CLIMADA MANUAL

20 Dec 2014

Obtain the full package from https://github.com/davidnbresch/climada
David Bresch, david.bresch@gmail.com
Lea Mueller, muellele@gmail.com

Uncertainty and risk of climate change: from probabilistic damage calculation to the economics of climate adaptation – shaping climate resilient development[1].

Instead of an Introduction: 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?

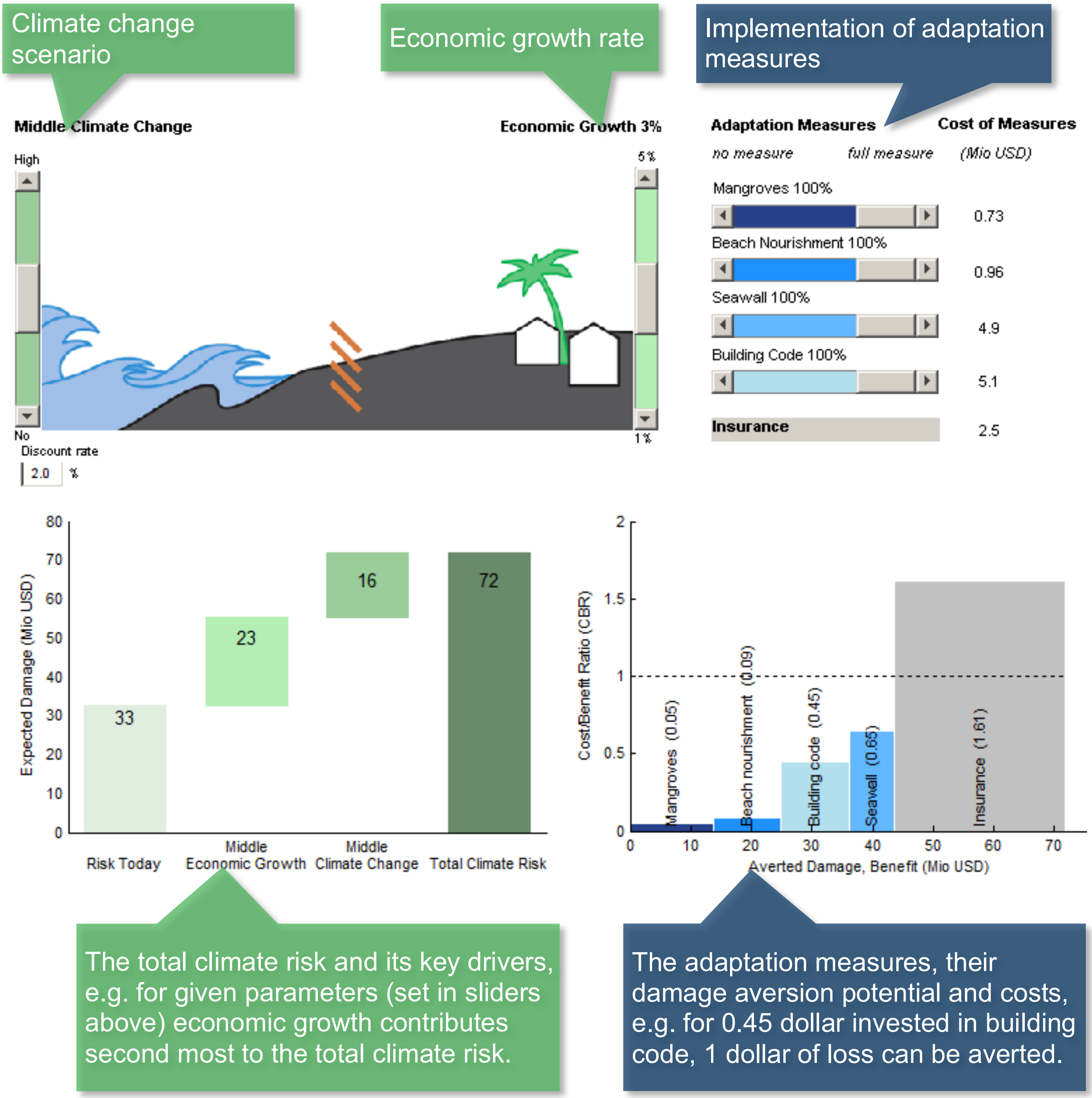
The economics of climate adaptation methodology as implemented in climada provides decision-makers with a fact base to answering these questions in a systematic way. It enables them to understand the impact of climate on their economies – and identify actions to minimize that impact at the lowest cost 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 managing total climate risk.

climada consists of the core module, providing the user with the key functionality to perform an economics of climate adaptation assessment. Additional modules implement global coverage (automatic asset generation), a series of hazards (tropical cyclone, surge, rain, European winter storms, … and even earthquake and meteorites) and further functionality, such as Google Earth access, animations…

[1] See lecture course at the Swiss Federal Institute of Technology (ETH): www.iac.ethz.ch/edu/courses/master/modules/climate_risk

Contents

A visual primer



Climate change scenario

Economic growth rate

Implementation of adaptation measures

Middle Climate Change

Economic Growth 3%

Discount rate 2.0 %

Adaptation Measures

Cost of Measures (Mio USD)

Mangroves 100% 0.73

Beach Nourishment 100% 0.96

Seawall 100% 4.9

Building Code 100% 5.1

Insurance 2.5

The total climate risk and its key drivers, e.g. for given parameters (set in sliders above) economic growth contributes second most to the total climate risk.

The adaptation measures, their damage aversion potential and costs, e.g. for 0.45 dollar invested in building code, 1 dollar of loss can be averted.

Figure 1: The demonstration code `climada_demo` implements the concept of total climate risk and cost-effective adaptation in an interactive way: The user can experiment with key 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).

A brief introduction to the concepts behind climada

Instead of studying this now, the user might also jump to the step-by-step introduction below and later come back.

Probabilistic damage model

A model is nothing more than a simplified representation of reality. Natural hazard models use the virtual world of computers in an attempt to simulate natural catastrophe damage expected in reality. The risk of natural depends on three basic sets of data, which must be fed into the damage model. They are:

- Hazard (sometimes also called peril): Where, how often and with what intensity do events occur?
- Damage function (sometimes referred to as vulnerability or vulnerability curve): What is the extent of damage at a given event intensity?
- Assets (also referred to as value distribution or portfolio of exposed assets): Where are the various types of insured objects located and how high is their value?

These three building blocks are quantified separately and are then combined in the process of estimating event damage. This approach may generally be applied to all forms of natural hazard, whether storm, flood or any other type of peril.

The simplest way to assess the damage is to simulate an individual natural catastrophe scenario. This is known as “deterministic” or “scenario-based” modeling. Such models often refer back to major historical damage events, applying these to the assets that exist now (“as-if analysis”). The disadvantage of this method is that, whilst it allows a single, extreme, individual event damage to be assessed, it fails to take account of all the other events that might occur. It is not possible to calculate an expected annual damage for a portfolio of assets on the basis of single event damage, and any prediction as to the occurrence frequency of the model scenario will remain very uncertain.

Today, in an attempt to avoid these problems, so-called “probabilistic” models are being used to assess hazards such as storms and floods. Rather than simply analyzing one event, the computer is programmed to function as a sort of time-lapse film camera, simulating all the possible events that could unfold within a sufficiently long period of time (thousands or tens of thousands of years). This type of model produces a “representative” list of event damages (i.e. a list that accurately reflects the risk). From this list it is possible to understand the relationship between damage potential and occurrence frequency, and hence the cost of average and extreme damage burdens.

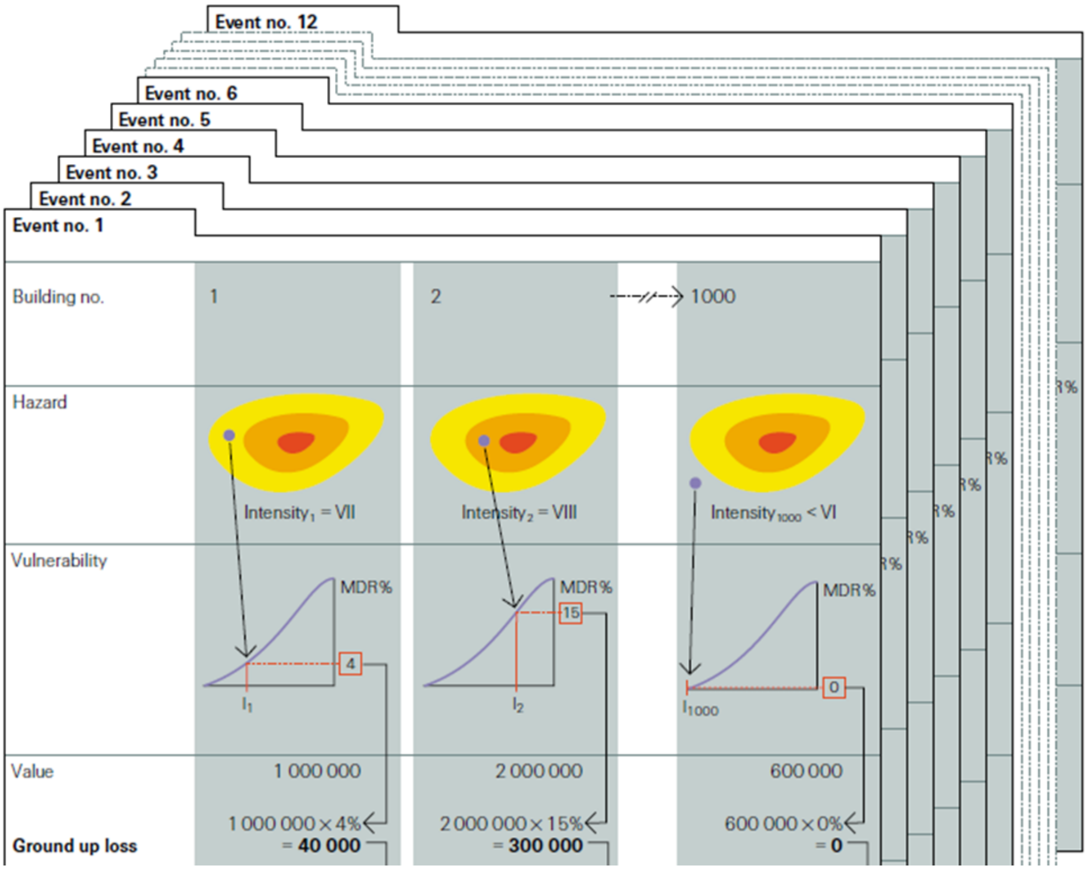


Figure: Using risk assessment tools to calculate event damage. Let's assume a hypothetical portfolio containing 1000 assets (buildings). For the sake of simplicity, let us assume that the risk assessment tool only contains 12 potential events over a projected period of 200 years. The following calculations would be performed:

- The hazard module generates the expected intensity (VII) for event no.1 at asset (building) location no.1.
- The damage function (called vulnerability in the figure) corresponding to the asset provides us with the mean damage ratio (MDR[2]) for given hazard intensity (4 stands for 4% of the asset's value)
- The damage is calculated by multiplying the MDR and the value of the asset (1'000'000), resulting in a (ground up) damage (called loss in the figure) of 40'000.
- Above steps are performed on all 1'000 assets in the portfolio. The sum of all damages produces the total damage from event no.1, i.e. event damage no1.
- All above steps are then repeated for the other (11) events in the event set.
- Upon completion of all these stages in the modeling process, a list of all event damage is produced, upon which damage statistics can be derived (average damage, max damage…).

See mentioned lecture course and the Swiss Re publication "Natural catastrophes and reinsurance", which covers the methodology in detail: http://media.swissre.com/documents/Nat_Cat_reins_en.pdf

[2] Please note that climada uses MDR=MDD*PAA, where Mean damage degree (MDD) and percentage of affected assets (PAA) allow to deal with local deductibles in a more appropriate form than a simple Mean damage ration (MDR) model could do, since one does, due to the PAA, know how many assets are affected, hence deductible application is more specifc.

Adaptation cost curve

Assessing the cost and damage aversion potential of each measure can be quite difficult. The potential damage aversion is particularly uncertain, even for measures for which extensive research exists – for example, for building codes to fix roofs against hurricane winds.

The assembled cost curve shows – from left to right – the range of measures from least cost-efficient to most cost-efficient. The results should thus be used to start discussions on the different measures and the opportunity to avert expected damage, rather than be read as recommendations to implement certain measures.

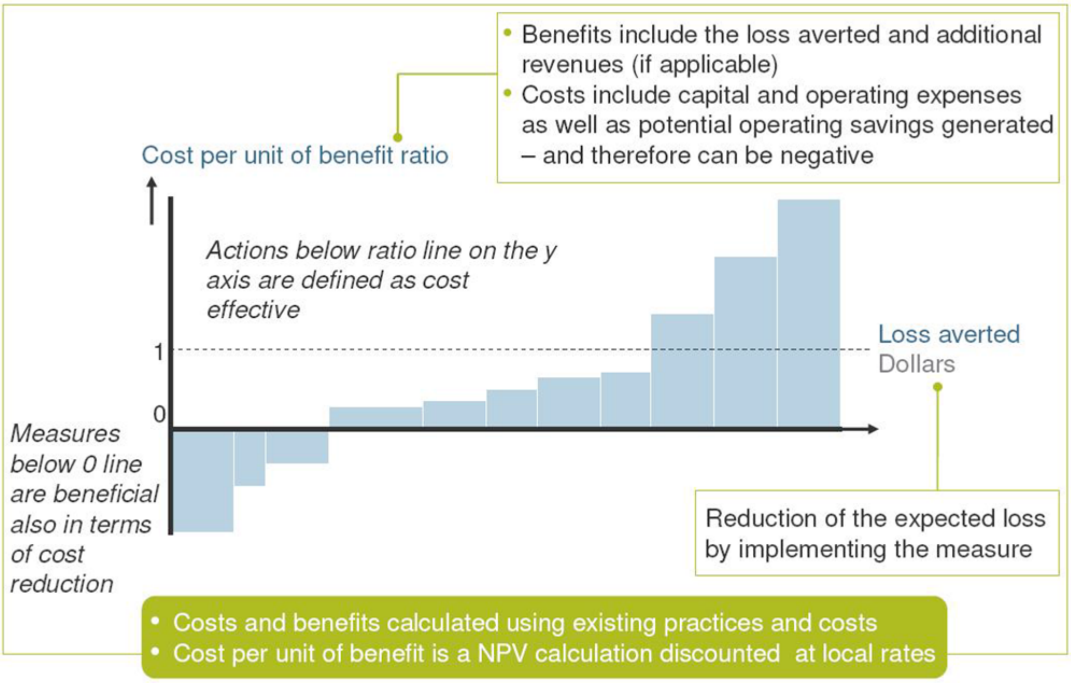


Figure: The width of each bar in a cost curve represents the cumulative potential of that measure to reduce total expected damage up to 2030 for a given scenario. The height of each bar represents the ratio between costs and benefits for that measure. Whether or not this ratio is attractive to a decision maker depends on many factors, including risk appetite. After considering the other – including non-economic – impacts and benefits related to implementing a measure, a risk-neutral decision maker would select measures based on a sense of how much protection they offer and at what cost. The advantage of calculating cost-benefit ratios for all measures is that doing so allows decision-makers to compare measures using a single simple metric.

In a recipe form, the adaptation cost curve is constructed as follows (repeat for each measure)

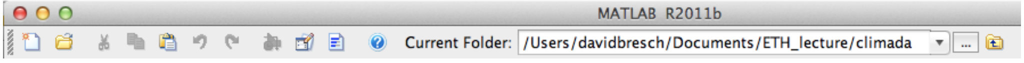
1. Calculate present value (PV) of costs of measure [e.g. Excel, outside of climada]
2. Risk today: import today's assets and damage functions (input via Excel) and expose them to present hazard (part of climada)
   2.1. climada calculates annual expected damage with no measures
   2.2. climada calculates annual expected damage with measure applied
   → difference 2.1) minus 2.2) shows benefit of measure today
3. Future risk (e.g. year 2030): import future assets and damage functions (input via Excel, damage functions likely to be unchanged) and expose them to future hazard (part of climada)
   3.1. climada calculates annual expected damage with no measures

3.2. climada calculates annual expected damage with measure applied
→ difference 3.1) minus 3.2) shows future benefit of measure

4. climada discounts benefits --> horizontal axis of adaptation cost curve
5. climada calculates the cost benefit ratio → vertical axis of adaptation cost curve

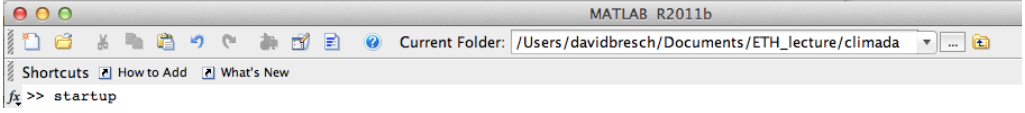
Getting started

- Set the MATLAB Current Folder to climada[3] (use the ... button to browse), e.g.:



Current Folder: /Users/davidbresch/Documents/ETH_lecture/climada

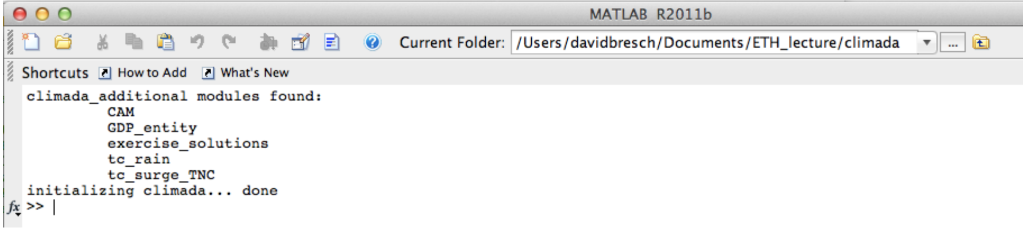
- Enter startup in the MATLAB Command Window:



Current Folder: /Users/davidbresch/Documents/ETH_lecture/climada

```
>> startup
```

  and press Enter (or Return). This initializes climada, sets some variables (e.g. the location of the data folder[4]) and detects any climada_additional modules[5]. After that, the Command Window looks something like:



Current Folder: /Users/davidbresch/Documents/ETH_lecture/climada

```
climada_additional modules found:
         CAM
         GDP_entity
         exercise_solutions
         tc_rain
         tc_surge_TNC
initializing climada... done
>> |
```

  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[6], which is also the best way to test whether climada works properly – you should see something as shown above as a visual primer (see above) 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[7] - or play at least a bit with `climada_demo` ;-)

While the standard climada setup contains the data folder within climada, one can also create a folder named climada_data parallel to climada to allow for core climada data NOT being synched (only the folder ../climada/data within core climada gets synched). This way, any data used in climada beyond the default files will not be synchronised.

In order to grant core climada access to additional modules (see https://github.com/davidnbresch), create a folder 'climada_modules' on the same level as the core climada folder to store any additional modules. This way, climada sources all modules' code upon startup.

[3] Usually the folder you extracted climada.zip to

[4] 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.

[5] 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.

[6] From now on, just type any command in `Courier` in the MATAB Command Window, as we will not state this each time again.

[7] www.iac.ethz.ch/education/master/climate_risk

Process on one page

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 hazard event set
    a. Generate a hazard event set for today's climate
        i. Obtain historical events
        ii. Produce the probabilistic events
        iii. Store intensities at centroids
    b. Repeat above steps for future hazard
    (climate change impact scenarios, e.g. for 2030)

2. Import a list of assets and corresponding damage functions[8]
(the so-called entity)
    a. Read the list of today's assets
        i. Encode to centroids
        ii. Read the damage functions and make sure they correspond to assets
    b. Repeat above steps for future assets (e.g. 2030)

3. Import the list of adaptation measures
(also stored into the entity structure)
    a. Read the list of measures

4. Calculate the damages and benefits of measures
    a. Calculate the damages[9] 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.

In the next section, we are going to illustrate this whole process ste-by-step, using tropical cyclone as the hazard and a few assets in South Florida for illustration purposes. Note already here that climada provided global coverage for tropical cyclone wind (often referred to as TC wind[10]) and storm surge (often referred to as TC surge[11]) as well as other hazards, such as global earthquake[12].

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

[9] In essence, we calculate $\text{damage}_{j,k}=\text{value}_k * f(\text{intensity}_{j,k})$, where value_k ist he value of asset k and $\text{intensity}_{j,k}$ the hazard intensity of event j at location of asset k. f denotes the damage function, i.e. the relation between the hazard intensity and the resulting damage (as a fraction of the asset value). See further below for some more details on the damage calculation ;-)

[10] Part of climada core module (i.e. the module this manual is part of)

[11] Obtain it from https://github.com/davidnbresch/climada_module_tc_surge and see https://github.com/davidnbresch/climada_module_tc_surge/blob/master/docs/climada_module_tc_surge.pdf for this module's manual.

[12] See https://github.com/davidnbresch/climada_module_eq_global

From tropical cyclone hazard generation to the adaptation cost curve[13]

Instead of starting with a simple hazard set generation, the user might also jump to the damage calculation right away, see second next section "Assets and damage functions".

Hazard set

First, obtain the historic tracks[14], i.e. define the name and location of the raw text file with historical tropical cyclone tracks

```
tc_track_file=[climada_global.data_dir filesep ...
        'tc_tracks' filesep 'tracks.atl.txt'];
tc_track=climada_tc_read_unisys_database(tc_track_file);
```

`tc_track(i)` contains position `tc_track(i).lon(j)` and `tc_track(i).lat(j)` for each timestep j as well as the corresponding intensity `tc_track(i).MaxSustainedWind`. E.g. track number 1170 is hurricane Andrew:

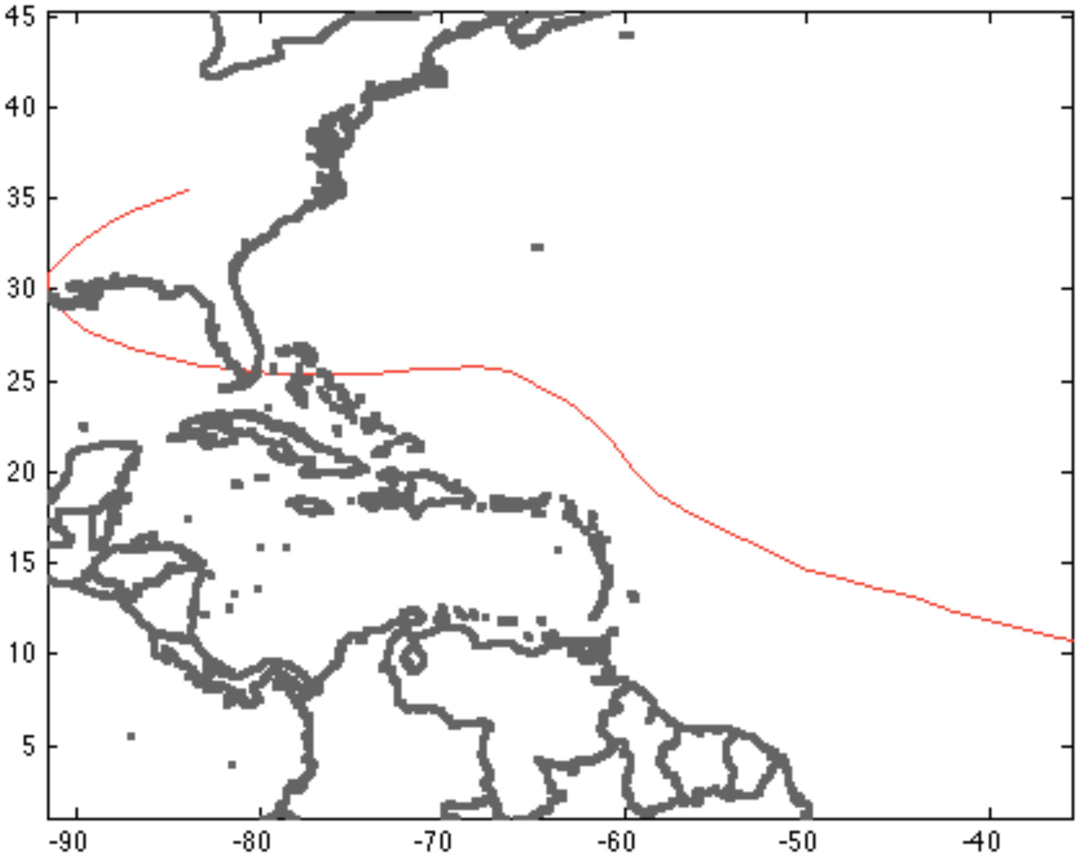


Figure: plot(tc_track(1170).lon,tc_track(1170).lat,'-r'); hold on; set(gcf,'Color',[1 1 1]); axis equal climada_plot_world_borders(2,'','',1) % plot world borders (for orinentation)

[13] See the climada code `climada_demo_step_by_step` which performs all the steps and illustrates the intermediate results by plots, just as shown here.

[14] See the function `climada_tc_get_unisys_databases` to automatically download all databasesfrom the internet (from weather.unisys.com/hurricane)

In order to calculate the windfield of this particular single track, we first generate a series of points on which to evaluate the windfield, we call these points centroids:

```
centroids.Longitude=[];centroids.Latitude=[]; % init
next_centroid=1; % ugly code, but explicit
for i=1:10
    for j=1:10
        centroids.Longitude(next_centroid)=i+(-85);
        centroids.Latitude(next_centroid) =j+  20;
        next_centroid=next_centroid+1;
    end % j
end % i
centroids.centroid_ID=1:length(centroids.Latitude);
```

Next, calculate the windfield[15] for a single track (Andrew again) as

```
res = climada_tc_windfield(tc_track(1170),centroids);
```

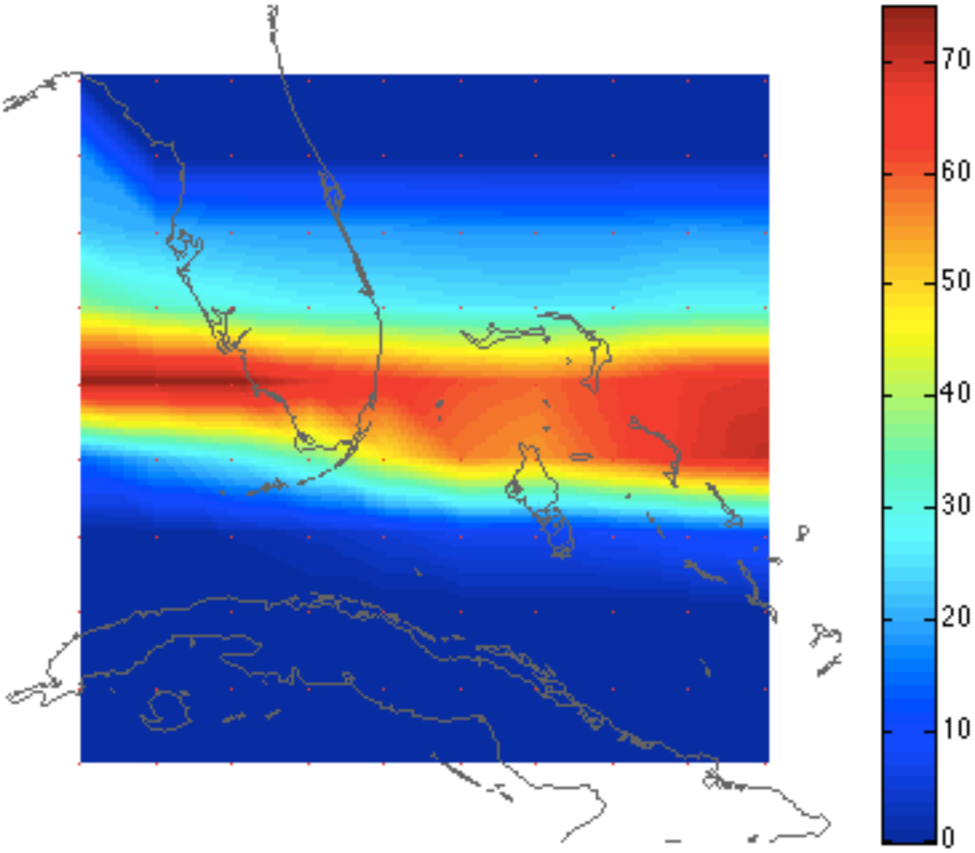


Figure: `climada_color_plot(res.gust,res.lon,res.lat); % plot the windfield`

We now generate the windfield not for one single hurricane, but for all events and store them in an organized way, the so-called hazard event set:

[15] We implement a windfield according to Holland, G. J., 1980: An analytic model of the wind and pressure profiles in hurricanes. Monthly Weather Review, 108, 1212-1218. In addition to the axisymmetric vortex, we take forward speed into account. See also Holland, G. J., 2008: A Revised Hurricane Pressure–Wind Model, Monthly Weather Review, 136, 3432-3445. A natural next step would be the consideration of roughness (not implemented), see e.g. Vickery, P.J. et al., 2009: A Hurricane Boundary Layer and Wind Field Model for Use in Engineering Applications. J. Appl. Meteor. Clim.

```
hazard_set_file=[climada_global.data_dir filesep ...
       'hazards' filesep 'atl_hist'];
hazard = climada_tc_hazard_set(tc_track,hazard_set_file,centroids);
```

And now this hazard event set contains the single Andrew windfield we generated before in hazard.intensity(1170,:) and therefore we can reproduce the same windfield with the following command (note the full(*), as we store a sparse matrix)

```
climada_color_plot(full(hazard.intensity(1170,:)),...
       hazard.lon,hazard.lat,'none')
```

Or, instead, we can plot all hazard intensities at a given point (greencircle) like

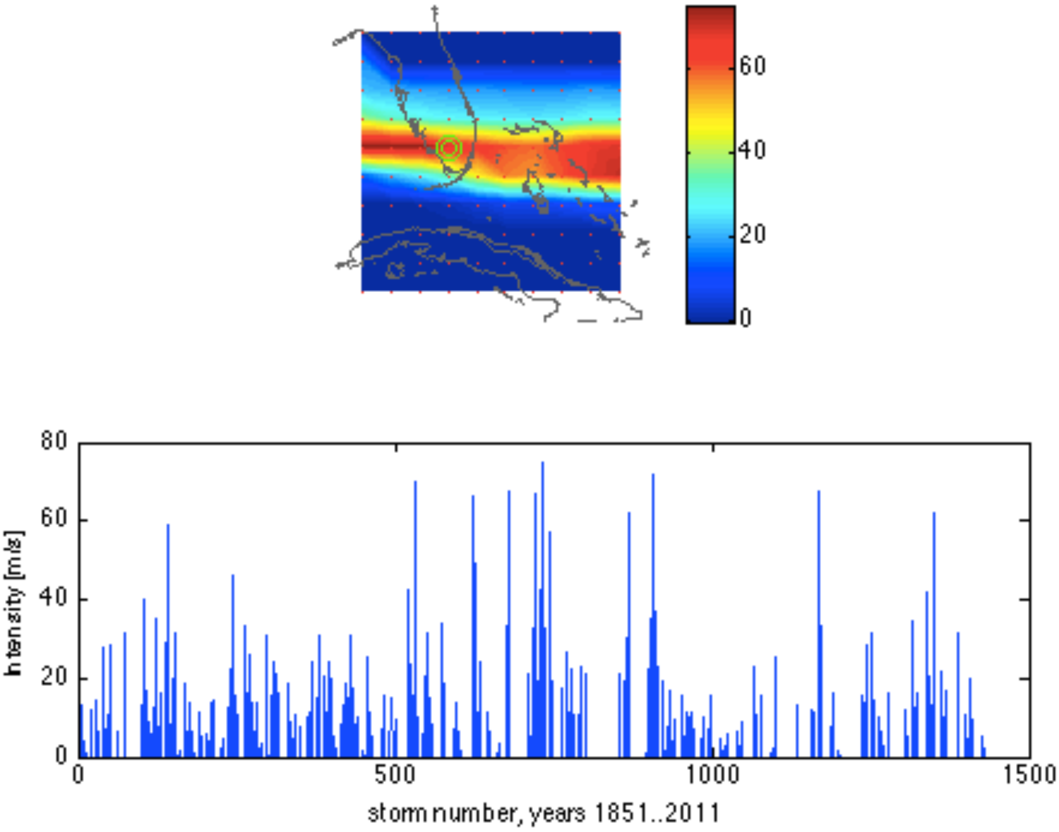


Figure:
```
figure; subplot(2,1,1)
climada_color_plot(full(hazard.intensity(1170,:)),...
hazard.lon,hazard.lat,'none'); hold on;plot(-81,26,'Og');
plot(centroids.Longitude(36),centroids.Latitude(36),'Og','MarkerSize',10);
subplot(2,1,2)
plot(full(hazard.intensity(:,36))); set(gcf,'Color',[1 1 1]);
xlabel(sprintf('storm number, years
%i..%i',tc_track(1).yyyy(1),tc_track(end).yyyy(end)))
ylabel('Intensity [m/s]')
```

Instead of only historic tracks, we can generate artificial or probabilistic tracks, simply by 'wiggling' the original tracks, eg for Andrew 1992 again:

```
tc_track_prob=climada_tc_random_walk(tc_track(1170));
```

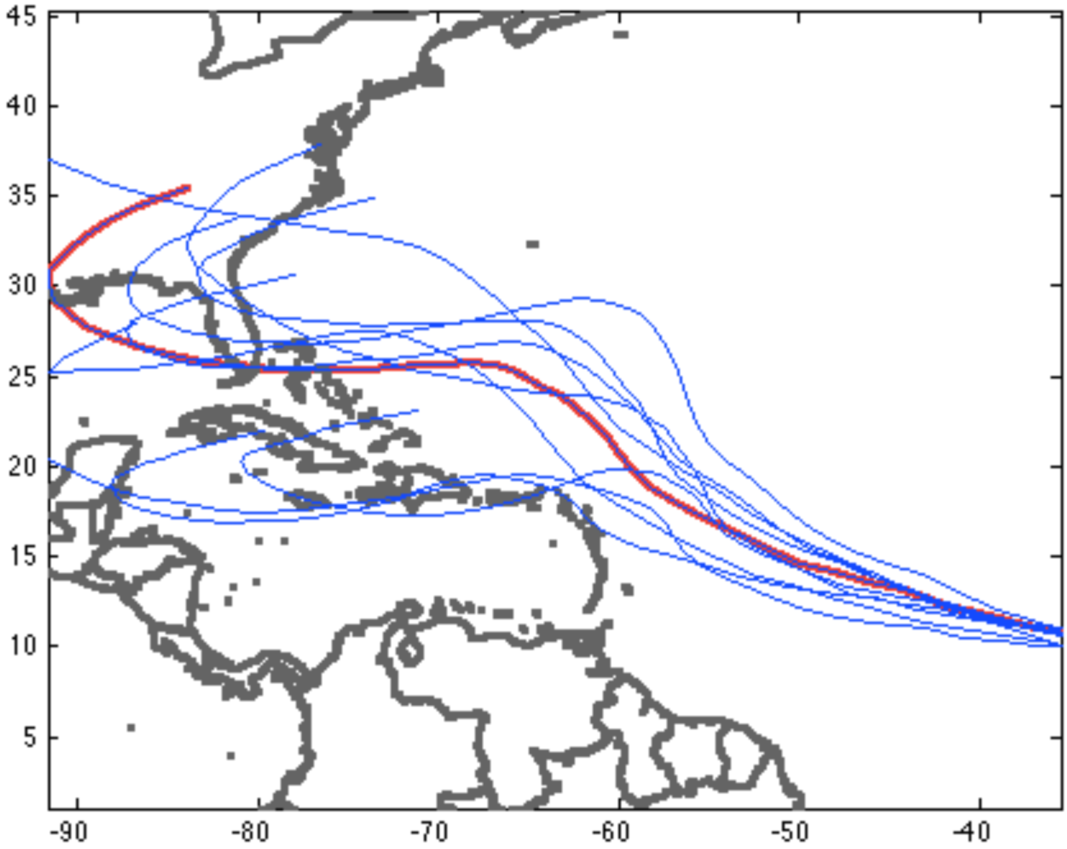


Figure: `plot(tc_track(1170).lon,tc_track(1170).lat,'-r','LineWidth',2);`

```
hold on; set(gcf,'Color',[1 1 1]); axis equal
climada_plot_world_borders(2,'','',1)
for track_i=1:length(tc_track_prob)
       plot(tc_track_prob(track_i).lon,tc_track_prob(track_i).lat,'-b');
end
```

And repeated for all historic tracks, we obtain the full probabilistic track set

```
climada_global.waitbar=0; % switch waitbar off, speeds up,
% hence the next line will take approx. 3 sec
tc_track_prob=climada_tc_random_walk(tc_track);
```

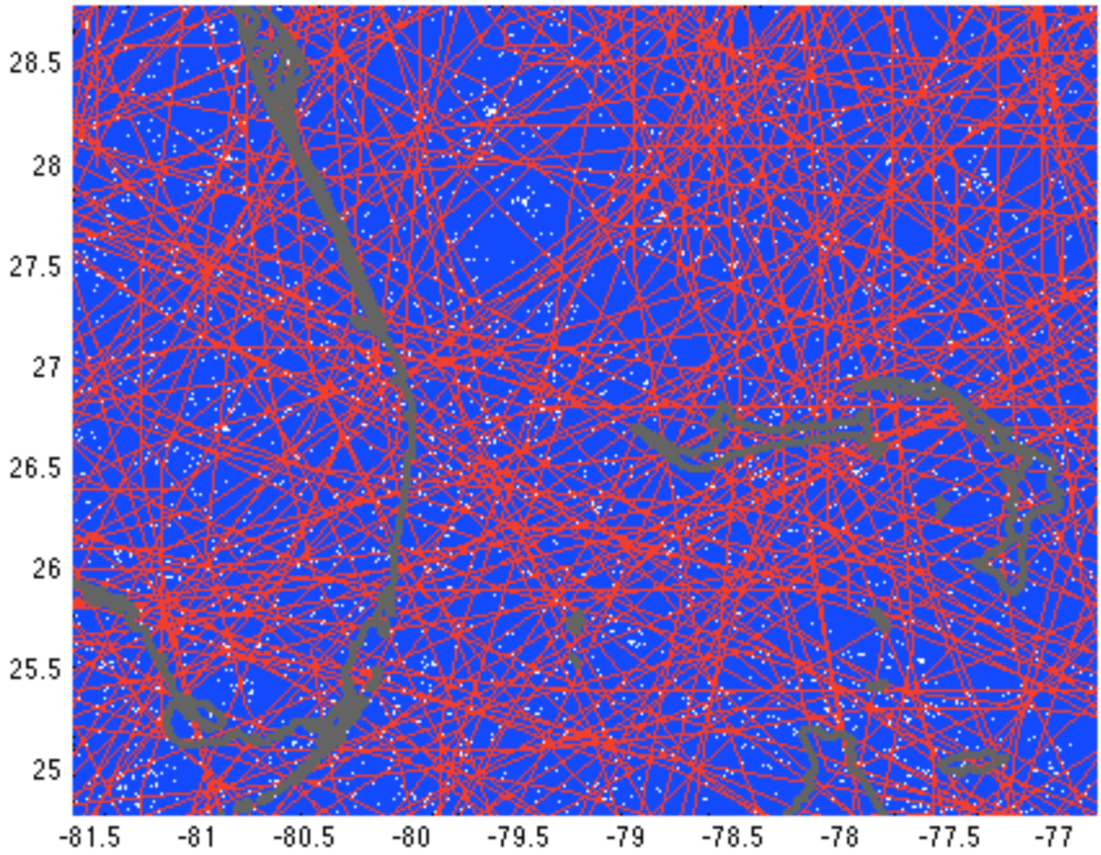


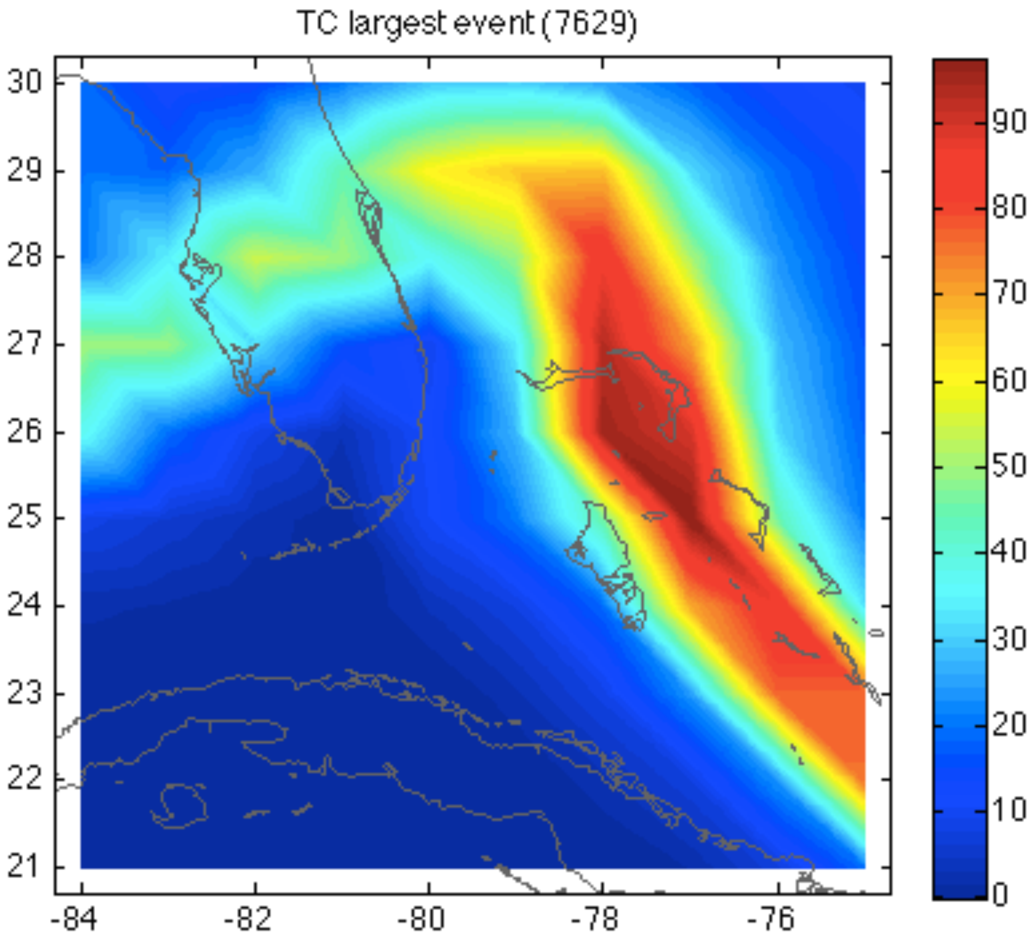
Figure (manually zoomed in):

```
for track_i=1:length(tc_track_prob)
      plot(tc_track_prob(track_i).lon,tc_track_prob(track_i).lat,'-b');
      hold on;end
for track_i=1:length(tc_track)
      plot(tc_track(track_i).lon,tc_track(track_i).lat,'-r');end
climada_plot_world_borders(2,'','',1); set(gcf,'Color',[1 1 1]);
```

Next, we generate the windfields for all 14450 probabilistic tracks (takes a bit less than 2 min)

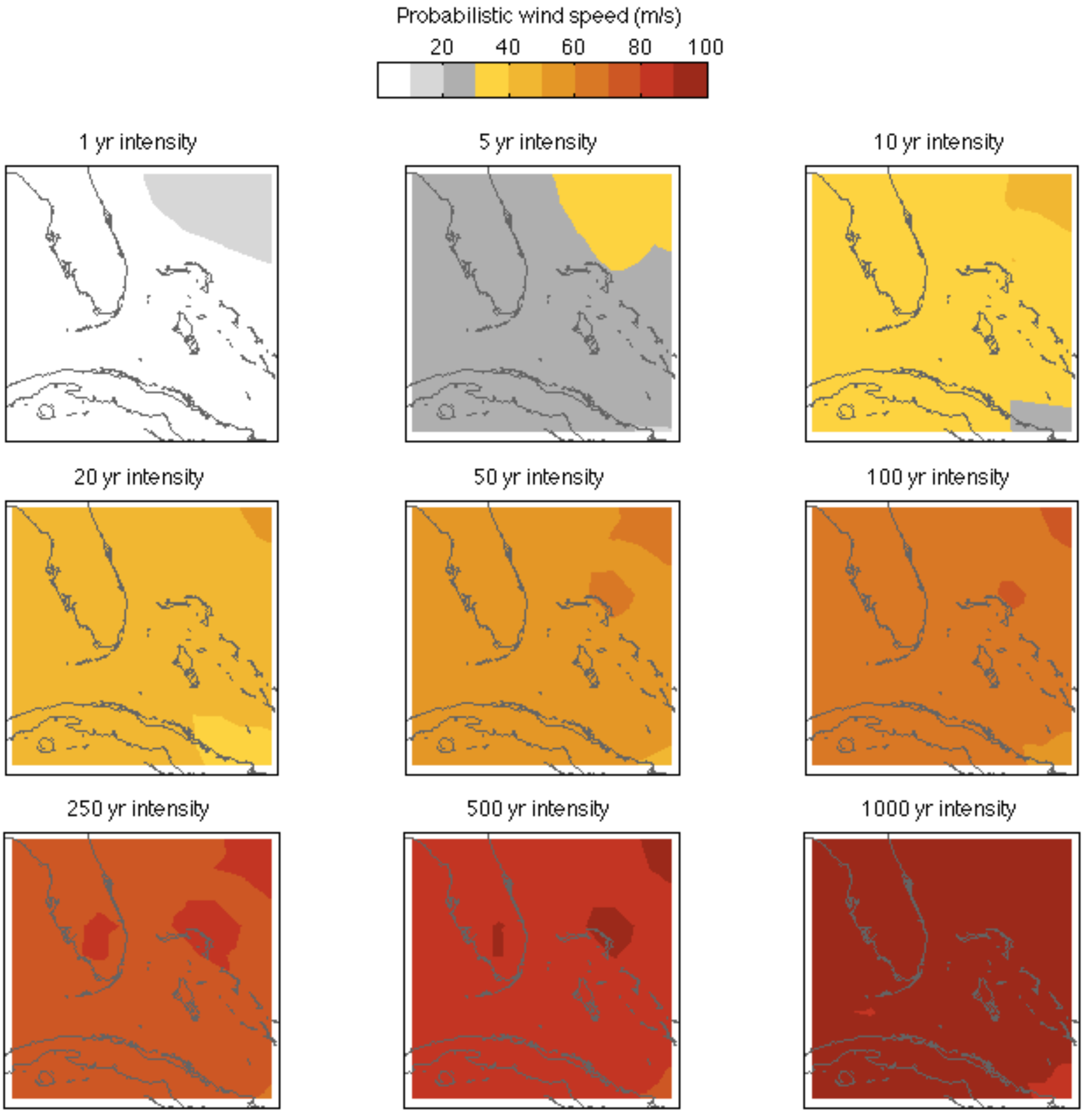
```
hazard_set_file=[climada_global.data_dir filesep ...
      'hazards' filesep 'atl_prob.mat'];
hazard = ...
      climada_tc_hazard_set(tc_track_prob,hazard_set_file,centroids);
```

The hazard set now contains ten thousands (in fact 14450) tropical cyclone footprints, each stored at all centroids. We can for example plot the largest single event with:



```
figure; climada_hazard_plot(hazard); set(gcf,'Color',[1 1 1]);
```

and generate the windspeed maps for several return periods:



```
climada_hazard_stats(hazard); set(gcf,'Color',[1 1 1]);
```

Before we move on, let's explain the key elements of the hazard structure: `hazard.lon(i)` and `hazard.lat(i)` contain the coordinates of centroid i, hence `hazard.intensity(j,i)` contains the hazard intensity of event j at centroid i. Further `hazard.frequency(j)` contains the single event frequency of event j. These are in fact the key elements of the hazard structure, note that `hazard.intensity` is a sparse array (refer to e.g. `help sparse` in MATLAB[16]). You might refer to functions such as the mentioned `climada_tc_hazard_set` or `climada_excel_hazard_set`[17] to see how a hazard event set is generated.

[16] In essence, a sparse array stores the non-zero elements of an array only. Since a single event hits only a few centroids – especially true for hazard set for a larger geographical region – we save a lot of memory and speed up the calculations substantially.

[17] This function generates a hazard event set based on Excel input. The Excel sheet needs to contain all the event footprints. An easy method to use climada with a finite (small) number of predefined events (more hazard event scenarios then a full probabilistic set). See file ../data/hazards/ Excel_hazard.xls which contains a small example (for Mozambique).

Assets and damage functions

So much for the hazard event set, let's now import an asset base (the small asset example as used in `climada_demo`, the demonstration GUI as shown above[18]). Before we do so, we load the hazard set file as used in `climada_demo`, in order to later reproduce the results:

```
hazard_set_file=[climada_global.data_dir filesep ...
      'hazards' filesep 'TCNA_today_small.mat'];

load(hazard_set_file) % loads hazard as variable
```

and are now in a position to import the Excel file with all the asset information[19]:

```
entity_excel_filename=[climada_global.data_dir filesep ...
'entities' filesep 'demo_today.xls'];

entity=climada_entity_read(entity_excel_filename,hazard)
```

Such an entity structure contains the asset, damage function and adaptation measures information, the tabs in Excel are named accordingly, and so are the elements of the imported structure[20]. In the asset sub-structure, we find[21] `entity.assets.Latitude(k)` and `entity.assets.Longitude(k)`, the geographical position of asset k (does not need to be the same geographic location as centroid I, since assets are encoded to the hazard[22]) `entity.assets.Value(k)` contains the Value of asset k. Please note that Value can be a value of any kind, not necessarily a monetary one, e.g. it could be number of people living in a given place. `entity.assets .DamageFunID(k)` contains a reference ID (integer) to link the specific asset with the corresponding damage function (see Excel tab damagefunctions and entity.damagefunctions). Before we move on the the damagefunctions, note that `entity.assets.centroid_index(k)` contains the centroid index onto which asset k is mapped in the hazard event set[23].

[18] One can also generate assets (value distributions) in climada, see e.g. the climada module https://github.com/davidnbresch/climada_module_country_risk

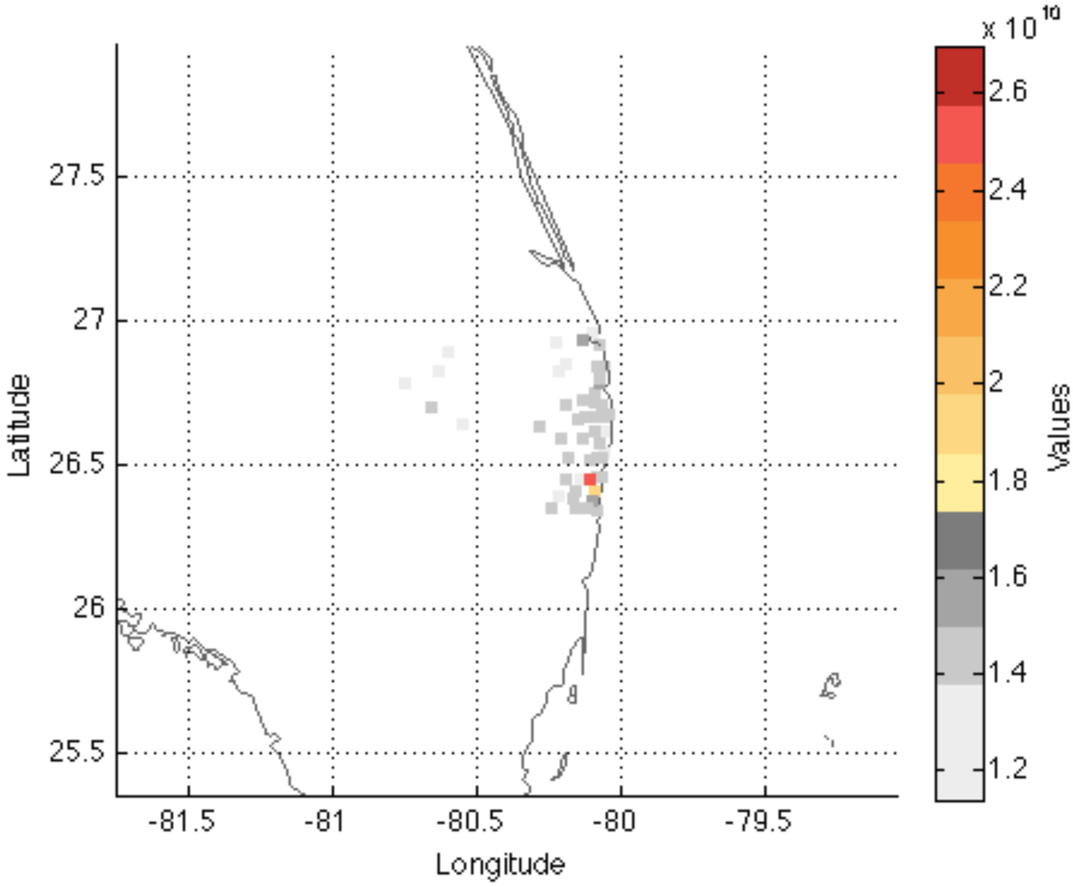
[19] Please have a look at the Excel file, each column header is explained by a small comment (tiny yellow triangel in the upper right corner of the cell).

[20] Please note that we discuss the measures information further below

[21] We focus on the key content here, please inspect the structure in MATLAB yourself.

[22] See function `climada_assets_encode`. Encoding means: map asset positions to calculation centroids oft he hazard event set. This step is required to allow the user to freely specify asset locations, rather than stick tot he centroids the hazard set has been stored at. A beginner-level user should not need to deal with such technical details, though.

[23] As mentioned in a previous footnote, the beginner level user does not need worry too much about, this simply speeds up damage calculation substantially.

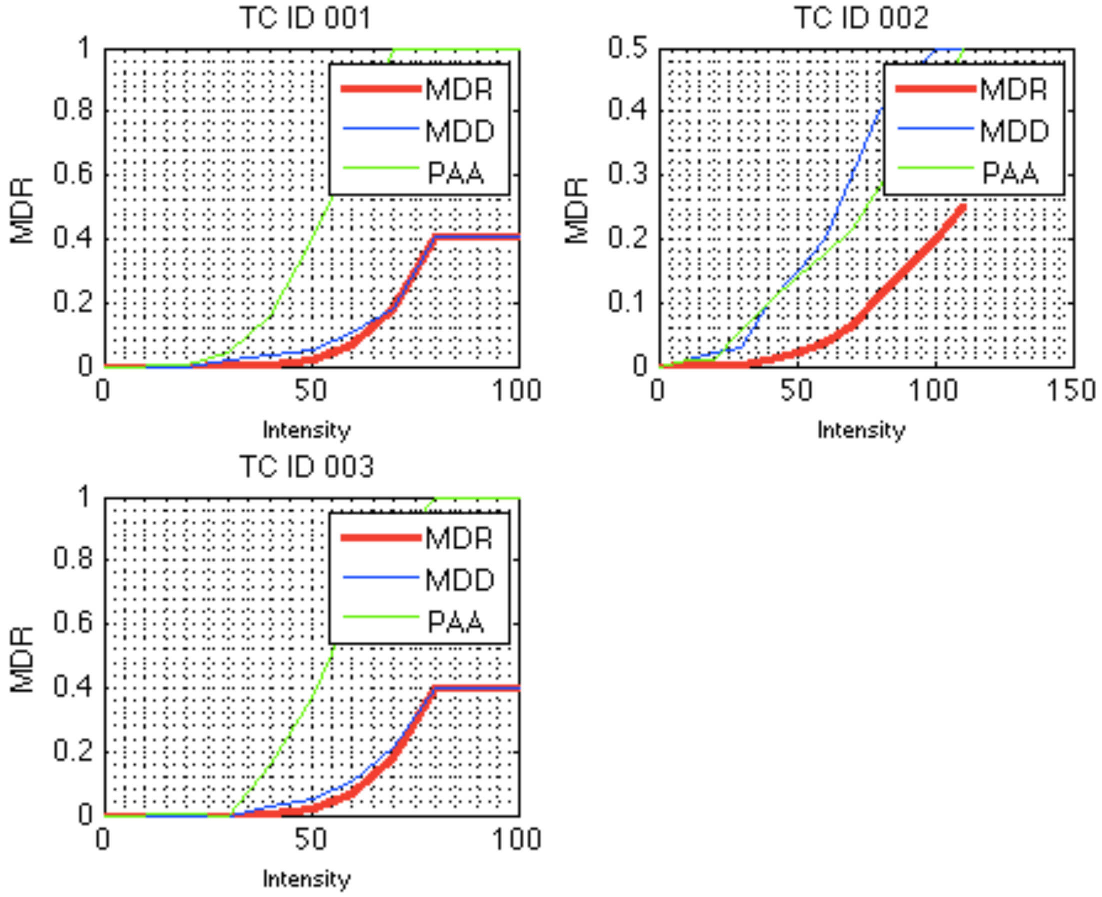


```
figure; climada_entity_plot(entity,4); set(gcf,'Color',[1 1 1])
% the asset distribution as stored in entity (read from Excel sheet)
```

The damagefunctions sub-structure contains all damage function information, i.e. `entity.damagefunctions.DamageFunID` contains the IDs which refers to the asset's DamageFunID. This way, we can provide different damage functions for different (groups or sets of) assets. `entity.damagefunctions.`Intensity contains the hazard intensity, `entity.damagefunctions.`MDD the mean damage degree and `entity.damagefunctions.`PAA the percentage of affected assets. Last but not least, `entity.damagefunctions.`peril_ID contains the peril ID (2-digit character) which allows to indentify specific damage functons with perils. This way, we can in fact use DamageFunID 1 in the assets to link to damage function one, which can exist several times, one for each peril. The damagefunctions are stored in a bit a special format, since we get the first damagefunction as[24]

```
pos=find(entity.damagefunctions.DamageFunID==...
      entity.damagefunctions.DamageFunID(1))
plot(entity.damagefunctions.Intensity(pos),...
      entity.damagefunctions.MDD(pos)) % not shown, see next figure
```

[24] In the case there is only one perilID, see further details in `climada_damagefunctions_plot`



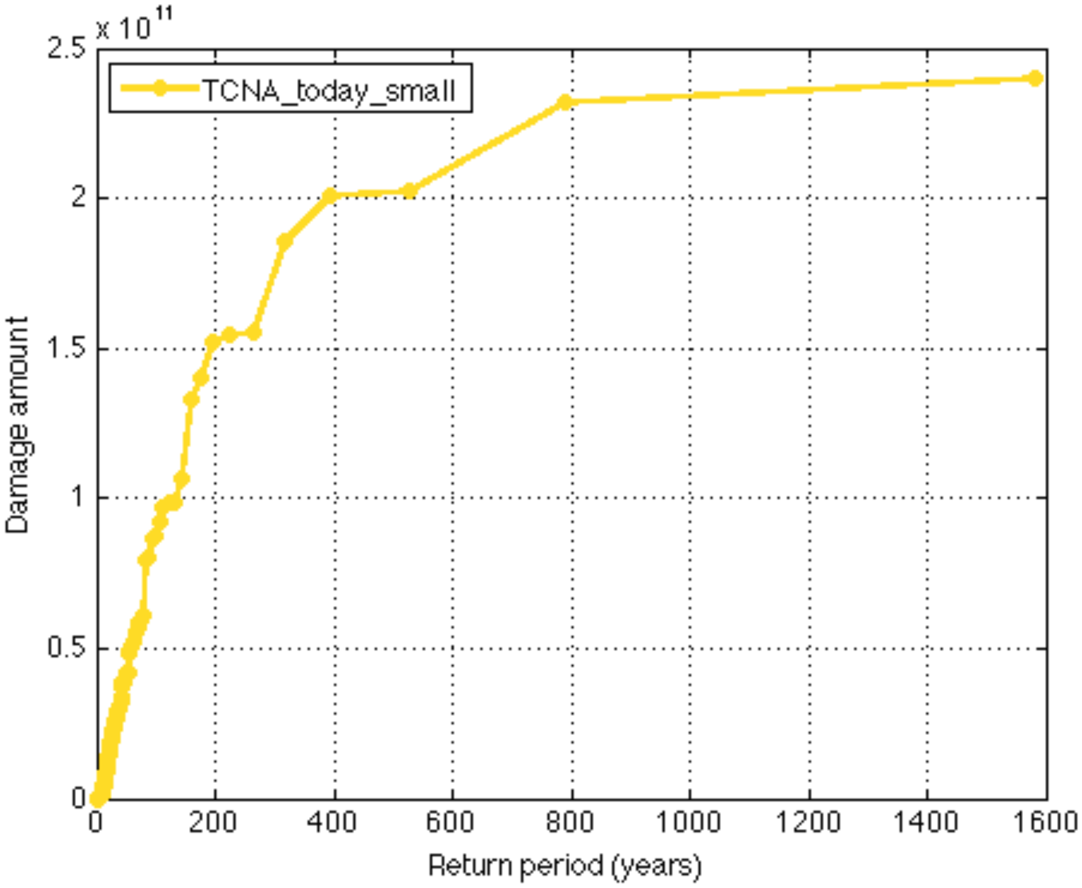
`figure;climada_damagefunctions_plot(entity)`. The three damage functions as defined in the damagefunctions tab of the Excel file. TC is the perilID and stands for tropical cyclone, while 001, 002 and 003 denote the DamageFunID. The horizontal axis denotes the hazard intensity (here tropical cyclone windpseed), the vertical axis is the same for MDD, PAA and MDR.

Damage calculation

And with that, we're ready for the damage calculation, simply as:

```
EDS=climada_EDS_calc(entity,hazard)
```

Where `EDS` contains the event damage set, it contains the annual expected damage in `EDS.ED`, the event damage for event j in `EDS.damage(j)`, the event frequency in `EDS.frequency(j)` and the event ID in `EDS.event_ID(j)`. In futher fields it stores the link to the original assets, the damagefunctions and hazard set used. Instead of plotting the event damage set (here a vector with 14450 elements), one rather refers to the damage excess frequency curve:



```
figure; climada_EDS_DFC(EDS) % show damage excess frequency curve
```

The horizontal axis denotes the return period in years, the vertical axis the damage (in unites the Values were provided, here USD). The label of the curve denotes the hazard set used.

While one would in a proper application of climada now calculate the damages of future assets (to obtain the effect f economic growth) and then further repeat the calculation with a future hazard set (to obtain the effect of climate change), we illustrate the benefit of adaptation measures by simply using the assets and hazard we have already used.

Adaptation cost curve

As mentioned, the entity structure contains not only assets and damagefunctions, it also holds the adaptation measures[25]. `entity.measures.name{m}` contains the name of measure m, `entity.measures.name.cost(m)` the cost[26]. The following fields allow the parameterization of the measure's impact on both the hazard as well as the damage function. `entity.measures.name.hazard_intensity_impact(m)` allows to reduce the hazard intensity (e.g. -1 reduces tropical cyclone windspeed by 1 m/s) for measure m. The `hazard_high_frequency_cutoff`[27] allows to specify a frequency below which damages are suppressed due to the measures, e.g. the construction/design level of a dam (`hazard_high_frequency_cutoff`=1/50 means the dam prevents damages up to the 50 year return period). `hazard_event_set` allows to specify a measure-specific hazard event set, i.e. for this particular measure, climada switches

[25] Please refer tot he measures tab in the Excel file and the comments in each of the header fields.

[26] entity.measures.color{m} contains the color (RGB) as shown in the adaptation cost curve of measure m. `colorRGB` contains this converted into an RGB triple.

[27] We do not `repeat entity.measures.X(m)` any more, just refer to `X`.

to the specified hazard event set instead of the one used to assess the damages of the reference case. MDD_impact_a and MDD_impact_b allow a linear transformation of the MDD (mean damage degree) of the damage function, such that MDD_{eff} = MDD_impact_a + MDD_impact_b * MDD. Similarly, PAA_{eff} = PAA_impact_a + PAA_impact_b * PAA. damagefunctions_map allows to map to a new damage function to render the effect of measure m, i.e. '1to3' means instead of DamageFunID 1, DamageFunID 3 is used[28]. risk_transfer_attachement and risk_transfer_cover define the attachement point and cover of a risk transfer layer[29].

The simple call

```
measures_impact=climada_measures_impact(entity,hazard,'no');
```

does it all, e.g. it takes the entity and first calculates the EDS_{ref} using hazard in order to create the baseline (situation with no measure applied). It then takes masure m (m=1…), adjusts either hazard and/or damagefunctions according to the measure's specification and calculates a new EDS_{m}. The difference to EDS_{ref} (i.e. EDS_{m}-EDS_{ref}) quantifies the benefit (averted damage) of measure m. By doing this on the event damage set, a variety of measures can be compared, even account for measures which for example only act on high frequency events (see hazard_high_frequency_cutoff) or risk transfer layers (see risk_transfer_attachement and risk_transfer_cover). This function further handles all the measure impact discounting etc[30].

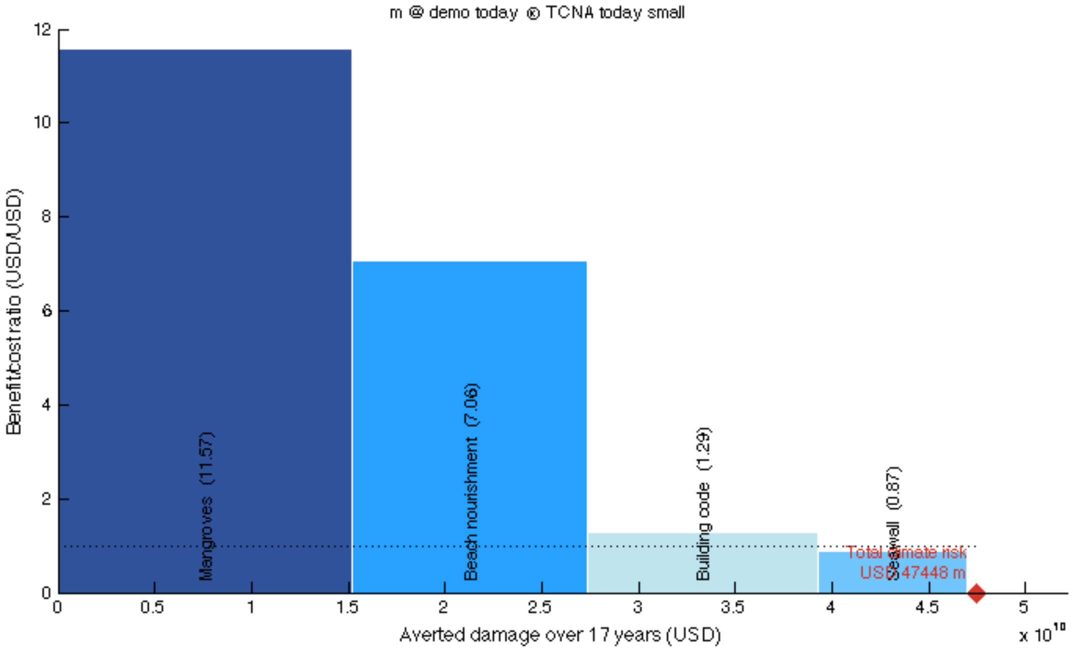
Since it would be quite cumbersome for the user to manually construct the adaptation cost curve based on the detailed output provided by climada_measures_impact, the following function does it all:

```
climada_adaptation_cost_curve(measures_impact)
```

[28] The filed entity.measures.damagefunctions_mapping contains the details, i.e. the mapping as used in climada, a kind of 'parsed' version of e.g. '1to3'.

[29] Please refer the tot he lecture, www.iac.ethz.ch/edu/courses/master/modules/climate_risk

[30] See function climada_NPV



and finally the effect of adaptation measures on different return periods:

```
climada_adaptation_event_view(measures_impact)
```

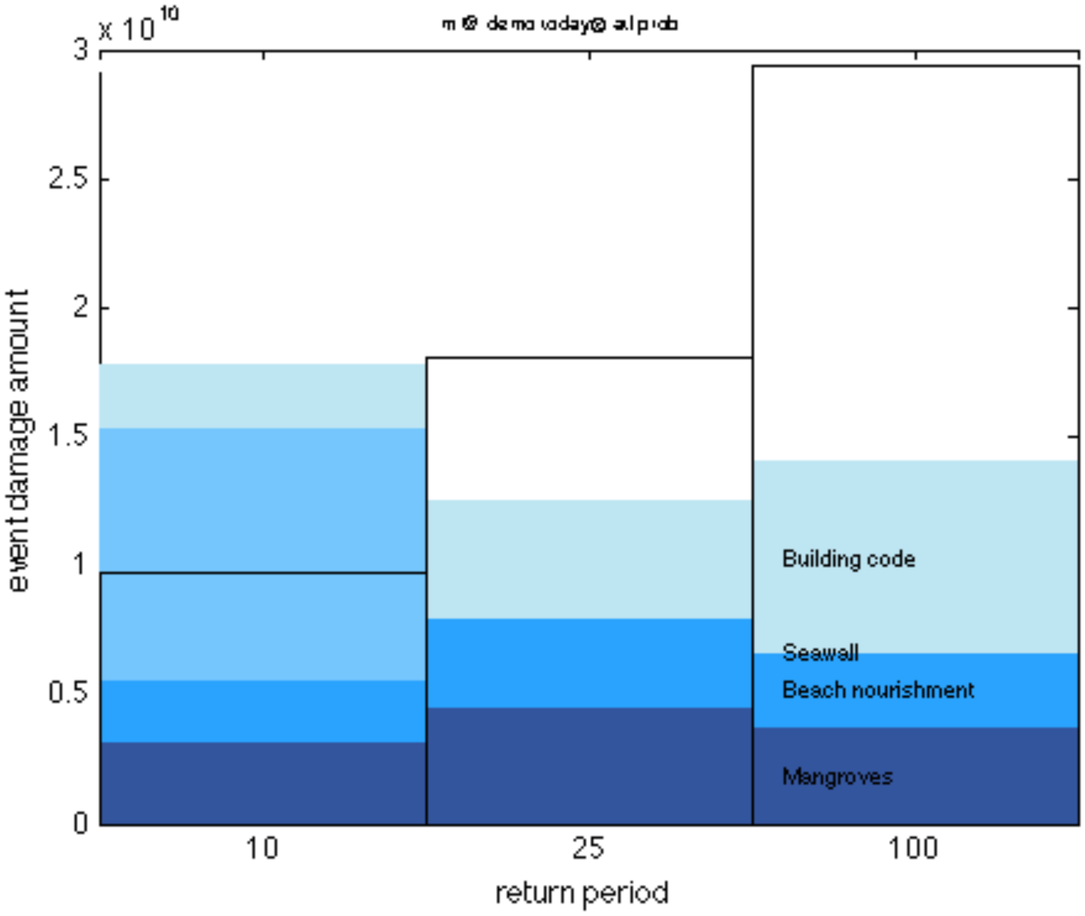


Figure: the effect of adaptation measures on return periods of 10, 25 and 100 years. Note that the 10-year event can be fully mitigated by proposed measures, about 70% of the 50-year and about half of the 100-year event.

Function reference

This section makes reference to key climada functions in order to provide the user with a starting point – the function are provided in a somewhat logical order, i.e. one would usually use functions listed further down later in the process. Please refer to each functions detailed header (use `help` *`functionname`* in MATLAB). You might also run `compile_all_function_headers` once in order to generate a .html file with all function headers for fast reference.

`climada_demo`: the demo GUI as documented above
`climada_demo_step_by_step`: the step-by-step demo as documented above

Basic entity functions

`climada_entity_read`: read entity from Excel file
- `climada_entity_load`: load a previously saved entity (`climada_entity_read` saves a .mat file – which speeds up subsequent read, unless the original Excel file has been changed, in which case it is re-read and the .mat file overwritten, see `climada_check_matfile`)
- `climada_entity_save`: save an entity (i.e. after modification in MATLAB)
- `climada_damagefunctions_read`: read damagefunctions tab only
- `climada_measures_read`: read measures tab only
- `climada_measures_encode`: encode measures, i.e. interpret them for use in climada_measures_impact

`climada_entity_plot`: plot assets distribution of an entity (entity.assets)
`climada_assets_encode`: encode assets (i.e. to switch to another hazard event set[31])
- `climada_assets_encode_check`: check encoding, plot asset locations and centroids

`climada_damagefunctions_plot`: plot damagefunctions
`climada_damagefunctions_map`: map damagefunctions (i.e. to another entity[32])

Core calculations

`climada_EDS_calc`: calculate event damage set (EDS)
- `climada_EDS_stats`: some statistics of an EDS
- `climada_EDS_save`: save EDS
- `climada_EDS_load`: load EDS

`climada_EDS_DFC`: plot damage frequency curve(s)
- `climada_damage_exceedence`: the damage exceedence calculation
- `climada_EDS_DFC_report`: write an Excel or .csv report of the DFC(s)

`climada_waterfall_graph`: plot the waterfall (risk today, Δeconomic, Δclimate) graph
`climada_measures_impact`: calculate the impact of adaptation measures
- `climada_NPV`: net present value (NPV) calculation

`climada_adaptation_cost_curve`: show the adaptation cost curve
`climada_adaptation_event_view`: the event view on adaptation measures

[31] climada_entity_read prompts for a hazard event set and hence encodes to the selected hazard's centroids already. For speedup, this is done prior to calling climada_EDS_calc, as mapping all asset locations to the centroids oft he hazard event set usually does not need to be repeated each time (e.g. only once for the series of calculations involved in assessment of adaptation measures). In case multiple hazards (perils) are assessed, re-encoding is required indeed (that's why climada measures impact works for one hazard at a time only – this code does indeed checl for encoding).

[32] This is especially useful if the user stores all damage functions in a kind of ‚reference' file and attaches the damage functions after reading any new entity, which might itself not contain (all) damage functions (in this case, just disregard the warnings issues by climada_entity_read).

Basic hazard functions

`climada_hazard_plot`: plot hazard events, max intensity etc.

`climada_hazard_load`: load hazard event set

`climada_hazard_stats`: plot hazard intensity return period maps

`climada_excel_hazard_set`: create a hazard set based on scenarios as provided in an Excel file, see ../data/hazards/Excel_hazard.xls

`climada_hazard_cleanup`: cleanup a hazard event set (check for internal consistency)

`climada_hazard_clim_scen`: create a climate scenario version of a hazard event set

Further display functions

`climada_plot_world_borders`: plot world borders

`climada_circle_plot`: plot any values at coordinates as circles

`climada_color_plot`: plot any values at coordinates as colored area

`climada_DFC_compare`: compare a damage frequency curve (DFC) with other model output

Tropical cyclone (TC) specific functions

`climada_tc_get_unisys_databases`: get all TC (besttrack) databases from www

`climada_tc_read_unisys_database`: read (besttrack) data

`climada_tc_random_walk`: generate probabilistic tracks

`climada_tc_windfield`: generate the windfield for one TC event

`climada_tc_hazard_set`: generate a TC hazard event set

`climada_tc_windfield_animation`: animate a single TC track's windfield

`climada_plot_ACE`: plot accumulated cyclone energy (ACE)

Basic functions

`climada_xlsread`: read Excel file

`climada_odsread`: read .ods (Open Office) file, see also climada_init_vars to set this as default

`climada_shaperead`: read shape file (does require MATLAB mapping toolbox)

`climada_centroids_read` and `climada_centroids_load`: read and load centroids

Admin functions

`compile_all_function_headers`: generate a html file with headers of all functions (these headers explain all input and output of each function)

`climada_code_copy`: copy all code into a folder for easy transfer (see GIT, too)

`climada_code_update`: update local code based on the file provided by `climada_code_copy`

`climada_git_pull_repositories`: on a machine with GIT (https://github.com) installed, update all local code and data

`climada_template`: the function template to start new code from

`climada_country_name`: get country name and admin0 ISO3 code, see also ../data/system/admin0.txt and admin.xls

`climada_init_vars`: init global varables (called upon startup by `startup`)

`climada_init_folders`: init folder structure (useful when creating a new module)

`startup`: the startup function, sets root folder and manages MATLAB path[33]

[33] Adds paths to all climada modules

`climada_check_matfile`: check whether the .mat (binary, fast access) version of a file is older than the (Excel) file, used e.g. in `climada_entity_read`, which reads the .mat file on second call, unless the Excel entity file has been edited.

Special functions

`climada_code_optimizer`: remove some parts from core code (like `climada_EDS_calc`) for speedup
`climada_distance_km`: calculate distance between points in km
`climada_nonspheric_distance_m`: more precise distance in m
`climada_collect_measures_impact`: collect impact files for two hazards created by `climada_measures_impact` (sometimes handy to process some measures separately)
`waitbar_toggle`: toggle waitbars (on/off), see also `climada_global.waitbar`

climada modules

While the core climada provides the user with the core probabilistic damage calculation and climate adaptation measures assessment functionalities, it only contains a simple tropical cyclone hazard. Therefore, there are climada extensions, called modules, to add functionality. Since the core climada only contains a simple tropical cyclone hazard, one of the first modules to be considered might be tc_hazard_advanced, which improves the quality of the tropical cyclone hazard event set. There exists modules for other perils (to generate or make sue of other hazards, such as tc_surge and tc_rain, ws_europe and eq_global) and for other functionality, like automatic generation of assets (country_risk and GDP_entity). Each module contains (similar to core climada) a code, data and docs folder, with a detailed documentation in the file {module_name}.pdf in the docs folder. Therefore, one might first inspect these files still on GitHub before downloading a specifc module[34].

GDP_entity, https://github.com/davidnbresch/climada_module_GDP_entity
Create a default asset base for almost any specific country, consisting of centroids (used later e.g. to generate a hazard event set of matching resolution) and assets scaled to the country GDP for today and future.

country_risk, https://github.com/davidnbresch/climada_module_country_risk
This module runs all (available) perils for one country (or list of countries). It generates country or admin1 (state/province) assets, the earthquake (EQ), tropical cyclone (TC), torrential rain (TR) and storm surge (TS) hazard event sets, checks for European winter storm (WS) exposure and runs all risk calculations for a given country.

eq_global, https://github.com/davidnbresch/climada_module_eq_global
This module implements a raw global earthquake model. Consider climada modules country_risk or GDP_entity to generate the centroids.

etopo, https://github.com/davidnbresch/climada_module_etopo
This module implements ETOPO, a global bathymetry (and topography) dataset. It’s a separate module, since topographic (and bathymetry) information can be used in various contexts – and since the dataset is quite large (ETOPO1 is 933 MB, ETOPO2 still 233 MB).

[34] Please refer to the section ‚Getting started’ above about where to store the module(s). The process is also described in each module’s readme file.

meteorites, https://github.com/davidnbresch/climada_module_meteorites
This module implements a basic meteorite global hazard. Consider climada modules country_risk or GDP_entity to generate the centroids.

tc_hazard_advanced,
https://github.com/davidnbresch/climada_module_tc_hazard_advanced
This module implements the tropical cyclone (TC) attenuation after landfall for probabilistic events. Make yourself familiar with the core climada tropical cyclone hazard event set (and its generation) first. A good implementation of both the basic, probabilistic and advanced tropical cyclone hazard generation can be found in the climada module country risk and there in the routine centroids_generate_hazard_sets.

tc_rain, https://github.com/davidnbresch/climada_module_tc_rain
This climada module allows to generate the precipitation fields accompanying a tropical cyclone - the torrential rain (TR) hazard event set. Please see tc_hazard_advanced, too, as generating rain fields makes far more sense after application of this module.

tc_surge, https://github.com/davidnbresch/climada_module_tc_surge
This module implements a tropical cyclone storm surge model . It's based on climada's core tropical cyclone (TC) module. Hence in order to run tc_surge, make sure you've made yourself familiar (to some extent at least) with the core climada tropical cyclone hazard module. Needs the climada module etopo, highly recommended to make use of module tc_hazard_advanced , too.

ws_europe, https://github.com/davidnbresch/climada_module_ws_europe
This climada module contains the European winter storm hazard event sets as used in the following publication: Schwierz, C., P. Köllner-Heck, E. Zenklusen Mutter, D. N. Bresch, P.-L.Vidale, M. Wild, C., and Schär, 2010: Modelling European winter wind storm losses in current and future climate. Climatic Change (2010) 101:485?514, doi: 10.1007/s10584-009-9712-1.

barisal_demo, https://github.com/davidnbresch/climada_module_barisal_demo
Barisal, Bangladesh, demo module, all numbers and results are for demonstration purposes only.

Appendices

The appendices contain detailed description of relevant aspects and shall provide the advanced user with further information and especially serve those consider expanding climada functionality.

climada, the inner workings

This section describes the core damage calculation. The damage is calculated for each single asset at each location for each scenario or event, so basically

damage = asset value * damage function

where damage is summed up over assets and events, i.e. above line is at the core of two loops, the outer one over assets, the inner one over events. More precisely:

damage = asset value * MDD * PAA, where

- MDD * PAA is the damage function
- damage is the damage 'from ground up', from the first dollar, so to speak
- asset value is the total value of the asset. Note again that value does not need to a monetary value, it can also e.g. signify number of people at a given location.
- MDD is the Mean Damage Degree (the damage for a given intensity at an affected asset) - how strongly an asset is damaged. Range 0..1 (from none to total destruction). In the case of asset value signifying number of people at a given location, MDD represents the severity with which those people are affected.
- PAA is the Percentage of Assets Affected (the percentage of assets affected for a given hazard intensity) - how many assets are affected. Range 0..1 (from none affected to all affected). In the case of asset value signifying number of people at a given location, PAA represents the percentage of people affected. As the product MDD*PAA ultimately counts, the user shall just make sure this product makes sense for the class of assets under consideration.

So far, the hazard intensity did not show up in the calculation, did we miss something? Well, the damage is a function of the hazard intensity, hence:

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

where hazard intensity is the hazard's intensity at each asset for each event. Since the damage also depends on the asset type, we have in fact:

MDD = f(hazard intensity, asset type)
PAA = f(hazard intensity, asset type)

While the hazard intensity is simply the `entity.damagefunctions.Intensity`, the asset type is referred to by the `DamageFunID`, i.e. for a certain type of assets, the user defines a specific `DamageFunID` in the assets tab and the corresponding damage function (`MDD` and `PAA` as function of `Intensity`) in the damagefunctions tab of the entity Excel file.

Implementation

The core calculation is done by `climada_EDS_calc`, where EDS stands for event damage set, i.e. a vector with calculated damage for each event (or simply the vector of event damages). The variables in the code have speaking names, but the inner loop is vectorized, hence warrants some comments.

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

- `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, one element for each hazard event, `PAA` is the vector or PAAs, also one element for each hazard event.
- `.*` is the element-wise (scalar) multiplication

So far, the hazard intensity did not show up in the calculation, did we miss something? Well, the damage function 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, about line 170ff of `climada_EDS_calc`) – how to get the vector of MDDs.

Remember: outer loop (explicit) over assets, inner loop (implicit) over events and also remember that the hazard event set contains

- `hazard.intensity(event_j,centroid_i)`: the hazard intensity (like windspeed in m/s) at centroid_i for event_j
- `hazard.frequency(event_j)` contains the event-frequency for event_j

```
for asset_i=1:n_assets % approx line 170ff in climada_EDS_calc.m

    % 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[35]
    MDD=spfun(@climada_sparse_interp,hazard.arr(:,asset_hazard_pos));
```

[35] `interp_x_table` and `interp_y_table` are passed as global variables to `climada_sparse_interp` for performance reasons

```
    % 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;

    % add to the resulting EDS (event damage set) structure:
    EDS.damage=EDS.damage+temp_damage'; % add to the EDS[36]
    EDS.Value=EDS.Value+entity.assets.Value(asset_i); % add Value

end % asset_i
```

Next, you might consider “climada implementation of insurance conditions” further below.

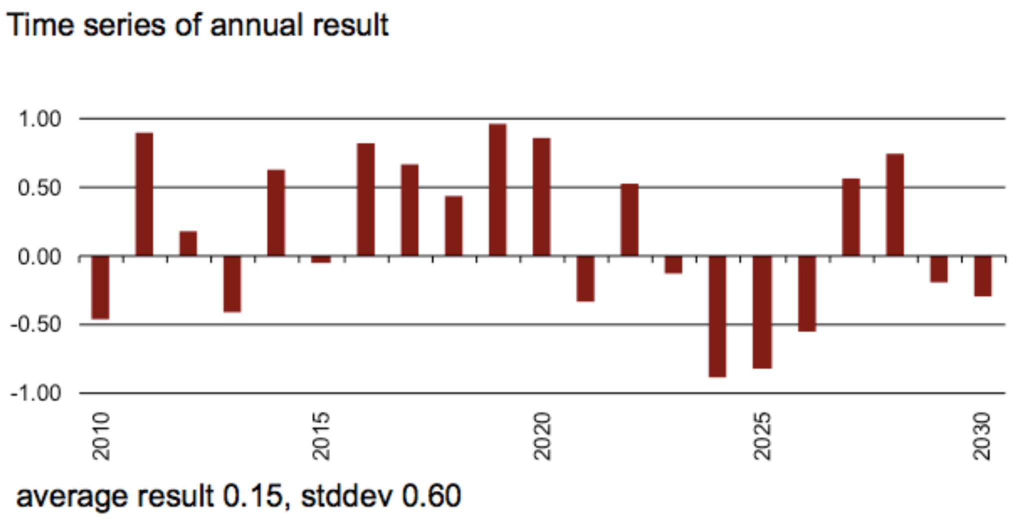
Insurance remarks

Insurability

Insurance is the mutual cover of a fortuitous, assessable need of a large number of similarly exposed business [Alfred Manes, 1877-1963].

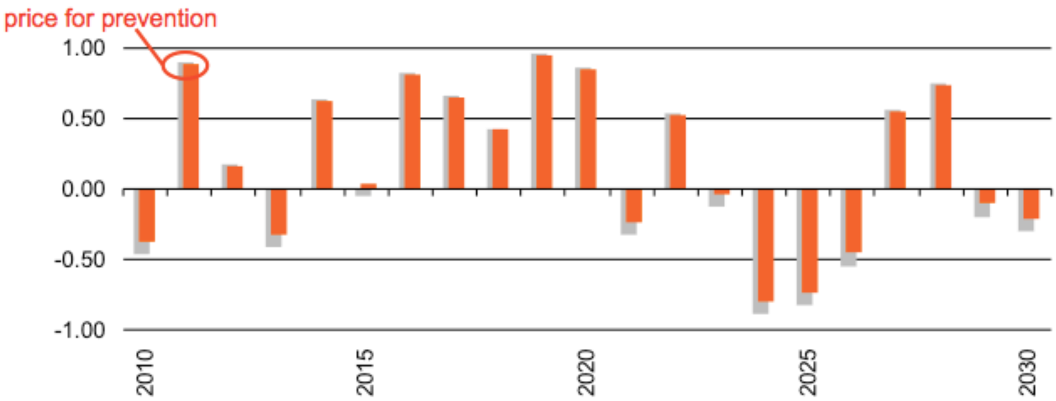
- mutuality: numerous exposed parties must join together to form a risk community, to share and diversify the risk à large number
- fortuitous or randomness: time of occurrence must be unpredictable, occurrence itself must be independent of the will of the insured
- assessability: damage probability and severity must be quantifiable
- similarly exposed business: a large number
- plus: economic viability: private insurers must be able to obtain a risk-adequate premium

The following figures show the effect of insurance, including the working (and benefit) of risk reducing prevention measures. (Climate) adaptation measures work in a similar fashion, hence do not only reduce risk, but render insurance more attractive (i.e. affordable).



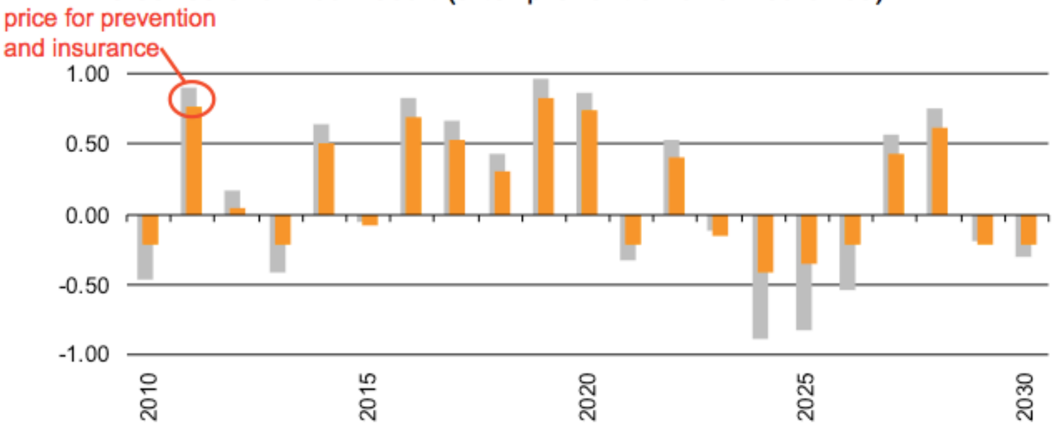
[36] A note on ' : for historical reasons the EDS.damage vector is transposed

Time series of annual result (after prevention)



average result 0.19 (+25%), stddev 0.56 (-8%), prevention price 0.01
→ effect of prevention: stabilize result, reduce volatility

Time series of annual result (after prevention and insurance)



average result 0.17 (+12%), stddev 0.43 (-29%), prev+ins price: 0.13
→ effect of insurance: reduce (extreme) volatility

In summary, we therefore have:

	Result	stdev	price[37]
raw	0.15	0.60	
+ prevention	0.19 (+25%)	0.56 (-8%)	0.01

→ cost-effective adaptation (net gain of 0.04 at cost of 0.01)

+ insurance (and prevention)	0.17 (+12%)	0.43 (-29%)	0.01+0.12

→ substantial reduction of volatility, result increase even after deduction of prevention cost and insurance premium → affordable!

for comparison: insurance alone	*0.12 (-17%)*	*0.45 (-25%)*	*0.153*

→ prevention (strongly) incentivizes insurance

Key drivers for risk transfer demand to complement risk reduction measures are:

- Volatility of (remaining) damage
- Level and trend of expected damage (related to budget)

[37] Note that price is already taken into account in result

- Damage clusters (relative to budget and financing capacity)
- Budget constraints and opportunity costs (e.g. school investments)
- Availability of emergency relief capital
- Subjective risk appetite

The last point shall be no means be underestimated, as it refers also strongly to risk culture.

Forms of insurance

Risk transfer can be agreed upon based on different triggers:

- indemnity[38], also called incurred or occurred damage: The incurred damage is compensated for, i.e. the insured sends the bill for fixing the damage and gets reimbursed. Pro: exact amount paid, Con: takes time, involves damage assessment and may contain moral hazard (insurance fraud).
- parametric, also called index: A physical parameter exceeding a certain threshold (e.g. wind speed above 35 m/s) triggers the payment of a pre-agreed amount – or more generally an amount as a function of the parameter(s). Pro: fast, unbureaucratic, pre-agreed. Con: the pre-agreed value is paid, actual damage might differ (so called basis risk retained by the insured[39]).
- modelled (well, a form of parametric): A model is run after an event, based on the key properties of the event (e.g. a wind footprint as measured by a meteorological office) and the resulting modeled damage amount is paid

and with different partners, such as:

- policyholders – from macro (e.g. large corporates in Texas) to micro (e.g. smallholder farmers in Ethiopia) – and the usual single household
- insurers (reinsurers insure them)
- other reinsurers, called retrocession
- capital market, called insurance-linked security (ILS) or Cat Bond[40]
- public sector (public-private partnership, PPP)

and can be based either on free choice (whether the take up insurance or not) or mandatory[41] – or any shade in between. Please note the issue of adverse selection (especially pertinent for perils such as flood), i.e. that only the most at risk seek cover, rendering the scheme costly or even non-commercially viable – and hence the justified consideration of compulsory or mandatory schemes.

Insurance conditions

[38] specific or market-share

[39] That's why one often finds hybrid solutions, but we refrain from getting into details.

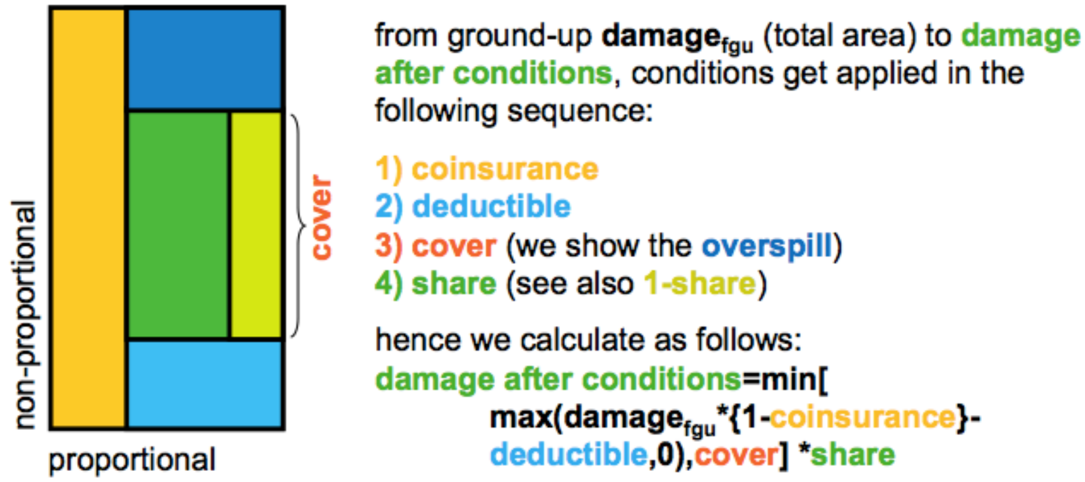
[40] Not this one:

[41] See e.g. Source: Efficient Monopolies. The Limits of Competition in the European Property Insurance Market, Thomas von Ungern-Sternberg, Oxford University press, 2004. ISBN 0-19-926881-9

There are basically two types of insurance conditions, proportional and non-proportional. Proportional means the insured retains a proportional (linear) fraction of the damage, non-proportional in essence means the insured is covered above a certain threshold (called deductible) for a certain cover[42].

proportional: $\text{damage}_{\text{after}}=\text{damage}_{\text{before}}$ * share
non-proportional: $\text{damage}_{\text{after}}=\min(\max(\text{damage}_{\text{before}} - \text{deductible},0),\text{cover})$

Where share needs to be defined depending on who shall be liable for damage after, i.e. if $\text{damage}_{\text{after}}$ is the damage to be reimbursed by the insurer, share is the insurer's share of the total or 'from ground up' damage. Likewise, $\text{damage}_{\text{after}}$ in above notation in the non-proportional case is the damage the insurer is liable for, as the deductible (and any damage exceeding the cover) remains with the insured.



from ground-up $\text{damage}_{\text{fgu}}$ (total area) to damage after conditions, conditions get applied in the following sequence:

1) coinsurance
2) deductible
3) cover (we show the overspill)
4) share (see also 1-share)

hence we calculate as follows:
damage after conditions=min[
max($\text{damage}_{\text{fgu}}$*{1-coinsurance}-deductible,0),cover] *share

Figure: the elements of insurance conditions. The event based approach as followed in climada does allow for the consideration of all these elements.

climada implementation of insurance conditions

Please read the above section "" carefully first. Remember that the outer (explicit) loop is over assets, the inner (implicit) one over events.

```
for asset_i=1:n_assets % approx line 170 in climada_EDS_calc.m
     […]
     % calculate the from ground up (fgu) damage
     temp_loss=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

     % add to the resulting EDS (event damage set) structure:
     EDS.damage=EDS.damage+temp_damage'; % add to the EDS
     […]
```

[42] There are unlimited covers, but this stretches the second principle of insurability, namely the ability to assess the outcome.

```
end % asset_i
```

Similarly, any conditions on the event damage set (EDS) can be evalaued, always of the form min(max(loss-deductible,0),cover), i.e. a non-proportional per-event cover (CatXL) can be evaluated on the EDS as:

`EDS.damage`=min(max(`EDS.damage` -deductible,0),cover)

Note on scenarios

Since climada makes use of scenarios in at least three instances, it might be worthwhile providing a definition and some remarks.

climada uses scenarios e.g. in:

- hazard event set generation (generating artificial single hazard events or scenarios, such as by 'wiggling' path and intensity of tropical cyclones)

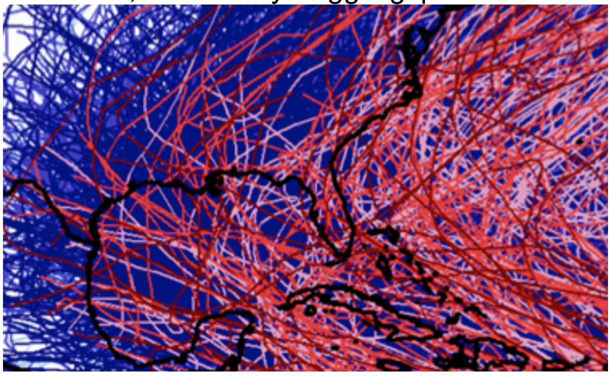
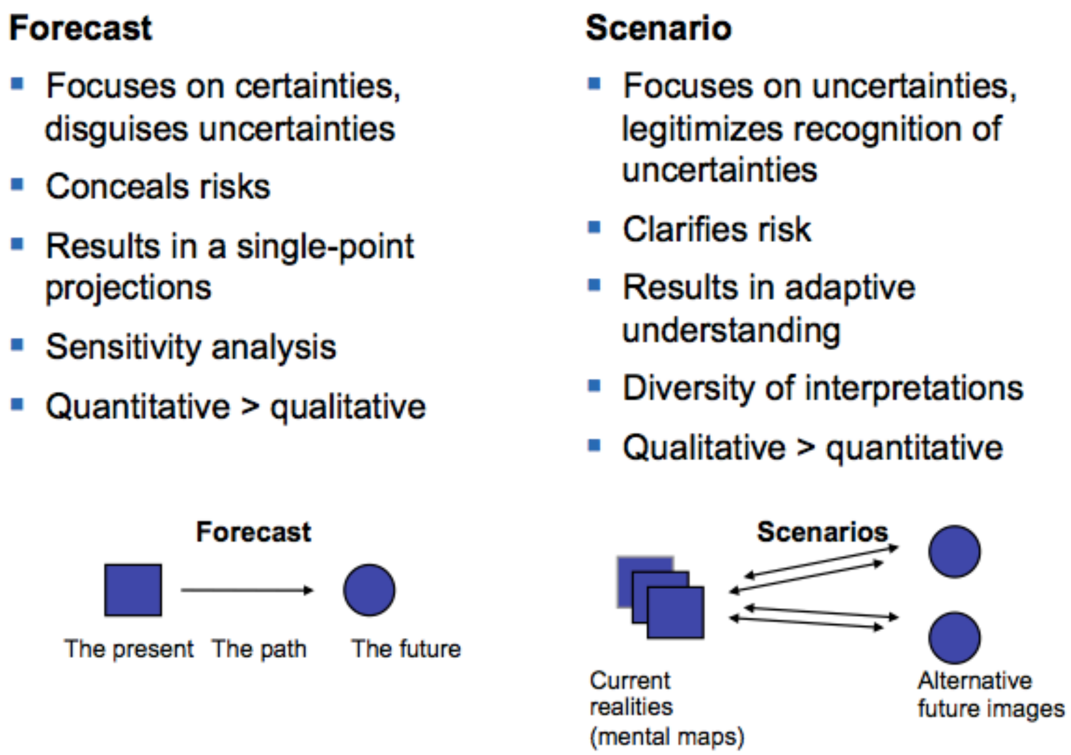
  

  Figure: historic (red) and probabilistic (blue) storms, see functions `climada_tc_*`
- charting out economic pathways (the economic development scenarios, leading to future assets) and
- last but not least climada uses climate impact scenarios in the sense of modified hazard event sets (one could also see them as events compatible with future climate conditions).

Definition: A scenario is a snapshot that describes a possible and plausible future. Scenario analysis is a systematic approach to anticipate a broad range of plausible future outcomes.

Scenario analysis is used in general …

- as a risk management tool to assess the potential impact of an event or development to anticipate and understand risks (as e.g. in the climada hazard event sets)
- as a tool to spot new business opportunities and to discover strategic options (e.g. as climada adaptation measures)
- as foresight in contexts of accelerated change, greater complexity and interdependency
- for evaluation of highly uncertain events that could have a major impact (e.g. climada climate change hazard event sets)
- to steer mitigation strategies, implementation and monitoring by reviewing and tracking different possible developments (as in the whole economics of climate adaptation assessment)



Forecast

- Focuses on certainties, disguises uncertainties
- Conceals risks
- Results in a single-point projections
- Sensitivity analysis
- Quantitative > qualitative

Scenario

- Focuses on uncertainties, legitimizes recognition of uncertainties
- Clarifies risk
- Results in adaptive understanding
- Diversity of interpretations
- Qualitative > quantitative

Figure: Some key properties of forecasts and scenarios in comparison.

climate impact scenarios – remarks on climada implementation

There are different ways to represent climate change scenarios in the model.

Representation is possible via

- Parameterized impact:
  Estimate the climate change impact on key hazard parameters and represent those changes in the probabilistic event set, either by
  - re-generating the probabilistic event set based on these parameters (e.g. consider changing properties of `tc_track` prior to calling `climada_tc_hazard_set`) or by
  - reflecting those changes by modification of the 'present climate' hazard event set (e.g. multiply the hazard intensity by a factor), see further below
- Downscaled event set:
  Extract events from a downscaled GCM-driven model chain[43]

Note that a changing climate might also have impacts on e.g. vulnerabilities

While there all degrees of freedom to implement climate change impact scenarios in climada, the following few remarks might be of value:

Remember that hazard contains the hazard event set, `hazard.intensity` the sparse array with intensities, `hazard.frequency` the vector of event (occurrence)

[43] Please refer to the climada module ws_europe (https://github.com/davidnbresch/climada_module_ws_europe) and see Schwierz, C., P. Köllner-Heck, E. Zenklusen Mutter, D. N. Bresch, P.-L.Vidale, M. Wild, C., and Schär, 2010: Modelling European winter wind storm losses in current and future climate. Climatic Change (2010) 101:485?514, doi: 10.1007/s10584-009-9712-1.

frequencies. Therefore, the following cases are very straightforward (we use wind speed as example, works similarly for parameters such as flood height):

- Increase wind speed for all events by 5%:
  `hazard.intensity =hazard.intensity *1.05;`
- Increase event frequencies by 5%:
  `hazard.frequency=hazard.frequency*1.05;`
- Increase all wind speeds by 5 m/s (note that hazard.intensity is sparse, hence we first need to identify the non-zero elements);
  ```
  nz_pos=find(hazard.intensity) %non-zeros
  hazard.intensity(nz_pos)=hazard.intensity(nz_pos)+5;
  ```
- Increase only wind speeds > 45 m/s by 5 m/s:
  ```
  pos45=find(hazard.intensity>45);
  hazard.intensity(pos45)=hazard.intensity(pos45)+5;
  ```

The code `climada_hazard_clim_scen` allows for such impact parameterizations.

Tropical cyclones – technical remarks

Windfield calculation

To determine the impact of any given storm, function `climada_tc_windfield` generates wind field resulting from single track of tropical cyclone. The function starts from ther track center `tc_track.MaxSustainedWind` in knots and generates the 2D windfield in `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 (as shown in the step-by-step approach at the beginning of the manual, see `climada_demo_step_by_step`). The windfield calculations are speeded up by only calculating for centroids within 750 km distance of min/max track lon/lat.

```
res = climada_tc_windfield(tc_track(1170),centroids);
```

For details, see the header of `climada_tc_windfield`

Method

Currently, the code implements the Holland windfield[44]. 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:

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

$$S = \begin{cases} \min\left(M, \quad M + 2 \cdot T \cdot \dfrac{D}{R}\right) & D \leq R \quad \text{in the inner core} \\ \max\left(0, \quad \left((M - \operatorname{abs}(T)) \cdot \left(\dfrac{R}{D}\right)^{3/2} \cdot e^{1-\left(\frac{R}{D}\right)^{3/2}}\right) + T\right) & D < 10 \cdot R \quad \text{in the outer core} \\ 0 & D > R \quad \text{out of radius} \end{cases}$$

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.

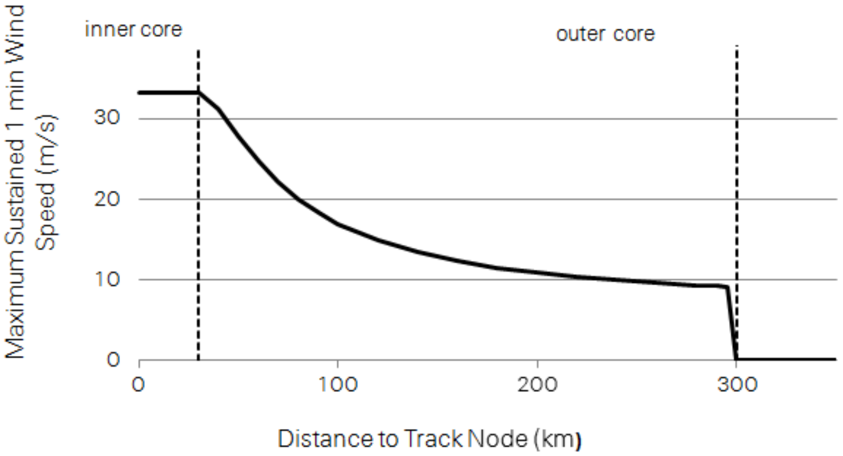


Figure 1: Maximum sustained 1 min wind speed in relation to the distance to the track node.

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

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

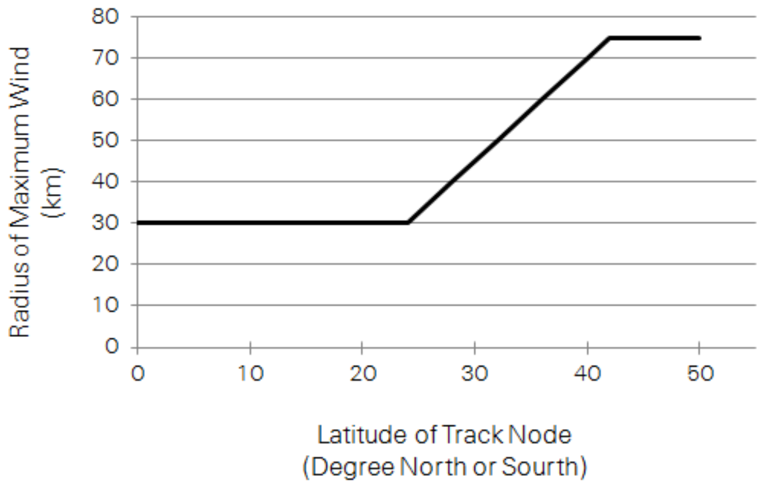


Figure 2: Radius of maximum wind in relation to latitude of track node.

Finally, the wind speed, *S*, describes the maximum sustained 1 min wind speed. To derive wind gusts lasting just a few seconds (3-5 s), we note that 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 wind field parameterization can be implemented in a similar fashion (just implement in a copy of `climada_tc_windfield`, e.g. `climada_tc_MY_windfield`, 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. 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
```

Obtain centroids (points at which to evaluate the winfield) using

```
centroids = climada_centroids_read('',1)
```

Note that this call with the 1 also plots the centroids (use the zoom function on the map). See also the parameter `check_plot` in the PARAMETER section of the `climada_tc_windfield` code or refer to the routine `climada_color_plot`.

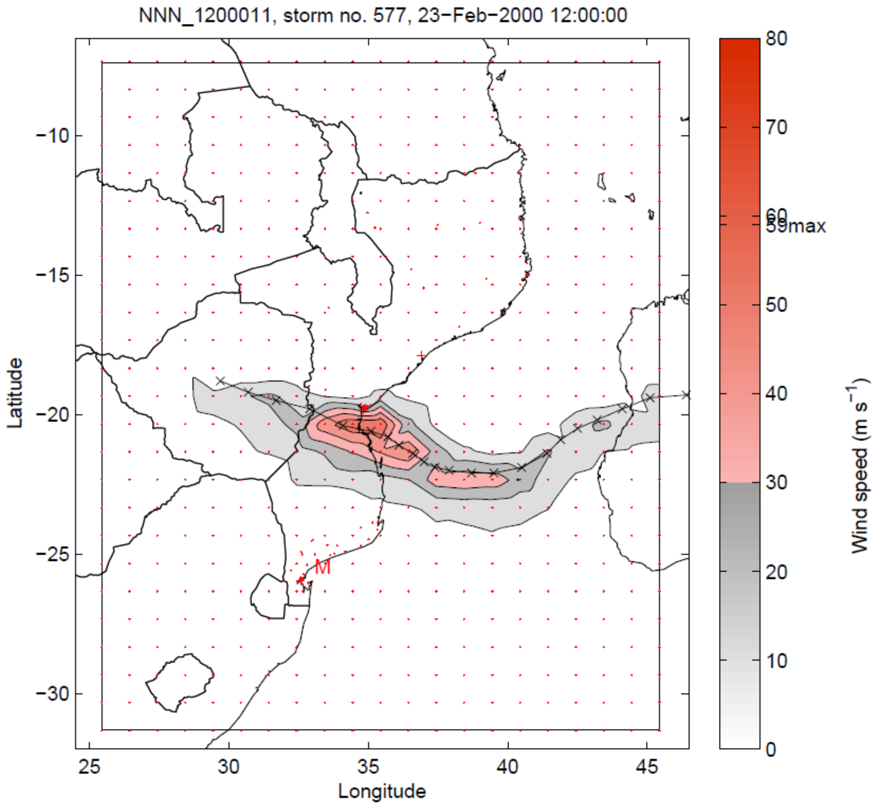


Figure: Wind field calculated based on track 577 of the South Indian Ocean. This particular track results in the second highest wind speed in the city of Beira, Mozambique.

Single cyclone track evolution animation

The function `climada_tc_windfield_animation`[45] 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.

[45] See climada module https://github.com/davidnbresch/climada_module_tc_hazard_advanced

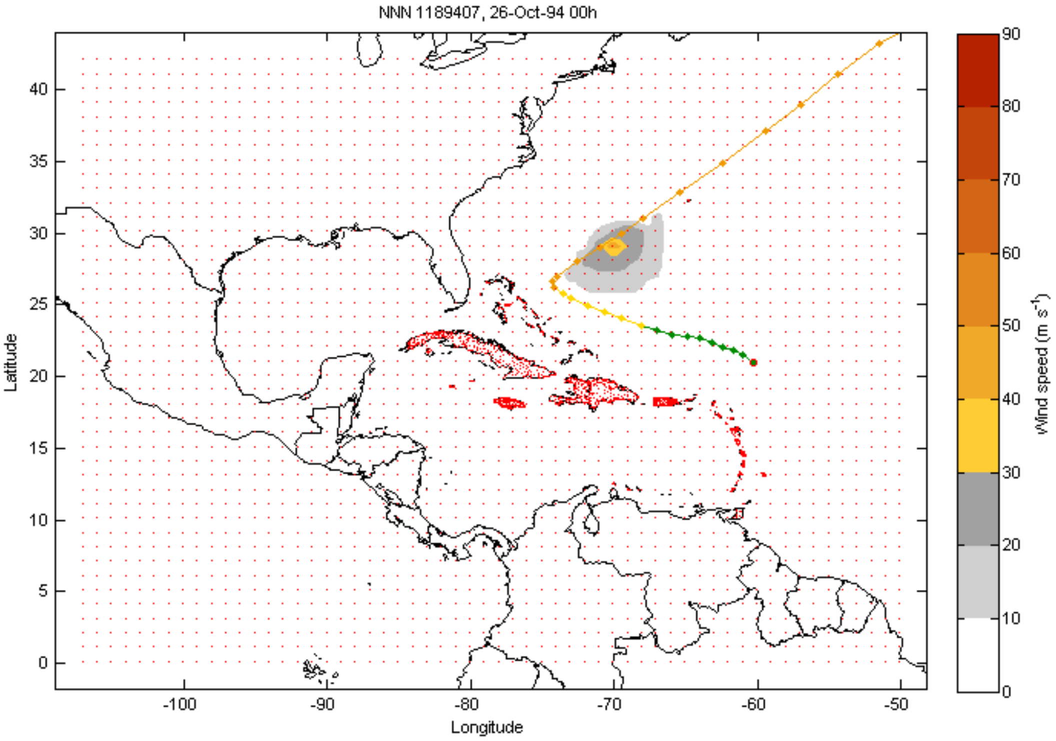


Figure: Snapshot from the animation, wind field calculated for every time step.

Economics of Climate Adaptation (ECA) – key routines

The function `climada_waterfall_graph` plots the waterfall figure for today's damage and future damage including economic growth and climate change for the annual expected loss or any specific return period. Inputs are the three event damage sets (e.g. EDS_today.mat, EDS_2030.mat, EDS_2030_clim.mat), prompted for if not given. Any specific return period or annual expected damage can be chosen.