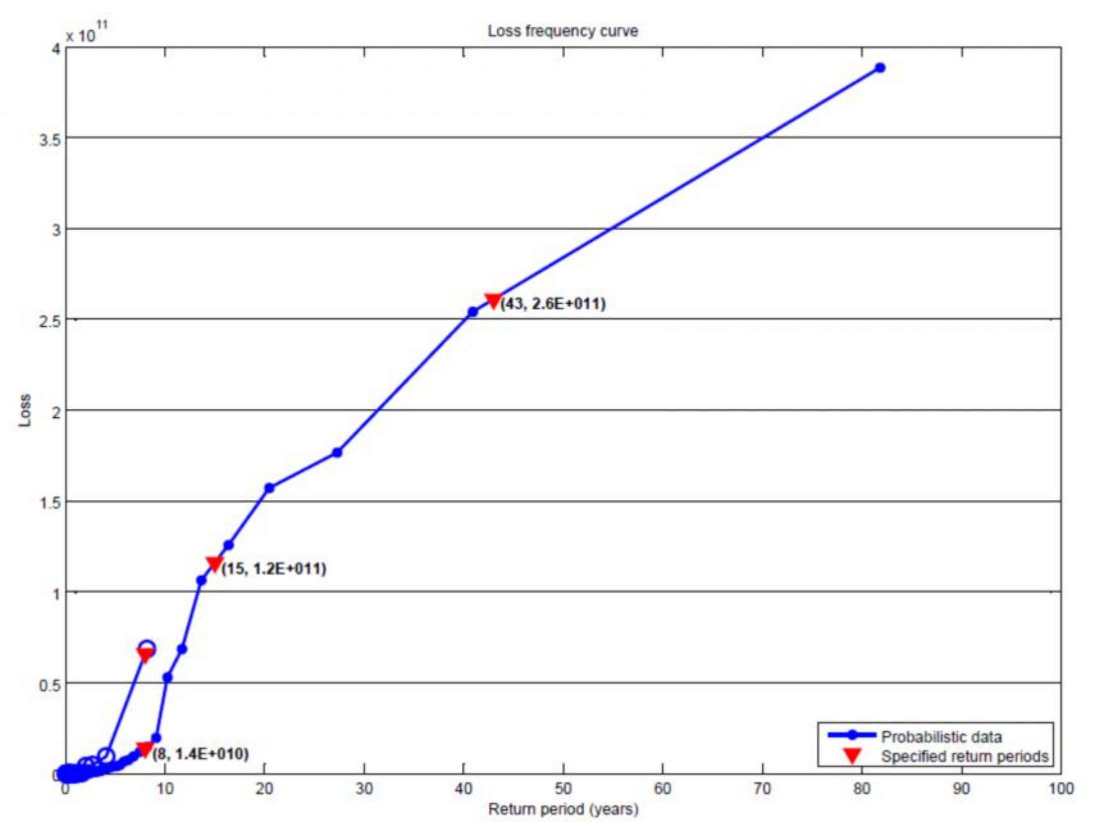


Figure 17: Damage frequency curve with specified return periods marked.

19. Plot Waterfall Figure for Today's Damage and Future's Damage Including Economic Growth and Climate Change Separately

Plot waterfall graph for specific return period based on today's damage data, economic growth damage, climate change scenario damage which corresponds to the future damage. If losses for requested return period are not available, the damage will be calculate with `climada_EDS_calc(EDS,'',return_period)`.

Inputs are the three event damage sets EDS (e.g. EDS_2010.mat, EDS_2030.mat, EDS_2030_clim.mat), prompted for if not given. Specific return period or annual expected damage can be chosen.

MATLAB call:

[EDS, damage] =	**climada_waterfall_graph** (EDS1, EDS2, EDS3, return_period, check_printplot);
	climada_waterfall_graph

Figure 18: waterfall plot.

Figure saved in
...\results\waterfall_graph_20_years.pdf

Function needs specific functions:

- arrow.m
- ellipse.m

20. Plot Waterfall Figure for all Return Periods (Animation)

MATLAB call:

climada_waterfall_graph_animation (EDS1, EDS2, EDS3);

climada_waterfall_graph_animation

Animation of todays damage, economic growth and climate change for all return periods. Press p, enter or x to print, go next return period or to end.

Figure saved in

...\results\waterfall_graph_20_years_animation.pdf

21. Calculate the Damages and Benefits of Measures

MATLAB call:

measures_impact = **climada_measures_impact** (entity, hazard, measures)

display the results with

MATLAB call:

climada_adaptation_cost_curve (measures_impact,measures_impact_comparison, x_text_control,y_text_control, scaled_AEL, nice_numbers)

climada_adaptation_cost_curve (
climada_measures_impact (entity, hazard, measures))

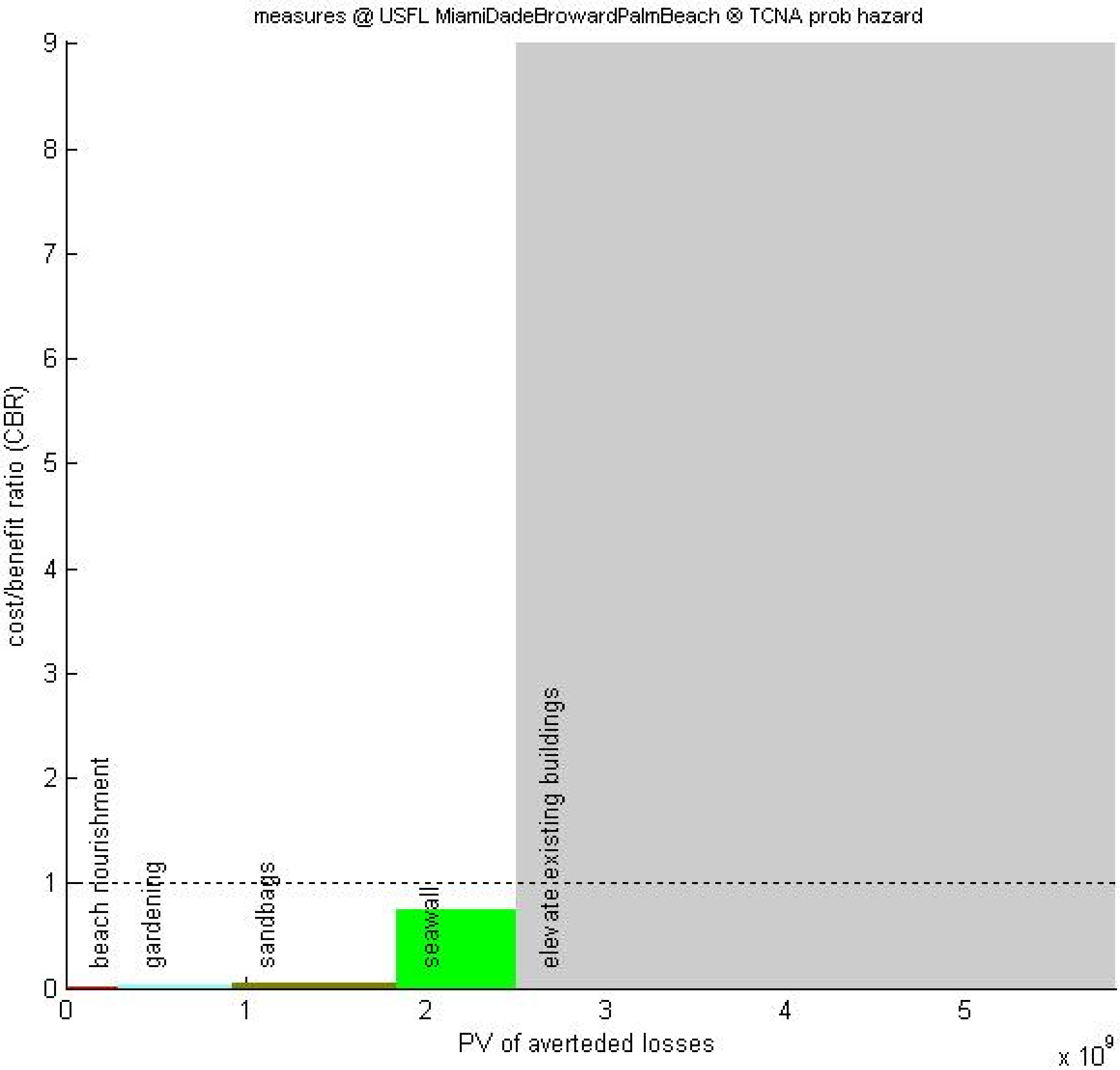


Figure 19: The adaptation cost curve for today's assets, today's hazard and measures.

Repeat the calculation now for future hazard:

MATLAB call:

climada_adaptation_cost_curve (measures_impact)

climada_adaptation_cost_curve (

climada_measures_impact (entity, hazard_clim))

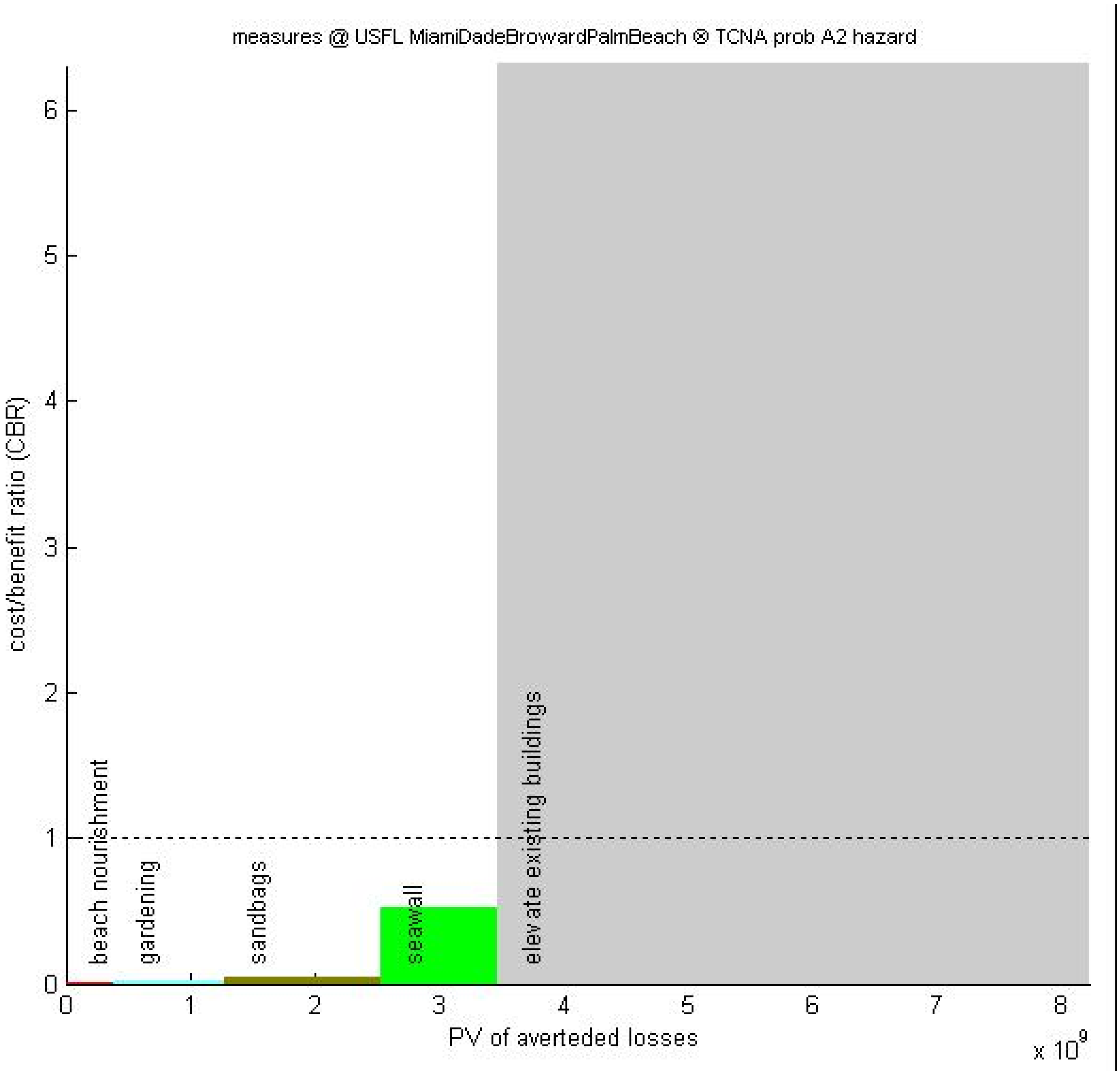


Figure 20: As above, but for future hazard (hazard_clim).

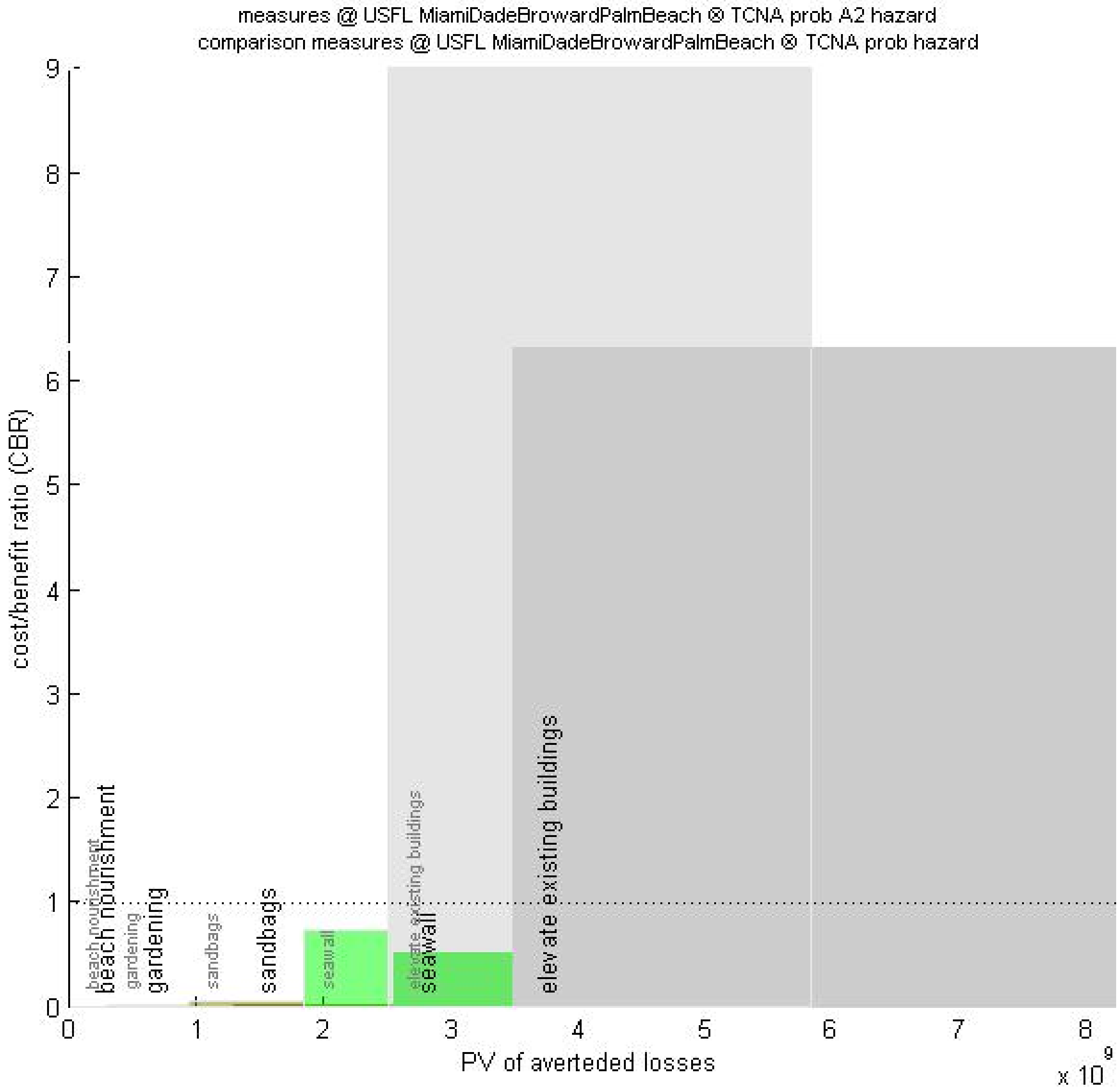


Figure 21: As above, but comparing the two hazards (future hazard (climate change), labelled A2, in darker colours)

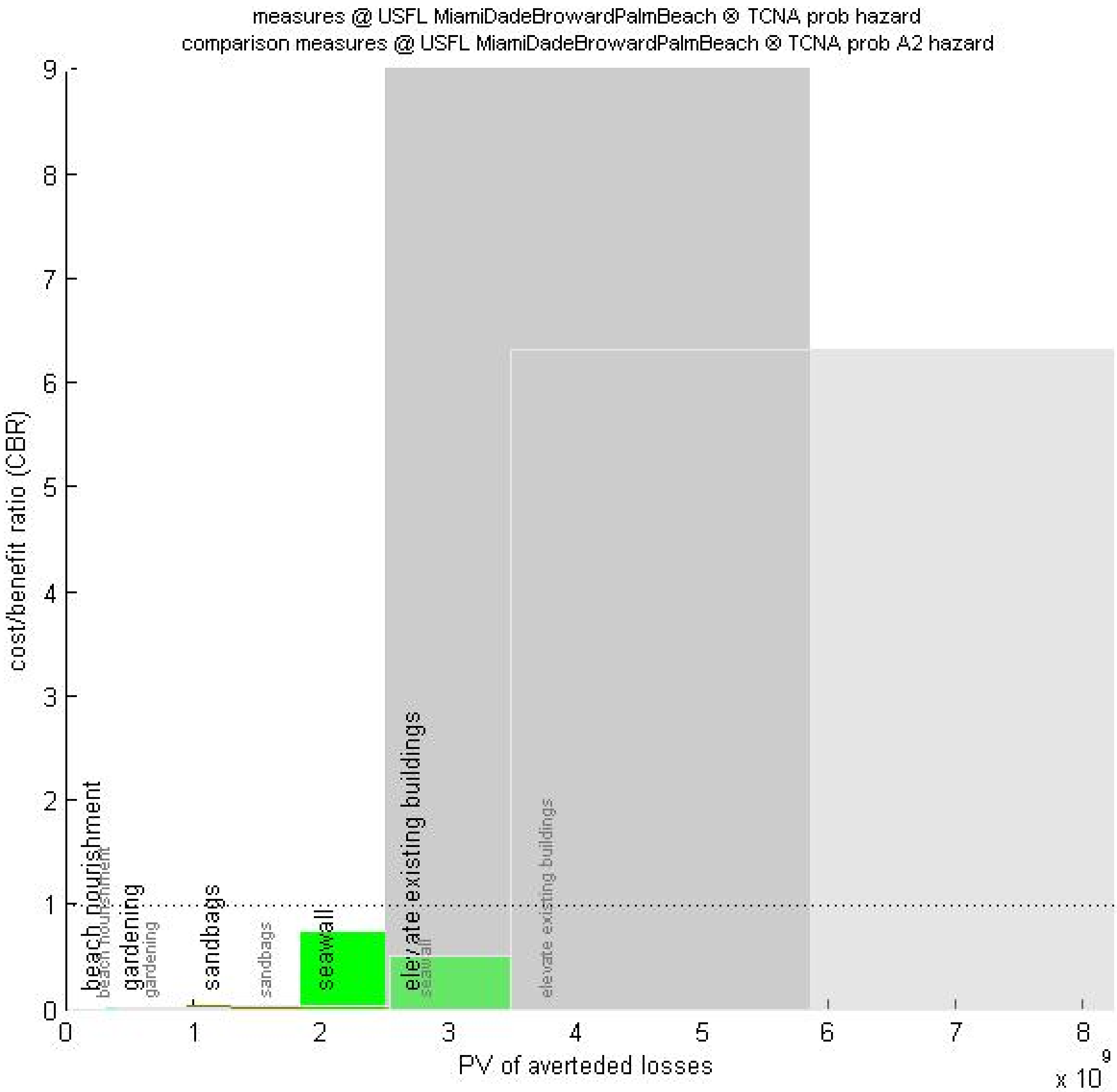


Figure 22: As above, but comparing the two hazards the other way round (future and today)

Proposed colours to use for measures:

Color	red	green	blue	red	green	blue	for excel
1	211	205	177	0.82	0.80	0.69	0.82 0.8 0.69
2	194	186	148	0.76	0.73	0.58	0.76 0.73 0.58
3	231	179	75	0.90	0.70	0.29	0.9 0.7 0.29
4	250	192	144	0.98	0.75	0.56	0.98 0.75 0.56
5	255	219	105	1.00	0.86	0.41	1 0.86 0.41
6	188	226	146	0.73	0.88	0.57	0.73 0.88 0.57
7	152	193	129	0.59	0.75	0.50	0.59 0.75 0.5
8	181	195	184	0.71	0.76	0.72	0.71 0.76 0.72
9	162	202	190	0.63	0.79	0.74	0.63 0.79 0.74
10	162	194	232	0.63	0.76	0.91	0.63 0.76 0.91
11	112	189	210	0.44	0.74	0.82	0.44 0.74 0.82
12	174	214	224	0.68	0.84	0.88	0.68 0.84 0.88
13	255	175	175	1.00	0.68	0.68	1 0.68 0.68
14	205	183	201	0.80	0.71	0.79	0.8 0.71 0.79
15	255	209	248	1.00	0.82	0.97	1 0.82 0.97
16	174	167	139	0.68	0.65	0.54	0.68 0.65 0.54
17	155	147	113	0.61	0.57	0.44	0.61 0.57 0.44
18	200	140	57	0.78	0.55	0.22	0.78 0.55 0.22
19	238	153	110	0.93	0.60	0.43	0.93 0.6 0.43
20	255	184	80	1.00	0.72	0.31	1 0.72 0.31
21	149	193	112	0.58	0.75	0.44	0.58 0.75 0.44
22	117	154	98	0.46	0.60	0.38	0.46 0.6 0.38
23	142	156	145	0.55	0.61	0.57	0.55 0.61 0.57
24	125	164	151	0.49	0.64	0.59	0.49 0.64 0.59
25	125	155	202	0.49	0.61	0.79	0.49 0.61 0.79
26	85	150	173	0.33	0.59	0.68	0.33 0.59 0.68
27	136	177	190	0.53	0.69	0.74	0.53 0.69 0.74
28	255	137	137	1.00	0.54	0.54	1 0.54 0.54
29	167	144	163	0.65	0.56	0.64	0.65 0.56 0.64
30	255	171	232	1.00	0.67	0.91	1 0.67 0.91

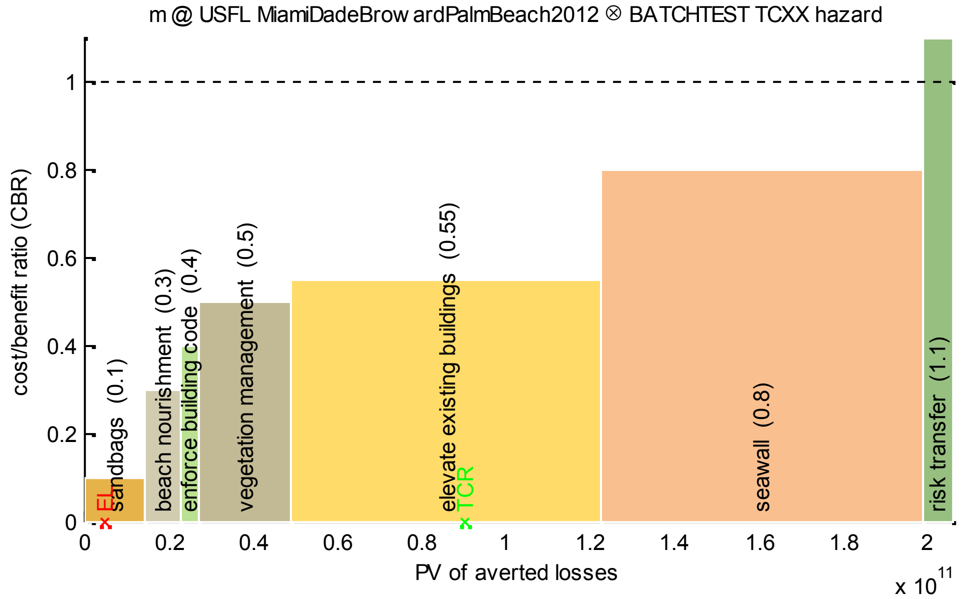


Figure 23: Example plot with proposed colours

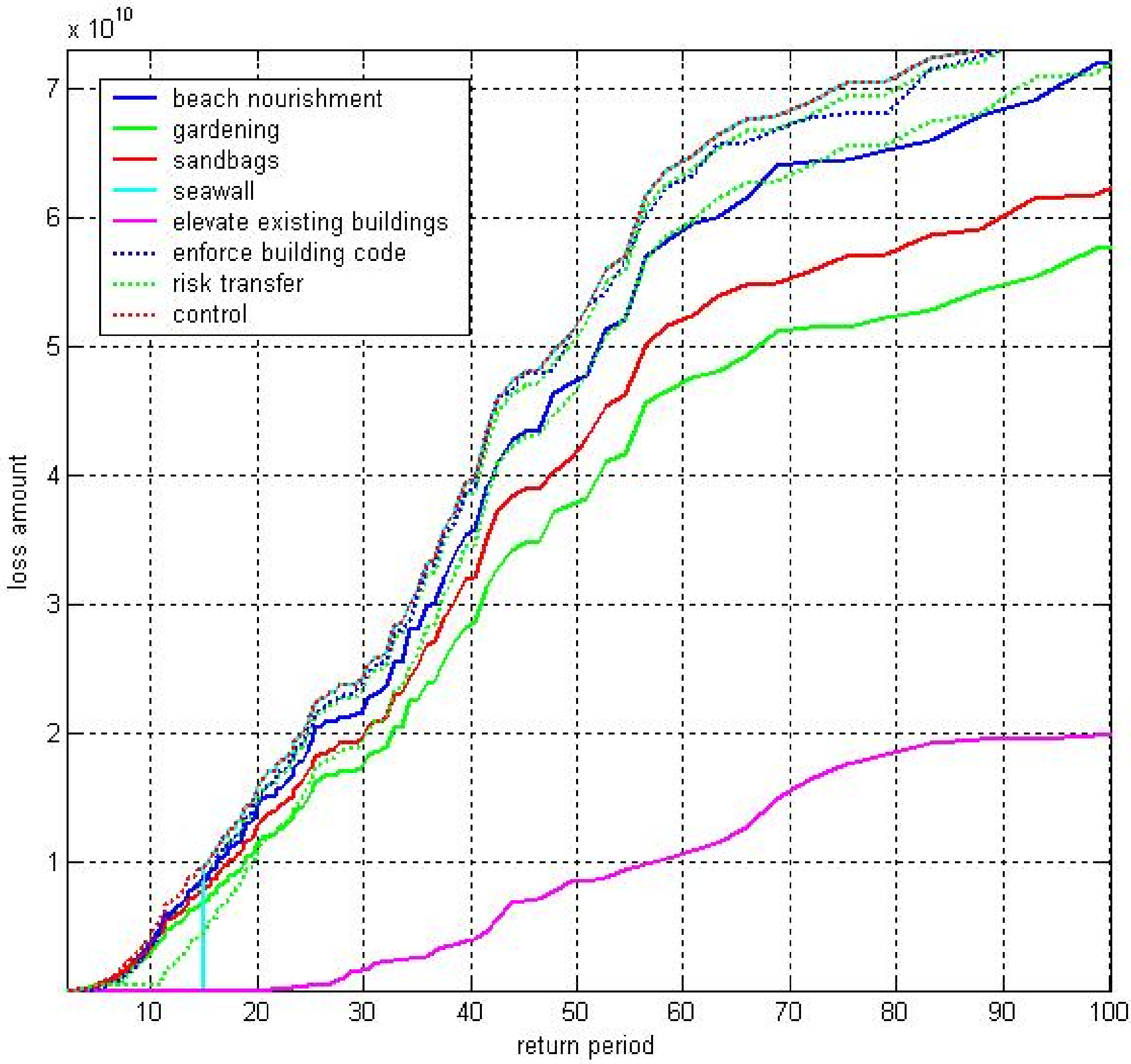


Figure 24: The occurrence damage exceedence frequency curve (DFC) for today's hazard. Note the effect of the dam (cut-off at 15yr return period, light blue curve) as well as the risk transfer (5 bn xs 500 mio USD, dotted yellow). Note further the prominent impact of 'elevate existing buildings', but the costs for that measure are prohibitively high.

21.1. Technical Hints

Risk transfer measures

Basically, any indemnity based risk transfer measure can be applied to the event damage set (EDS), simply as:

EDS_{after} = share*min(max(EDS_{before}-deductible,0), cover)

for any index based risk transfer, the EDS can be computed starting from the hazard event set and calculating the index value for each event. In the case of the simplest index, just a wind speed threshold T at a given location, payout P of $10 per m/s above threshold, this might look as simple as:

```
temp_hazard     = hazard.arr(*,hazard_pos);
nz_pos          = find(temp_hazard);
EDS_index       = min(temp_hazard(nz_pos)-T,0)*P
```

where hazard_pos contains the index of the centroid next to the station (determined using climada_geo_distance). Note that due to the sparsity of hazard.arr, the min function is speeded up by using find first.

More specifically, and to ease the use of risk transfer measures, they can also be specified in the measures tab of the entity Excel sheet[9]. In column 'risk transfer attachment', one enters the attachment point (synonym for deductible) in the same currency and currency unit as all other figures, and in column 'risk transfer cover' the cover. In column 'cost', one only needs to enter the cost in addition to the pure expected damage (which is calculated within climada, when the risk transfer gets applied). Costs for risk transfer are – to keep it simple here – a fixed amount for management expenses and capital costs that scale first order with the cover[10]. As an approximation, one might use rules of thumb to determine a proxy for the sum of management expenses and capital costs, like:

1. determine sqrt(GLM), where GLM=sqrt(attachment_point*cover)
2. look up the probability of a damage of size sqrt(GLM) on the DFC (without measures, to keep it simple)
3. proxy for sum of costs is max(sqrt(probability of damage),0.01)*cover

[9] all entity Excel files can be found in …\climada\data\entities
[10] this is a very crude assumption. As climada only adds the expected damage costs, one needs to be careful here.

Adaptation cost curve

Generally on how the adaptation cost curve calculation is performed:

1. Calculate Present Value (PV) of costs of measure (in Excel)
2. Today (year 2010): assets, hazard as per 2010
   a. calculate annual expected damage with no measures (climada_measures_impact)
   b. calculate annual expected damage with measure applied (climada_measures_impact)
      → difference 1.1) minus 1.2) gives you benefit of measure today
3. Future (year 2030): assets, hazard as per 2030
   a. calculate annual expected damage with no measures (climada_measures_impact)
   b. calculate annual expected damage with measure applied (climada_measures_impact)
      → difference 2.1) minus 2.2) gives you future benefit of measure
4. Discount benefits (climada_measures_impact)

   → horizontal axis of adaptation cost curve
   compare with PV of costs → vertical axis of adaptation cost curve

Part 2: Torrential Rain from Tropical Cyclones

22. Theoretical Background

Sources:

- Fabian Schönenberger, NatCat SwissRe
- Tuleya et at., 2006: Evaluation of GFDL and Simple Statistical Model Rainfall Forecasts for U.S. Landfalling Tropical Storms.

The symmetric rainfall field can be computed after Tuleya et al., 2006 who developed a simple rainfall climatology and persistence model (R-CLIPER). Rainfall rates are calculated as a function of storm intensity and radius. The accumulated rainfall rate along the storm track can then be derived by integrating the rainfall rate along the storm track, give the intensity.

R-CLIPER:

The radial structure of the TRMM profiles suggests that the following equation could be used to represent the TRMM rain rates (TRR) as a function of radius and maximum wind (V):

$$\mathrm{TRR}(r, V) = T_0 + (T_m - T_0)(r/r_m) \quad r < r_m \quad \text{(3a)}$$

and

$$\mathrm{TRR}(r, V) = T_m \exp[-(r - r_m)/r_e] \quad r \geq r_m, \quad \text{(3b)}$$

where T_0 is the rain rate at $r = 0$ and T_m is the maximum rain rate at $r = r_m$. In Eq. (3), TRR varies linearly with radius from $r = 0$ to $r = r_m$, and then decays exponentially for $r \geq r_m$ starting from T_m. Equation (3) has four parameters (T_0, T_m, r_m, and r_e). From a least squares fit of the TRMM radial profiles in Fig. 6, it was found that the dependence on storm intensity could be accounted for by making these four parameters linear functions of the storm intensity as follows:

$$T_0 = a_1 + b_1 U, \quad \text{(4a)}$$

$$T_m = a_2 + b_2 U, \quad \text{(4b)}$$

$$r_m = a_3 + b_3 U, \quad \text{(4c)}$$

and

$$r_e = a_4 + b_4 U, \quad \text{(4d)}$$

where U is the normalized maximum wind given by

$$U = 1 + (V_m - 35)/33 \qquad (5)$$

and V_m is the maximum wind speed in knots. Table 2 lists the a_1–a_4 and b_1–b_4 that were obtained from the fit of Eqs. (4) and (5) to the TRMM rainfall profiles, assuming that the maximum winds of the three profiles in Fig. 6 are 45, 80, and 115 kt. The units of the coefficients were chosen to give the rain rate in units of inches per day. Equations (3)–(5) provide an excellent represen-

TABLE 2. The constants from the fit of the TRMM rainfall rates as a function of radius and storm maximum wind for the R-CLIPER model. The bottom four rows are the bias-corrected constants used by the NHC in the operational version.

	Intercepts	Slopes
Preliminary	$a_1 = -2.11$ in. day^{-1}	$b_1 = 3.63$ in. day^{-1}
	$a_2 = -2.73$ in. day^{-1}	$b_2 = 4.24$ in. day^{-1}
	$a_3 = 69.1$ km	$b_3 = -8.49$ km
	$a_4 = 215$ km	$b_4 = -35.8$ km
NHC	$a_1 = -1.10$ in. day^{-1}	$b_1 = 3.96$ in. day^{-1}
	$a_2 = -1.60$ in. day^{-1}	$b_2 = 4.80$ in. day^{-1}
	$a_3 = 64.5$ km	$b_3 = -13.0$ km
	$a_4 = 150$ km	$b_4 = -16.0$ km

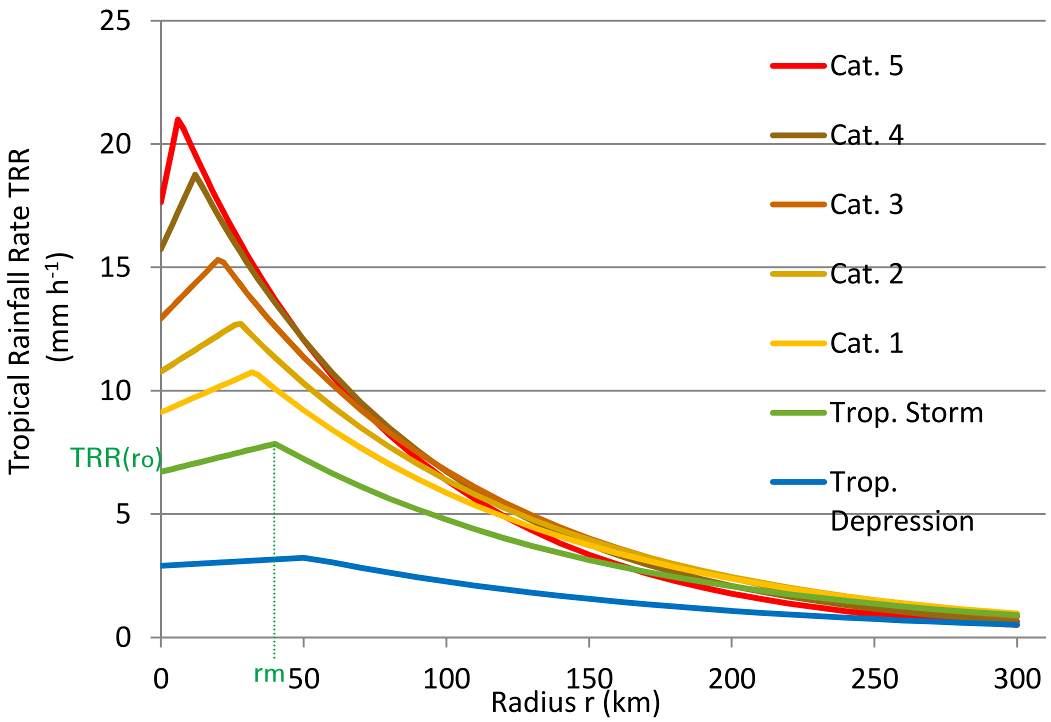


Figure 25: R-CLIPER radial rainfall rates profiles based on the Saffir-Simpson Hurricane wind speeds. Rainfall rates increase linearly between r = 0 and r = rm, which is the maximum rainfall rate, and decay exponentially with r>rm.

More variables that influence the distribution and amount of precipitation:
- Forward Speed
- Curvature
- Vertical Wind Shear:Precipitation Pattern Asymmetries
- Topography:Orographic Enhancement on windward hillside
- Interaction with Frontal Boundaries/Upper Level Troughs

23. Generate the Rain Sum Footprint

Generate rain sum field resulting from single track of tropical cyclone. The function computes rain rates for every hour and accumulates rain fall for every storm, based on R-CLIPER (symmetric rain field). Results saved in res.rainsum as total rain fall in mm per storm.

Equal time steps of one hour (interpolation of longitude, latitude, maximum sustained wind speed, minimum pressure).

Analog to climada_tc_windfield.

Plot footprint of rainfall sum. If silent_mode = 1, no graph displayed.

MATLAB call:

[res tc_track centroids] =	**climada_tc_rainfield** (tc_track, centroids, equal_timestep, silent_mode, check_plot)
	1x 1 struct
	.rainsum (total rain per storm in mm at every centroid)
	.ID .lat .lon — vector (1 x No. of centroids)

Uses functions
- climada_nonspheric_distance_m
- climada_RCLIPER

Figure can be saved in
\results\footprint_rainsum_stormname.pdf

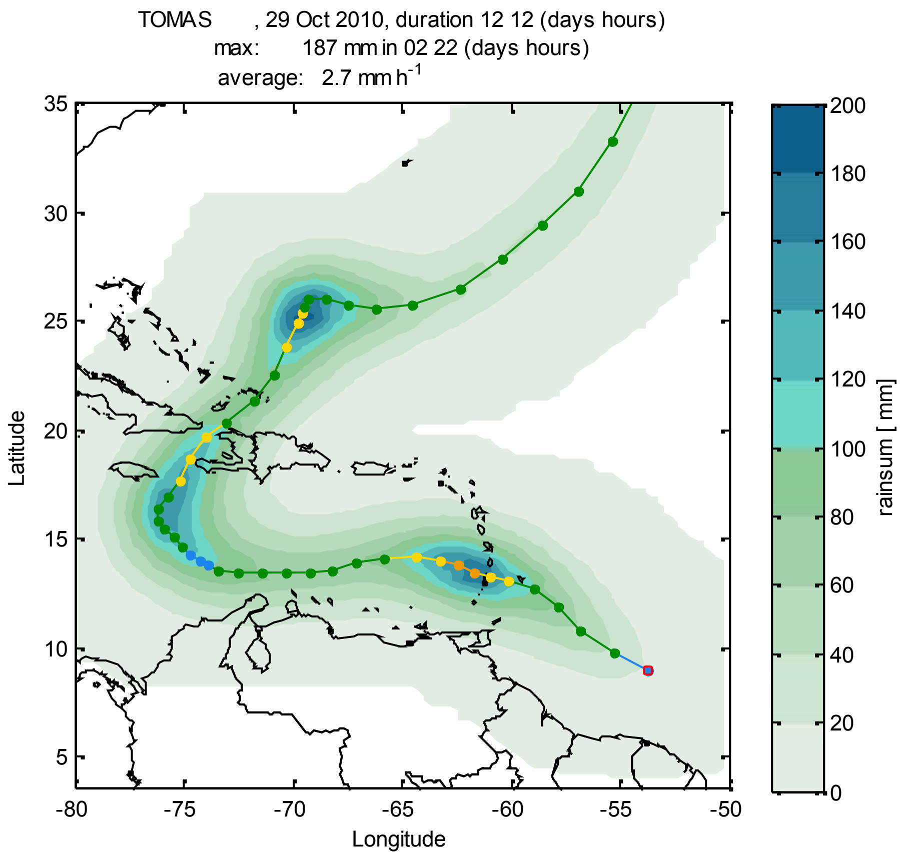


Figure 26: Rainfall sum footprint.

23.1. Calculate Rain Rate for Each Node of Specific Tc Track Based on Symmetric Rain Field R-CLIPER

Given the windspeed (kn) at a specific node calculate the rain rate at all centroids according to RCLIPER (symmetric rainfield),

R-CLIPER (Tuleya et at., 2006: Evaluation of GFDL and Simple Statistical Model Rainfall Forecasts for U.S. Landfalling Tropical Storms):

> The radial structure of the TRMM profiles suggests that the following equation could be used to represent the TRMM rain rates (TRR) as a function of radius and maximum wind (V):
>
> $$\mathrm{TRR}(r, V) = T_0 + (T_m - T_0)(r/r_m) \quad r < r_m \quad (3a)$$
>
> and
>
> $$\mathrm{TRR}(r, V) = T_m \exp[-(r - r_m)/r_e] \quad r \geq r_m, \quad (3b)$$

MATLAB call:

rainrate = **climada_RCLIPER** (fmaxwind_kn, inreach, Radius_km)

24. Calculate the Rain Rate Fields for a Single Track and Display as Animation

MATLAB call:

climada_tc_rainrate_field_animation (tc_track, centroids, aggregation, check_avi)

climada_tc_rainrate_field_animation

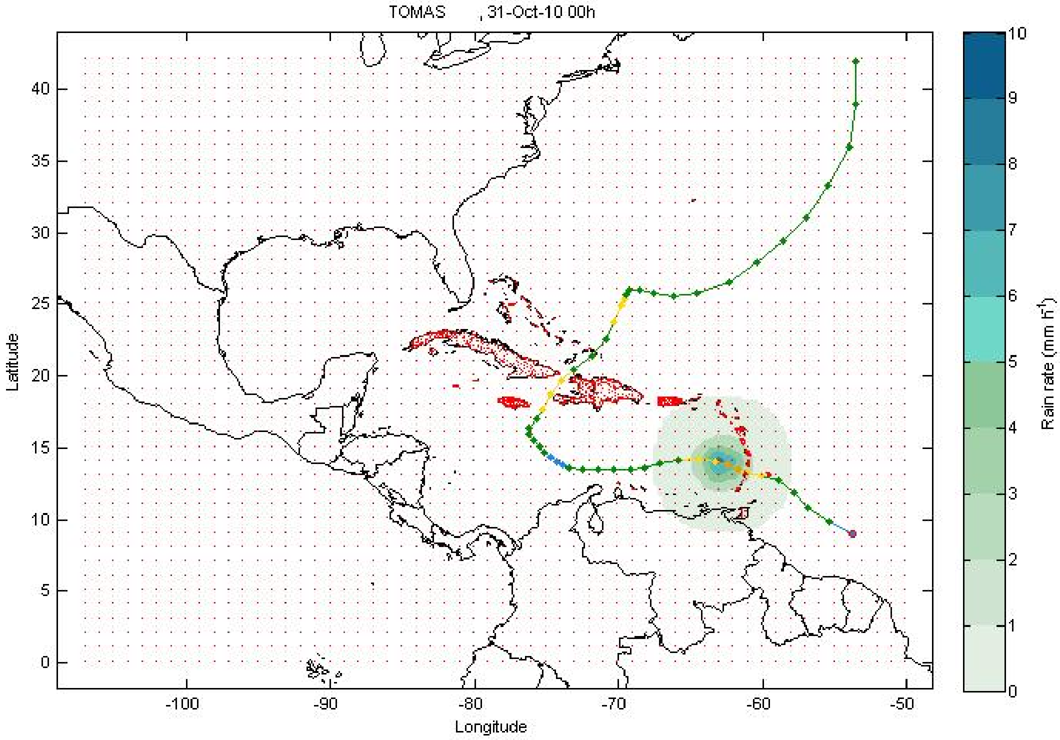


Figure 27: Rain rate field for the highest probabilistic storm

25. Calculate the rain sum fields for a single track and display as animation

MATLAB call:

climada_tc_rainsum_field_animation (tc_track, centroids, aggregation, check_avi)

climada_tc_rainsum_field_animation

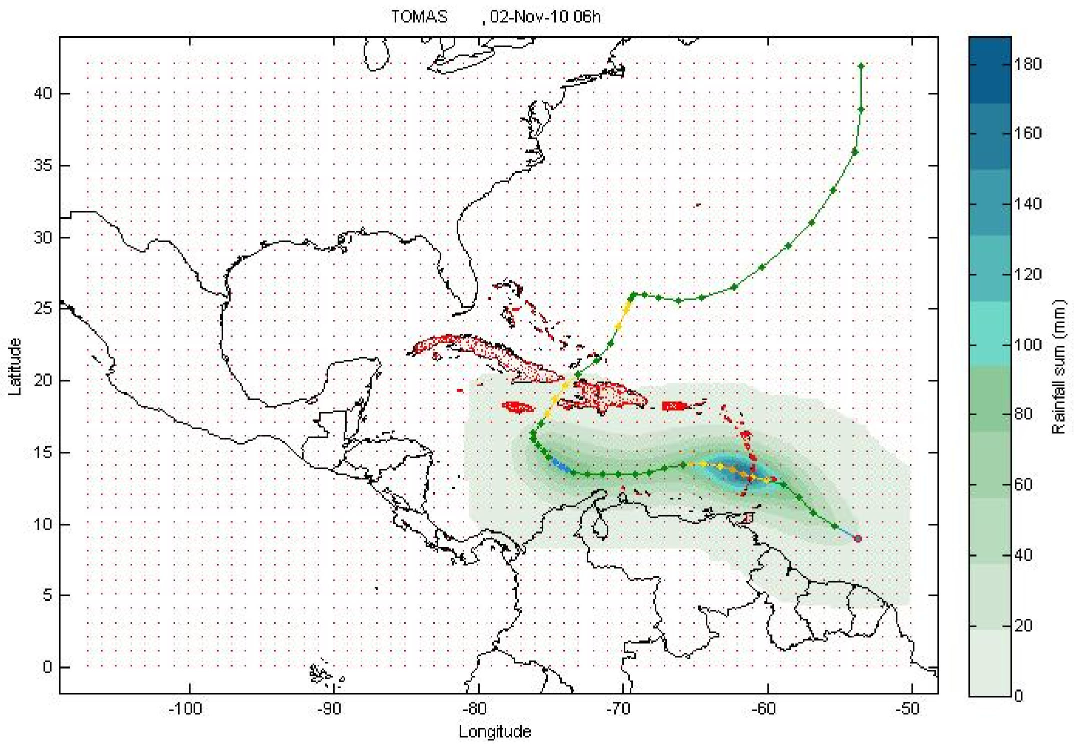


Figure 28: Rainfall sum field.

26. Generate the Rain Hazard Set (i.e. All Rain Sum Footprints)

Generate hazard rain set from tc_tracks.

MATLAB call:
hazard = **climada_tc_hazard_rain** (tc_track, hazard_set_file, centroids)

1x 1 struct

.reference_year	
.peril_ID	TC rain
.date	creation date of the set
.comment	free comment, normally containing the time the hazard event set has been generated
.windfield_comment	
.filename	filename of the hazard event set (if passed as a struct, this is often useful)
.orig_years	
.event_count	
.orig_event_count	
.lon **.lat** **.centroid_ID**	vector (No. of centroids x 1)
.matrix_density	density of the sparse array hazard.arr
.event_ID **.orig_event_flag** **.frequency**	vector (1 x No. of tc_tracks) 1 for original, 0 for probabilistic
.arr	sparse array, rain sum (mm) for all storms at each centroid

Needs functions

- climada_tc_rainfield (tc_track, centroids)
- climada_nonspheric_distance_m
- climada_RCLIPER

27. Analyze Statistics; Plot Rain Sum for Specific Return Periods at all Centroids for Historical Data Set, Probabilistic Data Set or Climate Change Scenario

Plot rain sum based historical, probabilistic or climate change data, for requested return periods at all centroids.

MATLAB call:

hazard = **climada_hazard_stats** (hazard, return_periods, hazard_R_file, check_plot, centroids, rain)

climada_hazard_stats ([], [], [], 1, [],1)

.intensity_fit_ori
.R_fit_ori
.intensity_fit
.R_fit

Figures can be saved in
...\results\hazard_stats_historical.pdf
...\results\hazard_stats_probabilisitc.pdf
...\results\hazard_stats_climate.pdf

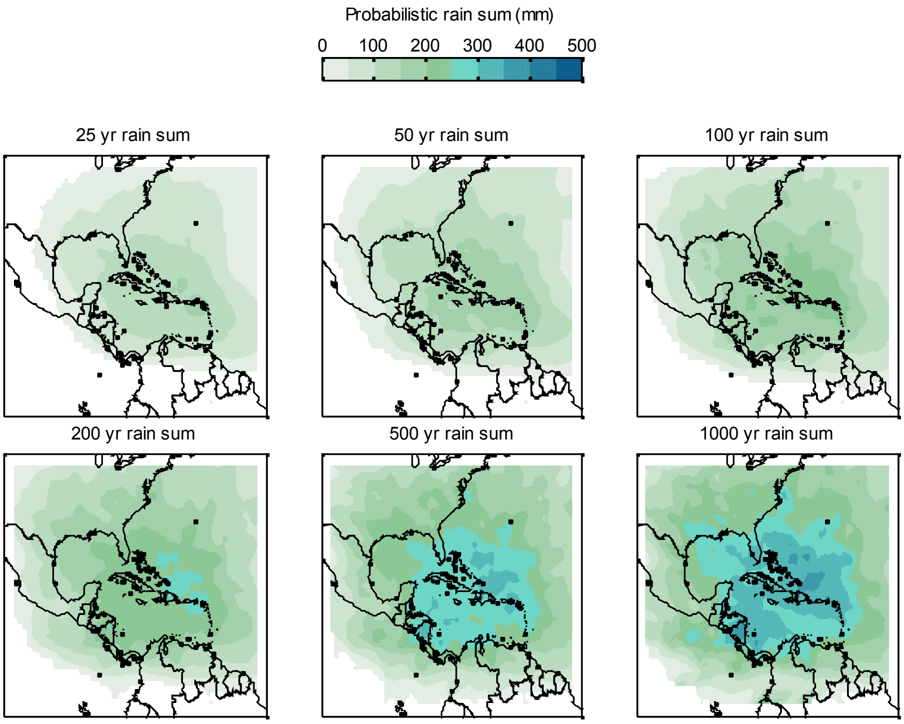


Figure 29: Rain sum maps for specific return periods.

28. Plot Waterfall Figure for Today's Damage and Future's Damage Including Economic Growth and Climate Change Separately for one or two hazards

Include one or two hazard (3 EDS per hazards). The EDS files get sorted (according to hazard type and EDS size automatically)

For two hazards
MATLAB call:
[EDS, damage] = **climada_waterfall_graph_advanced** `(return_period, check_printplot, EDS1, EDS2, EDS3, EDS4, EDS5, EDS6)`

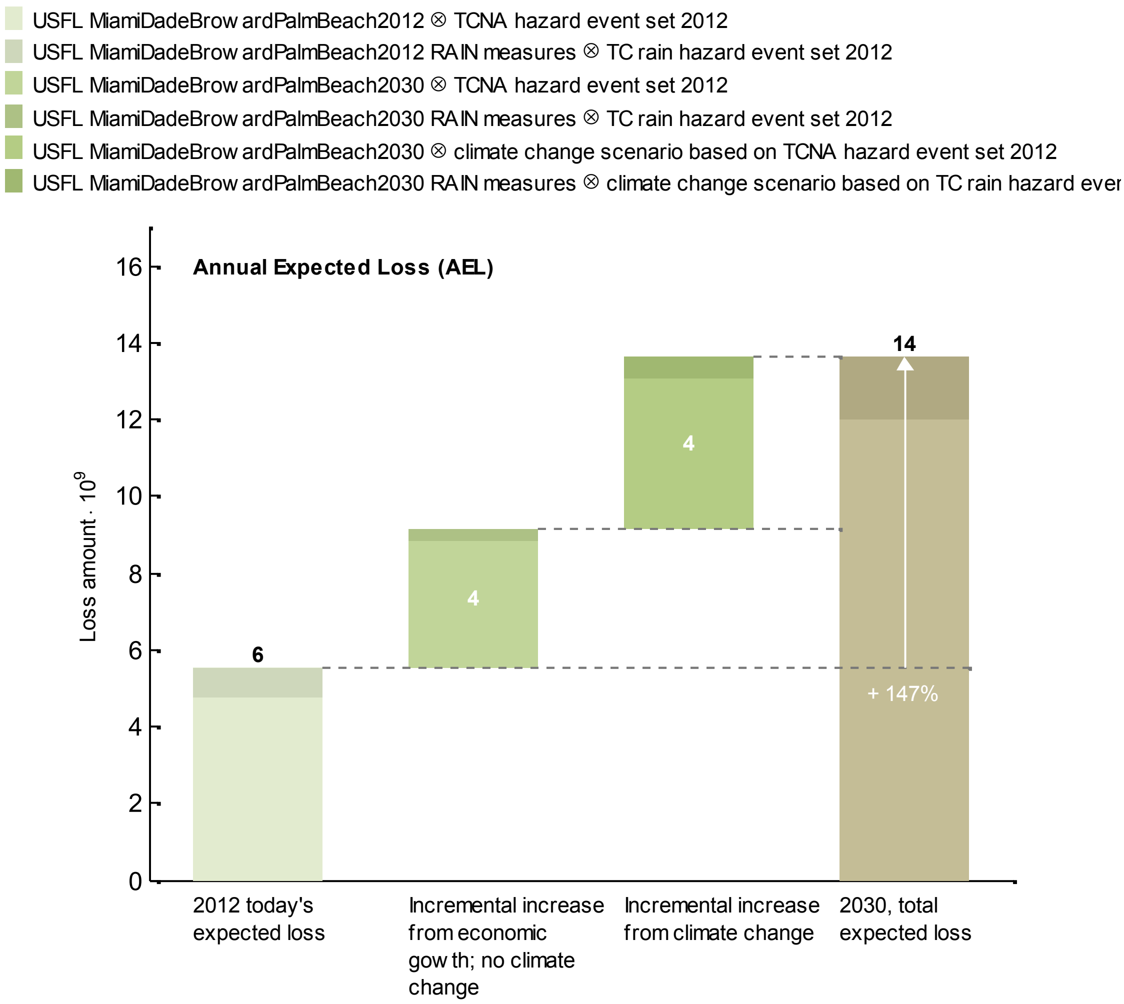


Figure 30: waterfall plot for the hazard rain and wind.

29. Collect the Damages and Benefits of Measures of different hazards

MATLAB call :

Impacts_collected**=climada_collect_measures_impact**`(impact1,impact2)`

1. sums up benefits (damage averted) for shared measures
2. cost of shared measures: take higher costs if costs not the same --> WARNING
3. It is assumed that the two hazards are insured separately, therefore

make sure to not use the same name for both hazards e.g. risk_transfer_rain and risk_transfer_wind. If one insurance covers both hazards sum up the losses of both hazards, apply the risk transfer and calculate the NPV of the benefits and the premium

display the results with

MATLAB call:

climada_adaptation_cost_curve (impacts_collected)

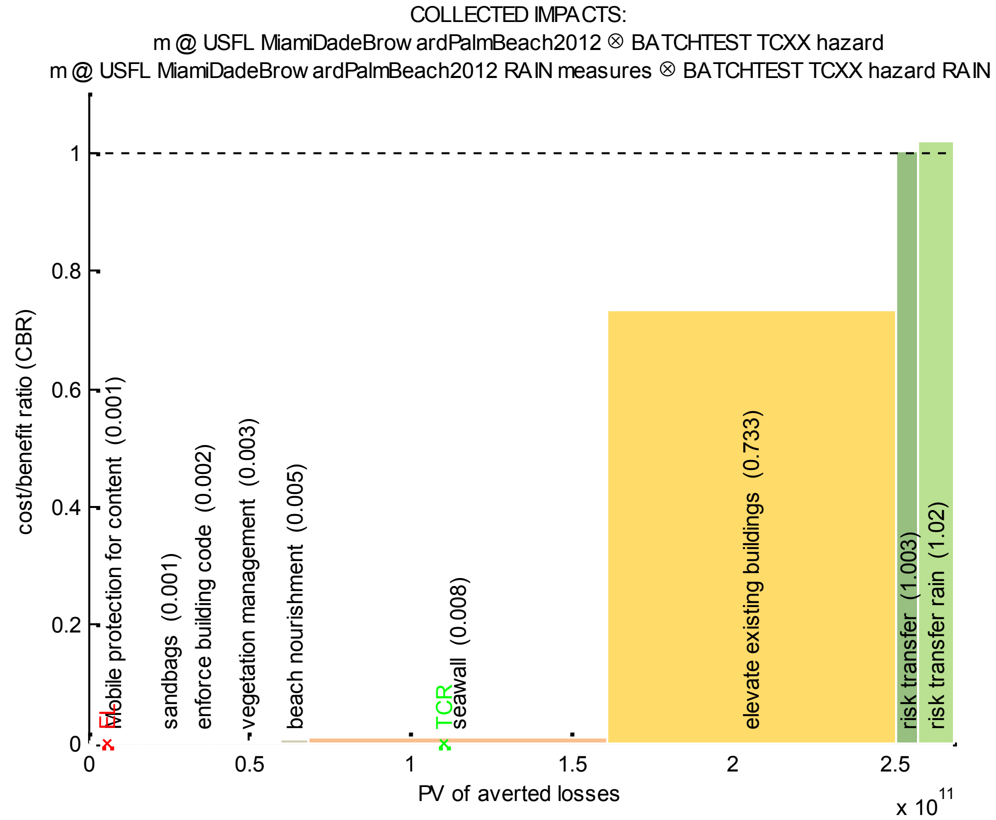


Figure 31: Cost curve of collected impacts.

Appendix

Appendix A – List of all Climada Functions

See the file *climada_code_overview.htm* for a list af all headers of all climada functions for reference. See the code `compile_function_header_doc` to automatically (re)generate this document.

Appendix B – Structure of the Raw Tropical Cyclone Data Files

See the function climada_tc_read_unisys_database to read raw tropical cyclone data files. There are three basic types of datalines in the Best Track.

```
a) TYPE A:

  92620 08/16/1992 M=13  2 SNBR= 899 ANDREW       XING=1
  SSS=4
  Card# MM/DD/Year Days S# Total#... Name........US Hit.Hi
  US category

  TYPE B:
  92580 04/22S2450610   30 1003S2490615   45 1002S2520620
  45 1002S2550624  45 1003*
  Card#
  MM/DD&LatLongWindPress&LatLongWindPress&LatLongWindPress
  &LatLongWindPress

  TYPE C:
  92760 HRCFL4BFL3 LA3
  Card# TpHit.Hit.Hit.

  TYPE A:

     Card# = Sequential card number starting at 00010 in
  1886

     MM/DD/Year = Month, Day, and Year of storm

     Days = Number of days in which positions are
  available (note that this
            also means number of lines to follow of type B
  and then one line
            of type C)

     S# = Storm number for that particular year (including
  subtropical storms)

     Total# = Storm number since the beginning of the
  record (since 1886)

     Name = Storms only given official names since 1950
```

```
   US Hit = '1' = Made landfall over the United States
as tropical
                  storm or hurricane,
            '0' = did not make U.S. landfall

   Hi US category = '9' = Used before 1899 to indicate
U.S. landfall as a
                          hurricane of unspecified
Saffir-Simpson category
                    '0' = Used to indicate U.S. landfall
as tropical storm,
                          but this has not been utilized
in recent years
                    '1' to '5' = Highest category on the
Saffir-Simpson scale
                                 that the storm made
landfall along the U.S.
                                 '1' is a minimal
hurricane, '5' is a
                                 catastrophic hurricane
```

b) TYPE B:

```
   Card# = As above.

   MM/DD = Month and Day of Storm

   & = 'S' (Subtropical stage), '*' (tropical
cyclone stage),
       'E' (extratropical stage), 'W' (wave stage -
rarely used)

   LatLong = Position of storm:  24.5N, 61.0W

   Wind = Maximum sustained (1 minute) surface
(10m) windspeed in knots (in
          general, these are to the nearest 5
knots).

   Press = Central surface pressure of storm in mb
(if available).  Since
           1979, central pressures are given
everytime even if a satellite
           estimation is needed.

   Positions and intensities are at 00Z, 06Z, 12Z,
18Z

TYPE C:

   Card# = As above.

   Tp = Maximum intensity of storm ('HR' =
hurricane, 'TS' = tropical
        storm, 'SS' = subtropical storm)
```

```
   Hit  =  U.S.  landfallings  as  hurricane  ('LA'  =
Louisiana, etc.) and
         Saffir-Simpson category at landfall ('1' =
minimal hurricane...
         '5'  =  super  hurricane).   (Note  that
Florida and Texas are split
         into smaller regions:  'AFL' = Northwest
Florida, 'BFL' =
         Southwest  Florida,  'CFL'  =  Southeast
Florida, 'DFL' = Northeast
         Florida,  'ATX'  =  South  Texas,  'BTX'  =
Central Texas, 'CTX' =
         North Texas.)
```

Appendix C - Generation of Probabilistic Tropical Cyclone Tracks

Theoretical Background

Total number of tropical storms.

http://www.metoffice.gov.uk/weather/tropicalcyclone/northatlantic.html
The number of tropical storms observed over the season is the best known measure of the level of storm activity. However, the total number of storms tells us little about variations in the intensity and lifetime of storms from one season to the next.

Accumulated cyclone energy (ACE)

http://en.wikipedia.org/wiki/Accumulated_cyclone_energy
This is a measure of the collective intensity and duration of all tropical storms over the season and thus reflects storm lifetimes and intensities as well as total numbers over the season.

Accumulated cyclone energy (ACE) is a measure used by the National Oceanic and Atmospheric Administration (NOAA) to express the activity of individual tropical cyclones and entire tropical cyclone seasons, particularly the North Atlantic hurricane season. It uses an approximation of the energy used by a tropical system over its lifetime and is calculated every **six-hour period**. The ACE of a season is the sum of the ACEs for each storm and takes into account the number, strength, and duration of all the tropical storms in the season.[1]

The ACE of a **season** is calculated by summing the squares of the estimated maximum sustained velocity of every active tropical storm (wind speed 35 knots (65 km/h) or higher), at **six-hour intervals**. If any storms of a season happen to cross years, the storm's ACE counts for the previous year.[2] The numbers are usually divided by 10,000 to make them more manageable. The unit of ACE is 10^4 kt^2, and for use as an index the unit is assumed. Thus:

$$\mathrm{ACE} = 10^{-4} \sum v_{\mathrm{max}}^2$$

where v_{max} is estimated sustained wind speed in knots.

Kinetic energy is proportional to the square of velocity, and by adding together the energy per some interval of time, the accumulated energy is found. As the duration of a storm increases, more values are summed and the ACE also increases such that longer-duration storms may accumulate a larger ACE than more-powerful storms of lesser duration. Although ACE is a value proportional to the energy of the system, it is not a direct calculation of energy (the mass of the moved air and therefore the size of the storm would show up in a real energy calculation).

A related quantity is hurricane destruction potential (HDP), which is ACE but only calculated for the time where the system is a hurricane.[1]

A season's ACE is used to categorize the hurricane season by its activity. Measured over the period 1951–2000 for the Atlantic basin, the median annual index was 87.5 and the mean annual index was 93.2. The NOAA categorisation system[3] divides seasons into:

- *Above-normal season*: An ACE value above 103 (117% of the 1951–2000 median), provided at least two of the following three parameters exceed the long-term average: number of tropical storms (10), hurricanes (6), and major hurricanes (2).
- *Near-normal season*: neither above-normal nor below normal
- *Below-normal season*: An ACE value below 66 (75% of the 1951–2000 median)

Atlantic hurricane seasons by ACE index, 1950–2010

The term hyperactive is used by Goldenberg et al. (2001) [4] based on a different weighting algorithm[5] which places more weight on major hurricanes, but typically equating to an ACE of about 153 (171% of the current median). For the in progress season be advised that the ACE is preliminary based on NHC bulletins, which may later be revised.

Key

- ACE Accumulated cyclone energy
- TS Number of tropical storms (including subtropicals)
- HR Number of hurricanes (S-S Category 1 – 5)
- MH Number of major hurricanes (Category 3 – 5)

Season	ACE	TS	HR	MH	Classification
2005 Atlantic hurricane season	**248**	**28**	**15**	7	Above normal (*hyperactive*)
1950 Atlantic hurricane season	243	13	11	**8**	Above normal (*hyperactive*)
1995 Atlantic hurricane season	228	19	11	5	Above normal (*hyperactive*)
2004 Atlantic hurricane season	225	15	9	6	Above normal (*hyperactive*)
1961 Atlantic hurricane season	205	11	8	7	Above normal (*hyperactive*)
1955 Atlantic hurricane season	199	12	9	6	Above normal (*hyperactive*)

Fields with record values are in **bold**

Maximum intensity theory

Kerry. 1988: The Maximum Intensity of Hurricanes

ftp://texmex.mit.edu/pub/emanuel/PAPERS/max88.pdf

DeMaria and Kaplan, 1993: Sea Surface Temperature and the Maximum Intensity of Atlantic Tropical Cyclones.
http://studentresearch.wcp.muohio.edu/HurricanesPhyBioIEffects/articles/SST.MI.atlantic.pdf

Tropical Cyclone Tracks in the Southwest Indian Ocean

Information on historical tropical cyclones is available from Unisys Weather Homepage (http://weather.unisys.com/hurricane/index.php) that was recorded by the Joint Typhoon Warning Center (JTWC). It provides position in latitude and longitude, maximum sustained winds and central pressure.
Following World Meteorological Organization (WMO) guidelines, most regions use a 10-minute average. However, the Joint Typhoon Warning Center (JTWC), Guam, and WMO Region IV (United States and Caribbean area) use a **1-minute standard average**. However, the Saffir-Simpson Hurricane Scale is based on wind speed measurements averaged over a 1-minute period, at 10 m (33 ft) above the surface (http://en.wikipedia.org/wiki/Tropical_cyclone_scales).

As a rule of thumb the factor 0.88 is used in going from a 1-minute system to a 10-minute system such that (http://www.nrlmry.navy.mil/~chu/chap6/se200.htm):

- TEN-MINUTE MEAN = 0.88 * ONE-MINUTE MEAN or
- ONE-MINUTE MEAN = 1.14 * TEN-MINUTE MEAN

Data is available from January 1978 to March 2011. Multiple tropical cyclone tracks are occasionally linked together due to the date line (Figure 32) and need to be separated. For the further analysis only tropical cyclone tracks in the southwestern Indian Ocean are taken into account.

448 tracks are recorded to have occurred in the south western Indian Ocean. Tropical cyclone tracks for the period of record are displayed in Figure 33.

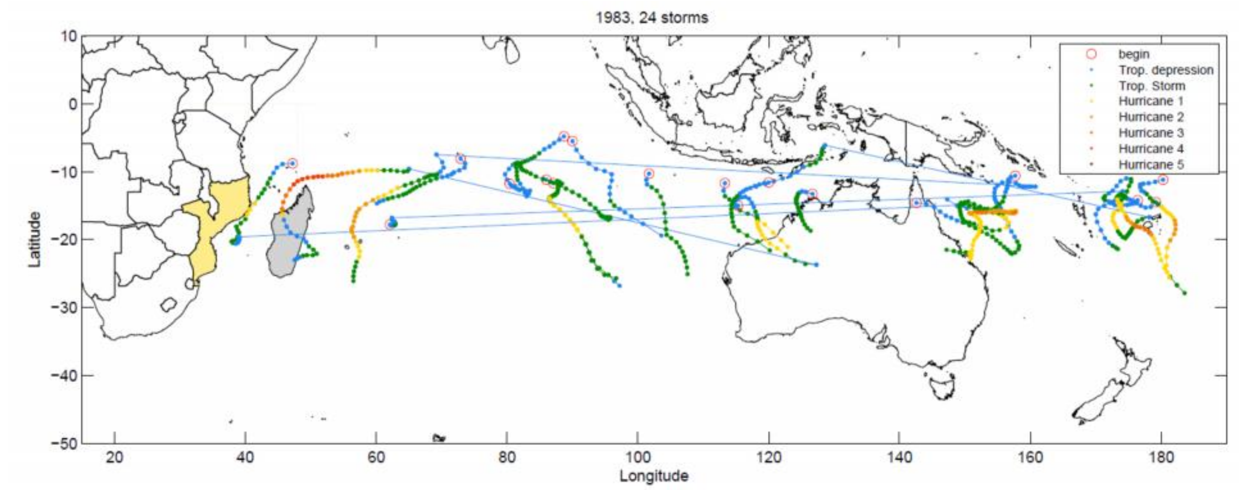


Figure 32: Tropical cyclone tracks in the year 1983 with colour coding based on the Saffir-Simpson Hurricane Scale. Tracks can be linked together due to the date line and need to be separated for the further analysis.

Figure 33: Movie of all the tropical cyclones recorded in the south western Indian Ocean from 1978 to 2011.

Characteristics of Tropical Cyclones from 1978 to 2011

The tropical cyclones in the south western Indian ocean from 1978 to 2011 can be characterized by available cyclone energy (ACE), number of tropical storms, hurricanes and major hurricanes per season.

Figure 34 show the above mentioned characteristics and the histograms based on all the tracks recorded in the south western Indian Ocean and based on the 161 tracks recorded with a longitude smaller than 60°, respectively.

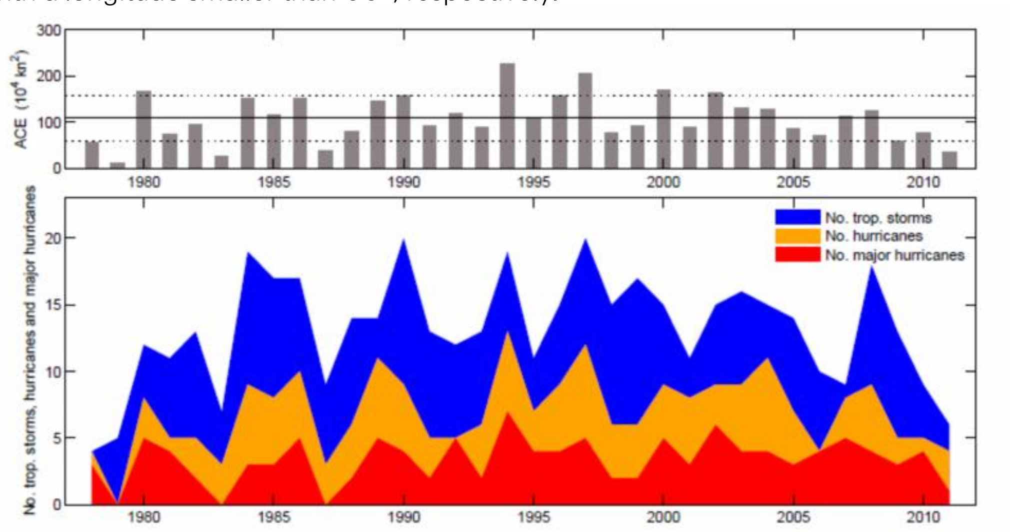


Figure 34: Available cyclone energy (ACE) during the cyclone seasons from 1978 to 2011 (top) and number of tropical storms, hurricanes and major hurricanes for the same seasons, based on all 448 tracks recorded in the south western Indian Ocean.

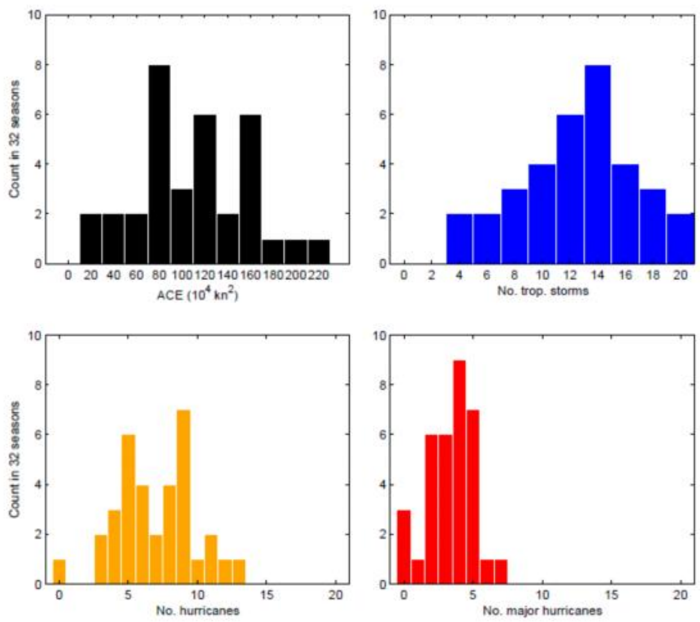


Figure 35: Histograms of available cyclone energy (ACE) (top left), number of tropical storms per season (top right), number of hurricanes per season (bottom left) and number of major hurricanes per season (bottom right). The histograms are based on all 448 tracks recorded in the south western Indian Ocean.

Modeling Approach for Wind Speed

Wind speed is the sum of an initial wind speed v0 and independent and identically distributed changes in wind speed vi (http://www.mathematik.uni-ulm.de/stochastik/aktuelles/sh06/sh_rumpf.pdf, 15 June 2011)

Figure 36 shows the distribution of initial wind speed v0 and difference in wind speed vi for all the tracks recorded in the south western Indian Ocean.

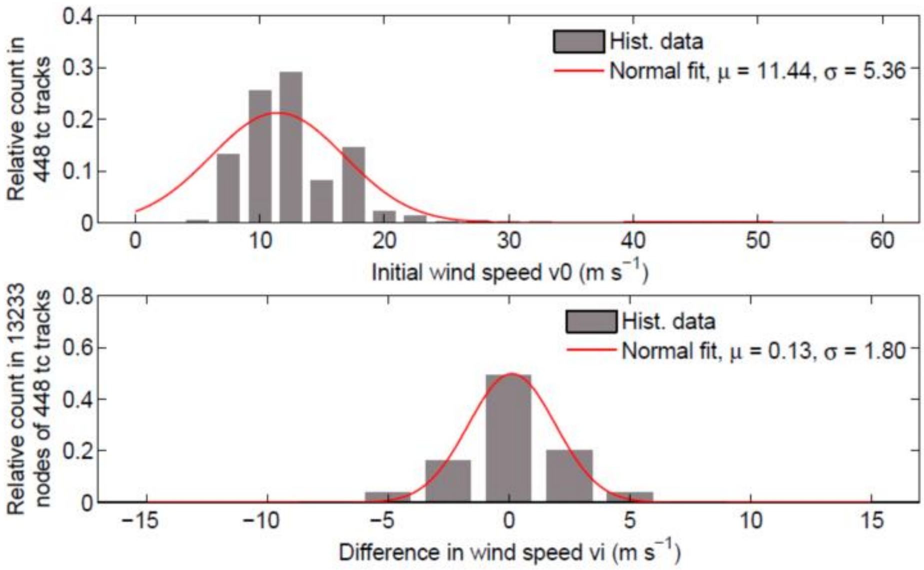


Figure 36: Distribution of initial wind speed v0 (top) and difference in wind speed vi (bottom) for the tracks recorded in the south western Indian Ocean during 1978 and 2011.

Normal distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$f(x) = e^{ax^2+bx+c}.$$

Mean $\mu = -b / \boxed{2a}$

Variance $\sigma^2 = -1 / \boxed{2a}$

one must choose c such that $\int_{-\infty}^{\infty} f(x)\, dx = 1$

Generation of Probabilistic Wind Speed

Plot distribution of initial wind speed and change in wind speed from historical tracks. Fit a normal distribution to each and display according mu and sigma of initial wind speed and change in wind speed.

Generate probabilistic change in wind speed for every node by using mu and sigma from the historical data set.

Vi = mu(2) + sigma(2) .* randn(ens_size,nodes_count);

Add up probabilistic changes in wind speed.

vi_cum = cumsum(vi(ens_i,:));

Sum up initial wind speed and following changes in wind speed.

V = tc_track(track_i).MaxSustainedWind + vi_cum;

Round to 5 kn.

v = round(v /5)*5;

Figure 37 shows the histograms based on the probabilistic wind speed (bars) and based on the historical track data (black dotted lines) for available cyclone energy, number of tropical storms, number of hurricanes and major hurricanes.

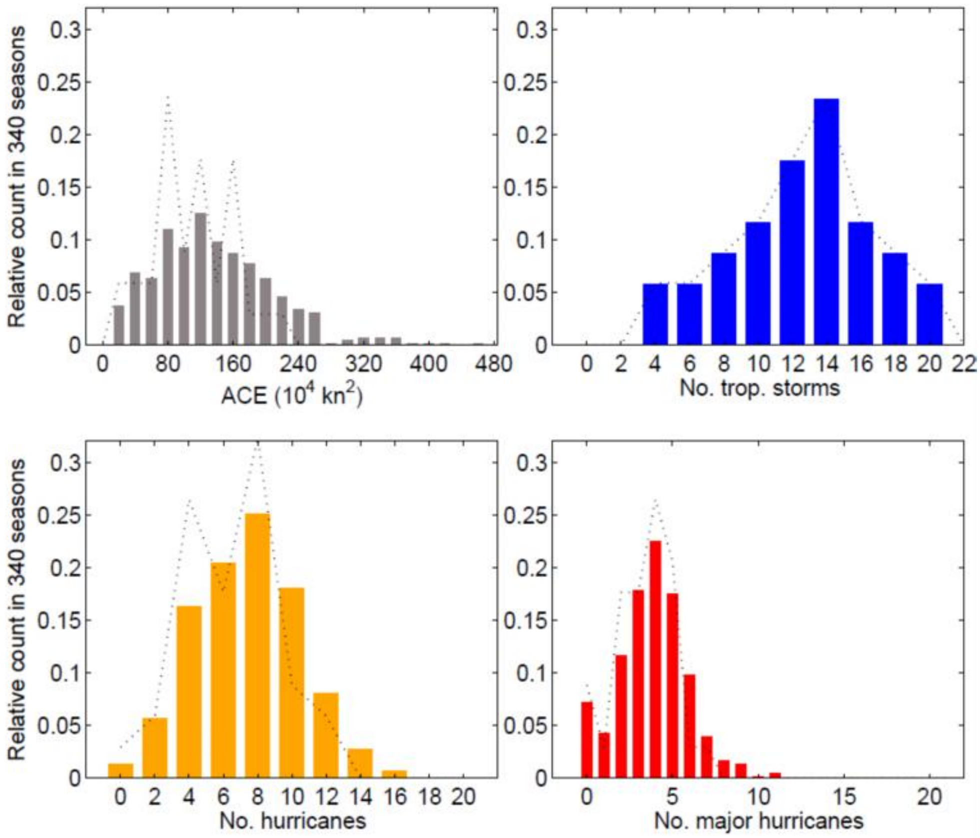


Figure 37: Histograms of available cyclone energy (ACE) (top left), number of tropical storms per season (top right), number of hurricanes per season (bottom left) and number of major hurricanes per season (bottom right). The histograms are based on 4480 probabilistic tracks. The dotted lines indicate the histograms based on the historical tracks.

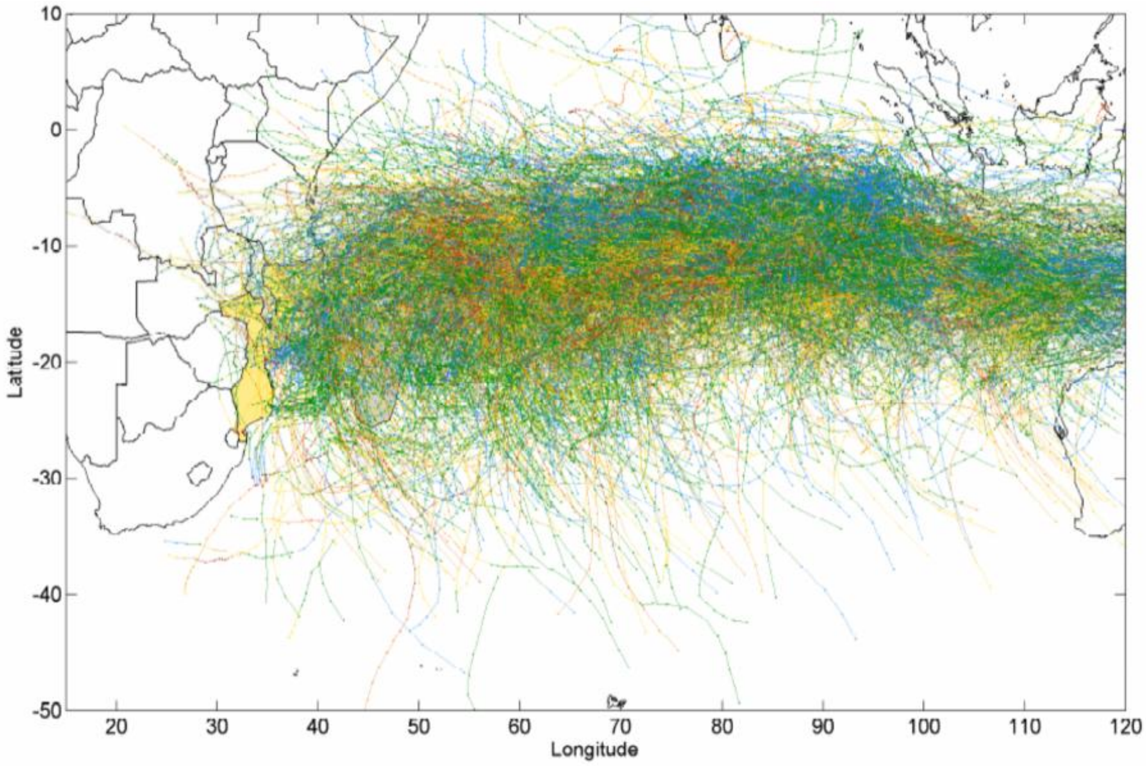


Figure 38: Map view of all 4480 probabilistic generated tropical cyclone tracks in the south western Indian Ocean based on448 track records.

Appendix D - Visualisation in Google Earth

1. Visualize Tropical Cyclone Tracks

The Google Earth toolbox is located in ...\climada\code\googleearth and this path is added in the startup file

```
addpath([climada_root_dir filesep 'code' filesep 'googleearth']);
```

MATLAB call:

climada_tc_track_google_earth (tc_track, add_circles, google_earth_save)

climada_tc_track_google_earth

The function creates a kml file with the visualisation of historical tracks with time stamp in google earth.

Kml file saved in
...\climada\data\tc_tracks\tc_track....kml

It takes quite some time for a bunch of tracks and the kml file is heavy. It can be zipped and renamed to .kmz which can be doubled clicked and displayed in google earth.

Figure 39: Visualization of tropical cyclone tracks in google earth.

2. Visualize Windfield of Specific Tropical Cyclone Track (Animation)

The function creates a kml file with the visualization of the windfield of a specific historical or probabilistic tc track with time stamp in google earth.

MATLAB call:

climada_tc_track_google_earth_windfield
(tc_track,centroids,aggregation,google_earth_save)

climada_tc_track_google_earth_windfield

Kml file saved in
...\climada\data\tc_tracks\tc_track_windfield......kml

The kml file can be zipped and renamed to .kmz which can be doubled clicked and displayed in google earth.

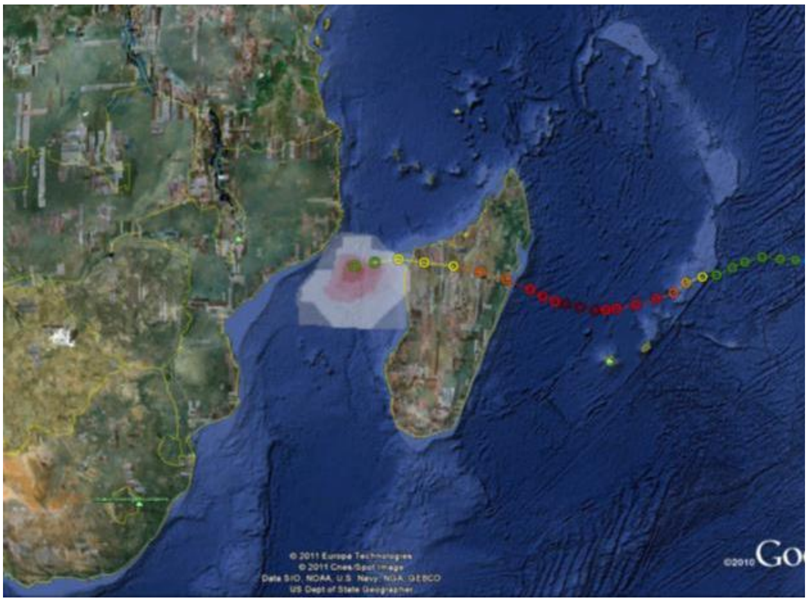


Figure 40: Visualization of wind field at every node for a specific tropical cyclone track.

Appendix E - Centroid Specific Wind Return Periods and Fitted Gumbel Distribution

1. Histogram of probabilistic and historical wind speed at one specific location (centroid)

The function plots the histogram of historical and probabilistic wind speed at one specific centroid.

MATLAB call:

climada_plot_histogram (hazard, centroids, important_centroid, check_printplot)

climada_plot_histogram

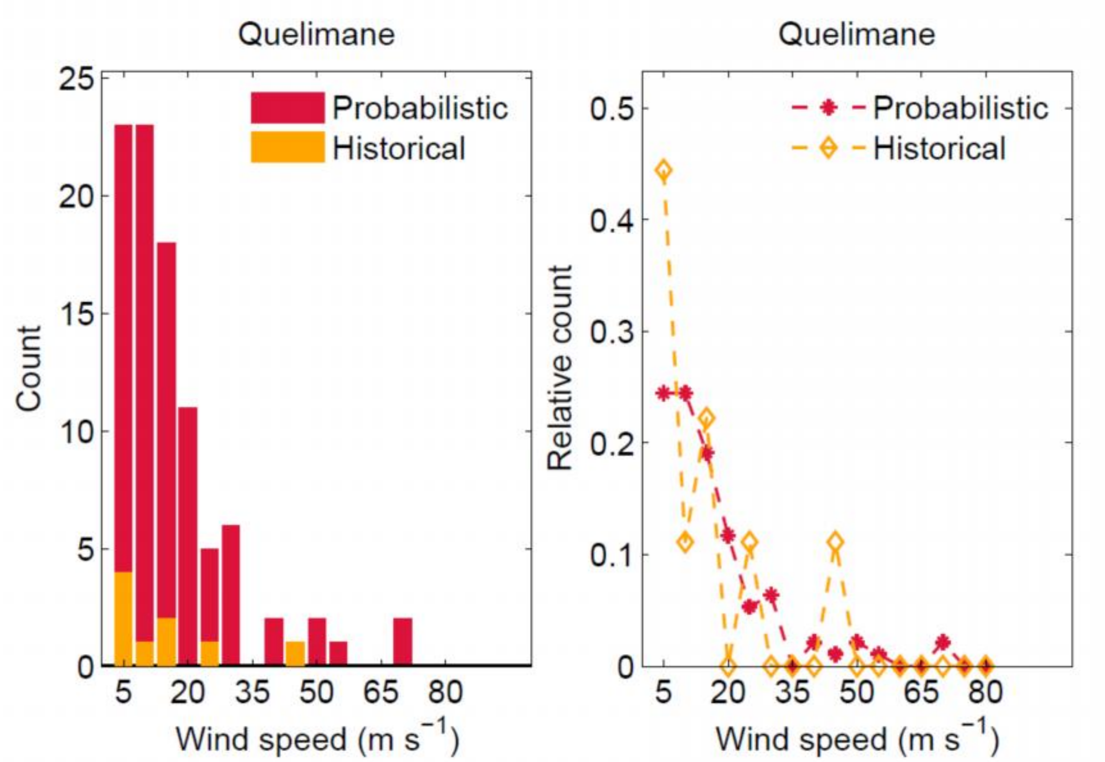


Figure 41: Histogram of probabilistic and historical wind speed at one specific centroid. Left shows the absolute count, right shows the relative count.

Figure can be saved in

...\histogram_centroidNo.pdf or
...\histogram_Quelimane.pdf

2. Plot Wind Speed vs. Frequency at one or Multiple Specific Centroids

The function plots historical and probabilistic wind speed at one or multiple specific centroids vs. frequency.

MATLAB call:

climada_plot_IFC (hazard, centroids, important_centroid, check_printplot)

climada_plot_IFC

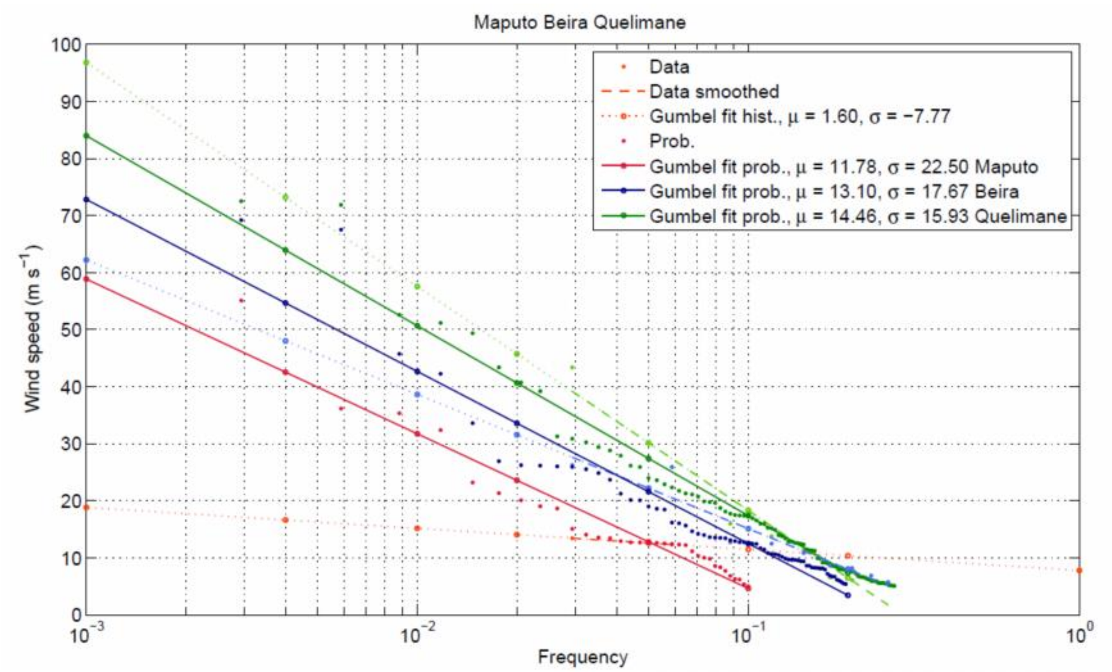


Figure 42: Wind speed vs. frequency at the three centroids Maputo, Beira and Quelimane. Shown for each centroid are the historical data, smoothed data, gumbel fit of historical data, probabilistic data and gumbel fit for probabilistic data.

3. Plot Wind Speed vs. Return Period at one or Multiple Specific Centroids

The function plots historical and probabilistic wind speed at one or multiple specific centroids vs. return period.

MATLAB call:

climada_plot_IFC_return (hazard, centroids, important_centroid, check_printplot)

climada_plot_IFC_return

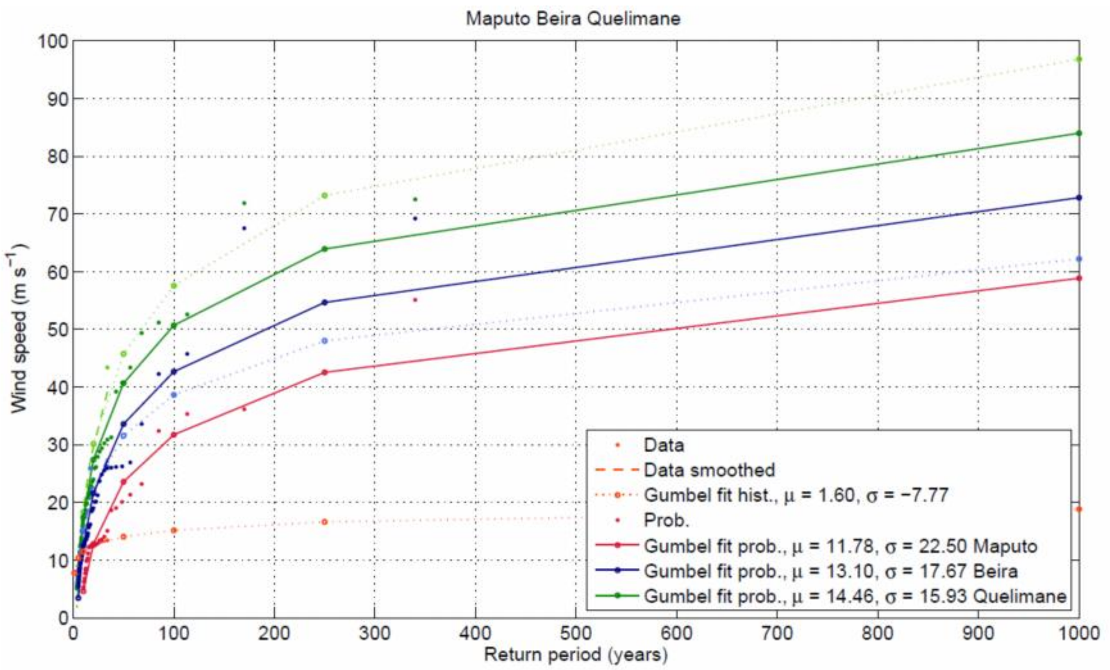


Figure 43: Wind speed vs. return period at the three centroids Maputo, Beira and Quelimane. Shown for each centroid are the historical data, smoothed data, gumbel fit of historical data, probabilistic data and gumbel fit for probabilistic data.