VAWS user manual

Release 3.3

Geoscience Australia

Sep 12, 2019

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER

ONE

INTRODUCTION

Vulnerability and Adaptation to Wind Simulation (VAWS) is a software tool that can be used to model the vulnerability of small buildings such as domestic houses and light industrial sheds to wind. The primary use-case of VAWS is the examination of the change in vulnerability afforded by mitigation measures to upgrade a building's resilience to wind hazard.

1.1 Background

Development of VAWS commenced in 2009-2010 in a collaborative project, partly funded by the then Department of Climate Change and energy Efficiency (DCCE), between Geoscience Australia, James Cook University and JDH Consulting *[2][7]*. The development of the current version was undertaken as part of the Bushfire and Natural Hazard Cooperative Research Centre (BNHCRC) project "Improving the Resilience of Existing Housing to Severe Wind" led by James Cook University.

1.2 Overall logic

The VAWS tool takes a component-based approach to modelling building vulnerability. It is based on the premise that overall building damage is strongly related to the failure of key connections.

The tool generates a building model by randomly selecting parameter values from predetermined probability distributions using a Monte Carlo process. Values include component and connection strengths, external pressure coefficients, shielding coefficients, wind speed profile with height, building orientation, debris damage parameters, and component masses.

Then, for progressive gust wind speed increments, it calculates the forces in all critical connections using influence coefficients, assesses which connections have failed and translates these into a damage scenario and costs the repair. Using the repair cost and the full replacement cost, it calculates a damage index for each wind speed.

For more details, see *Chapter 5*.

1.3 Key features

- Component-based approach:

  A house is modelled consisting of a large number of components, and overall damage is estimated based on damage to each of the components.

- Uncertainty captured through a Monte-Carlo process:

  Various uncertainties affecting house performance are modelled through a monte-carlo process.

- Inclusion of debris and water ingress induced damages:

  In addition to the damage to the connections by wind loads, debris and water ingress induced damages are modelled.

- Internal pressurisation:

  Internal pressure coefficients are calculated at each wind speed following the procedures of AS/NZS 1170.2 (Standards Australia, 2011) using the modelled envelope failures to determine envelope permeability.

1.4 Key uncertainties

The Monte Carlo process capture a range of variability in both wind loading and component parameters. The parameter values are sampled for each model and kept the same through the wind steps.

- Wind direction

  For each house, its orientation with respect to the wind is chosen from the eight cardinal directions either randomly, or by the user.

- Gust wind profile

  Variation in the profile of wind speed with height is captured by the random sampling of a profile from a suite of user-provided profiles.

- Pressure coefficients for zones and coverages

  Pressure coefficients for different zones of the house surfaces envelope are randomly chosen from a Type III (Weibull) extreme value distribution with specified means for different zones of the house envelope, and specified coefficients of variation for different load effects.

- Construction level (or quality)

  Multiple construction levels can be defined with mean and coefficient of variation (CV) factors which will be used to adjust the mean and CV of distribution of connection strength.

- Strength and dead load

  Connection strengths and dead loads for generated houses are sampled from lognormal probability distributions.

1.5 Caveats and limitations

VAWS has been designed primarily as a tool for assessing vulnerability of houses to wind hazard. The simulation outcomes should be interpreted as vulnerability of a group of similar houses on average, even though an individual house is modelled. In other words, the tool is not capable of predicting performance of an individual house for a specific wind event.

CHAPTER TWO

GETTING STARTED

This chapter provides instructions on how to install and run the code for general users. Also it provides instructions for developers on how to install, test and build the package of the code. These instructions have been tested on *Windows 7*, *Linux*, and *OS 10.11.x* and is expected to work on most of modern operating systems.

2.1 Instructions for general users

2.1.1 Installation

The VAWS code currently runs with Python 3.6 with many dependencies. It is recommended to create a Python environment dedicated to the code without disrupting the existing environment. With conda, you can manage environments easily. Instructions below are based on conda, but virutalenv can be used alternatively.

1. Install Miniconda

Download and install Miniconda(https://conda.io/miniconda.html) with Python 3.7. This step can be skipped if either Miniconda or Anaconda with Python 3.7 is already installed.

- Windows
  - Double-click the downloaded *Miniconda3-latest-Windows-x86_64.exe* file.
  - When installation is finished, from the *Start* menu, open the *Anaconda Prompt*.
- Linux
  - In Terminal window, run

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

- Mac
  - In Terminal window, run

```
$ bash Miniconda3-latest-MacOSX-x86_64.sh
```

2. Create a conda environment.

In the terminal client, enter the following command to create the environment called *vaws_env*.

```
conda create -n vaws_env
```

3. Activate the environment.

In the terminal client, enter the following to activate the environment.

```
conda activate vaws_env
```

4. Install the code from conda channel

In the terminal client, enter the following to install the code.

```
conda install -c dynaryu vaws
```

2.1.2 Updating

In case new version of the code is available, you may update the code. The conda environment *vaws_env* should be activated first as *2.1.1 step 3*. And then enter the following commands with in the terminal to update with the new version of the code.

```
conda update -c dynaryu vaws

## Package Plan ##

      environment location: /foo/vaws_env

      added / updated specs:
            - vaws


    The following packages will be downloaded:

            package           |            build
            ------------------|-----------------
            vaws-3.3          |           py36_1         2.3 MB  dynaryu
            ------------------------------------------------------
                                            Total:         2.3 MB

    The following packages will be UPDATED:

      vaws                              3.2-py36_1 --> 3.3-py36_1


    Proceed ([y]/n)?
```

2.1.3 Running through GUI

To run the code, the conda environment *vaws_env* should be activated first as *2.1.1 step 3*. And then enter the following command in the terminal.

```
vaws
```

The default scenario will be loaded as shown in Fig. 2.1. See *chapter 4* for details.

2.2 Instructions for developers

The development of the code is tracked using the git version control system. The source code is at https://github.com/GeoscienceAustralia/vaws.

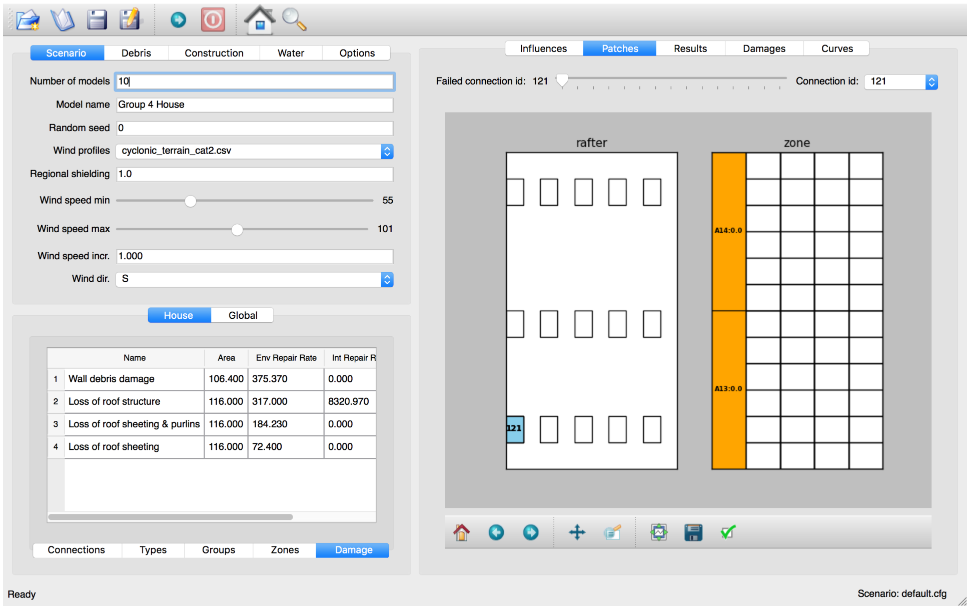

Fig. 2.1: Program main window with default scenario loaded

2.2.1 Installation

1. Get the source code

Source code can be copied by cloning the git repository or downloading the zip file from the git repository.

- If git is installed, run the following command in the terminal

```
$ git clone git@github.com:GeoscienceAustralia/vaws.git
```

- Otherwise download the zip file (https://github.com/GeoscienceAustralia/vaws/archive/master.zip) and then extract it.

This step will create directory called <vaws dir>.

2. Create a conda environment.

Make sure either miniconda or anaconda is installed. Otherwise install either Miniconda or Anaconda with Python 3.6 as *2.1.1 step 1*. Then create the environment called *vaws_env*. by entering the following command in the terminal.

- Windows

```
cd <vaws dir>
conda env create --name vaws_env --file vaws_win64_py3.6.7.yml
```

- Linux/Mac

```
cd <vaws dir>
conda env create --name vaws_env --file vaws_linux64_py3.6.7.yml
```

This will create the environment called vaws_env. The *vaws_env* can be activated as *2.1.1 step 3*.

3. Create GUI

To create the GUI of the code, enter the following commands in the terminal.

- Windows

```
cd <vaws dir>\vaws\gui
build.cmd
```

- Linux/Mac

```
cd <vaws dir>/vaws/gui
./build.sh
```

4. Run the code

The code can be run in either GUI or CLI mode.

- GUI

```
cd <vaws dir>
python -m vaws.gui.main  # for default scenario
python -m vaws.gui.main -c <config_file>  # for a specific scenario
```

- CLI

```
cd <vaws dir>
python -m vaws.model.main -c <config_file>  # for a specific scenario
```

2.2.2 Building the conda package

Steps for the conda package is described below. Please refer to (https://conda.io/docs/user-guide/tutorials/build-pkgs.html) for details.

1. Install conda-build and anaconda-client

To build the package, you need to install *conda-build* and *anaconda-client* in the conda *root* environment not the *vaws_env* environment. And then enter the following in the terminal.

```
conda install conda-build anaconda-client
```

2. Build the package

In the terminal client, enter the following to build the package.

```
cd <vaws dir>/build
conda-build .
```

At the end of the building, you should see something like below:

```
Updating index at /foo/anaconda2/conda-bld/noarch to make package␣
↪installable with dependencies
INFO:conda_build.build:Updating index at /foo/anaconda2/conda-bld/noarch␣
↪to make package installable with dependencies
Nothing to test for: /foo/anaconda2/conda-bld/osx-64/vaws-2.0.3-py27_1.
↪tar.bz2
# Automatic uploading is disabled
# If you want to upload package(s) to anaconda.org later, type:

anaconda upload /foo/anaconda2/conda-bld/osx-64/vaws-2.0.3-py27_1.tar.bz2

# To have conda build upload to anaconda.org automatically, use
```

(continues on next page)

(continued from previous page)

```
# $ conda config --set anaconda_upload yes

anaconda_upload is not set.  Not uploading wheels: []
```

3. Upload to anaconda channel

In the terminal client, enter the following to upload the package to the channel.

```
anaconda login
anaconda upload <package>
```

2.2.3 Testing the code

To test the code, the conda environment *vaws_env* should be activated first as *2.1.1 step 3*. And then enter the following command in the terminal.

```
cd <vaws dir>
python -m unittest -v
```

You should see something similar to below.

```
test_distribute_damage_by_row (vaws.model.tests.test_simulation_batten.
↪TestHouseDamage) ... ok
test_calc (vaws.model.tests.test_stats.MyTestCase) ... ok
test_calc2 (vaws.model.tests.test_stats.MyTestCase) ... ok
test_calc_big_a_b_values (vaws.model.tests.test_stats.MyTestCase) ... ok
test_compute_arithmetic_mean_stdev (vaws.model.tests.test_stats.MyTestCase) ... ok
test_compute_logarithmic_mean_stdev (vaws.model.tests.test_stats.MyTestCase) ... ok
test_gev_calc (vaws.model.tests.test_stats.MyTestCase) ... ok
test_gev_calc2 (vaws.model.tests.test_stats.MyTestCase) ... ok
test_sample_logrnormal (vaws.model.tests.test_stats.MyTestCase) ... ok
test_calc_zone_pressures (vaws.model.tests.test_zone.MyTestCase) ... ok
test_get_grid (vaws.model.tests.test_zone.MyTestCase) ... ok
test_set_differential_shieding (vaws.model.tests.test_zone.MyTestCase) ... ok
test_str2num (vaws.model.tests.test_zone.MyTestCase) ... ok

----------------------------------------------------------------------
Ran 134 tests in 131.287s

OK (skipped=1)
```

2.2.4 Documentation

The VAWS user manual is generated using the Sphinx documentation tool (http://www.sphinx-doc.org/en/1.7/). The source code of the manual is located at the subdirectory called *docs* under *vaws* root directory. Both the *html* and *pdf* versions of the manual can be generated using the commands below:

```
cd <vaws dir>/docs
make html    # for html format
make latexpdf   # for pdf format
```

The html version of the manual can be updated to the manual website (http://geoscienceaustralia.github.io/vaws) as below. Note that the <build_dir> can be set in the *Makefile* in the *docs* directory.

```
cd <build_dir>/html
git checkout gh-pages
git commit -m <commit_message>
git push origin gh-pages
```

The pdf version of the manaual can be accessed at https://github.com/GeoscienceAustralia/vaws/tree/master/manaual.pdf.

CHAPTER

THREE

INPUT AND OUTPUT

The input data for a scenario consists of a configuration file and a large number of files located in three different directories. This chapter provides details of input data using the template of default scenario, which can be downloaded from https://github.com/GeoscienceAustralia/vaws/blob/master/scenarios/default. The folder structure of the default scenario is shown Listing 3.1, which consists of a configuration file (default.cfg) and input directory with three sub-directories (debris, gust_envelope_profiles, and house). The output file named *results.h5* is located in the output directory.

Listing 3.1: Folder structure

```
default
+-- default.cfg
+-- input
    +-- debris
    |   +-- debris.csv
    |
    +-- gust_envelope_profiles
    |   +-- cyclonic_terrain_cat2.csv
    |   +-- cyclonic_terrain_cat2.5.csv
    |   +-- cyclonic_terrain_cat3.csv
    |   +-- non_cyclonic.csv
    |
    +-- house
        +-- house_data.csv
        +-- conn_groups.csv
        +-- conn_types.csv
        +-- connections.csv
        +-- zones.csv
        +-- zones_cpe_mean.csv
        +-- zones_cpe_str_mean.csv
        +-- zones_cpe_eave_mean.csv
        +-- zones_edge.csv
        +-- coverages.csv
        +-- coverage_types.csv
        +-- coverages_cpe.csv
        +-- influences.csv
        +-- influence_patches.csv
        +-- damage_costing_data.csv
        +-- damage_factorings.csv
        +-- water_ingress_costing_data.csv
        +-- footprint.csv
        +-- front_facing_walls.csv
+-- output
   +-- results.h5
```

3.1 Configuration file

Each simulation requires a configuration file where basic parameter values for the simulation are provided. The configuration file can be created either by editing the template configuration file using a text editor or through GUI.

The configuration file consists of a number of sections, among which *main* and *options* are mandatory while others are optional. An example configuration file is shown in Listing 3.2.

Listing 3.2: Example configuration file: default.cfg

```
[main]
no_models = 10
model_name = Group 4 House
random_seed = 0
wind_direction = S
wind_speed_min = 55
wind_speed_max = 101
wind_speed_increment = 0.1
wind_profiles = 'cyclonic_terrain_cat2.csv'
regional_shielding_factor = 1.0

[options]
debris = True
differential_shielding = False
water_ingress = True
    save_heatmaps = True

[debris]
region_name = Capital_city
staggered_sources = False
source_items = 250
boundary_radius = 24.0
building_spacing = 20.0
debris_radius = 200
debris_angle = 45

[construction_levels]
levels = low, medium, high
probs = 0.33, 0.34, 0.33
mean_factors = 0.9, 1.0, 1.1
cv_factors = 0.58, 0.58, 0.58

[water_ingress]
thresholds = 0.1, 0.2, 0.5
speed_at_zero_wi = 40.0, 35.0, 0.0, -20.0
speed_at_full_wi = 60.0, 55.0, 40.0, 20.0

[fragility_thresholds]
states = slight, medium, severe, complete
thresholds = 0.02, 0.1, 0.35, 0.9

[heatmap]
vmin = 54.0
vmax = 95.0
vstep = 21.0
```

3.1.1 Main section

Parameters of the main section are listed in Table 3.1. In the GUI window, they are displayed in the Scenario tab as box shown in Fig. 3.1.

Table 3.1: Parameters of the main section

Name	Name in GUI	Description
no_models	Number of models	number of models
model_name	Model name	name of model
random_seed	Random seed	a number used to initialize a pseudorandom number generator
wind_profiles	Wind profiles	file name of wind profile
regional_shielding_factor	Regional shielding	regional shielding factor (default: 1.0)
wind_speed_min	Wind speed min	minimum wind speed (m/s)
wind_speed_max	Wind speed max	maximum wind speed (m/s)
wind_speed_increment	Wind speed incr.	the magnitude of the wind speed increment (m/s)
wind_direction	Wind dir.	wind direction (S, SW, W, NW, N, NE, E, SE, or RANDOM)

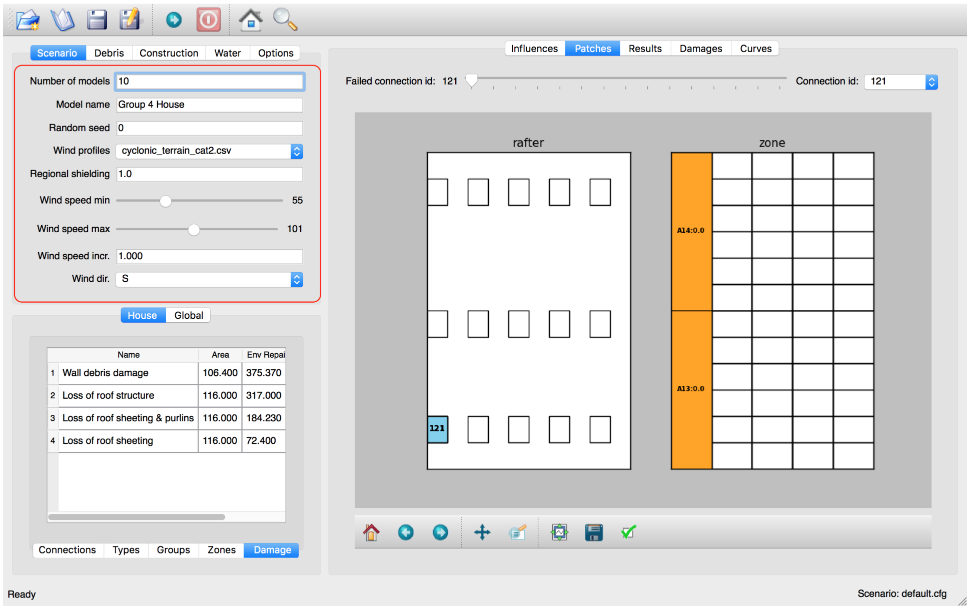

Fig. 3.1: Parameters of main section in the Scenario tab

3.1.2 Options section

Parameters of the Options section are listed in Table 3.2. Note that all the parameter values of the option section should be chosen between *True (or 1)* or *False (or 0)*. In the GUI window, they are displayed in the Debris, Water, Construction, and Options tab as listed in the Table 3.2.

Table 3.2: Parameters of options section

Name	Name in GUI	Description
debris	'Enabled' tick box in the Debris tab	if True then debris damage will be simulated.
differential_shielding	'Differential shielding' tick box in the Options tab	if True then differential shielding effect is applied.
water_ingress	'Enabled' tick box in the Water tab	if True then damage due to water ingress will be simulated.
construction_levels	'Enabled' tick box in the Construction tab	if True then construction level will be sampled.
save_heatmaps	'Save heatmaps' tick box in the Options tab	if True then heatmap plot of each model will be saved.

3.1.3 Debris section

Parameters of the debris section are listed in Table 3.3. Note that debris section is only required if *debris* is set to be *True* in the options. In the GUI window, they are displayed in the Debris tab as box shown in Fig. 3.2.

Table 3.3: Parameters of debris section

Name	Name in GUI	Description
region_name	Region	one of the region names defined in the Listing 3.3. Each region has different debris source characteristics.
building_spacing	Building spacing	distance between debris sources (m)
debris_radius	Radius	radius (in metre) of debris sources from the modelled house
debris_angle	Angle	included angle (in degree) of the sector in which debris sources exist
source_items	Source items	number of debris items per debris sources
boundary_radius	Boundary	radius (in metre) of boundary for debris impact assessment
staggered_sources	Staggered sources	if True then staggered sources are used. Otherwise, a grid pattern of debris sources are used.

3.1.4 Construction_levels section

Parameters of the construction_levels section are listed in Table 3.4. In the GUI window, they are dispalyed in the Construction tab as box shown in Fig. 3.3. The parameters are used as shown in (5.1).

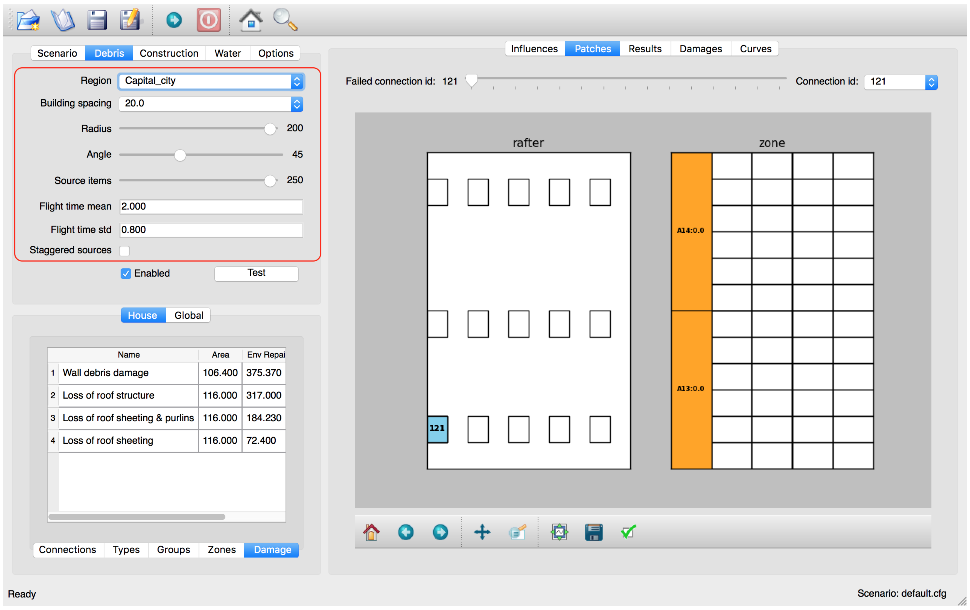

Fig. 3.2: Parameters of debris section in Debris tab

Table 3.4: Parameters of construction_level section

Name	Name in GUI	Description
levels	Levels	comma separated list of construction levels (default: low, medium, high)
probabilities	probabilities	comma separated list of probabilities of a modelled house being of a construction level (default: 0.33, 0.34, 0.33)
mean_factors	Mean factors	comma separated list of mean factors of construction levels (default: 0.9, 1.0, 1.1)
cv_factors	CV factors	comma separated list of CV factors of construction levels (default: 0.58, 0.58, 0.58)

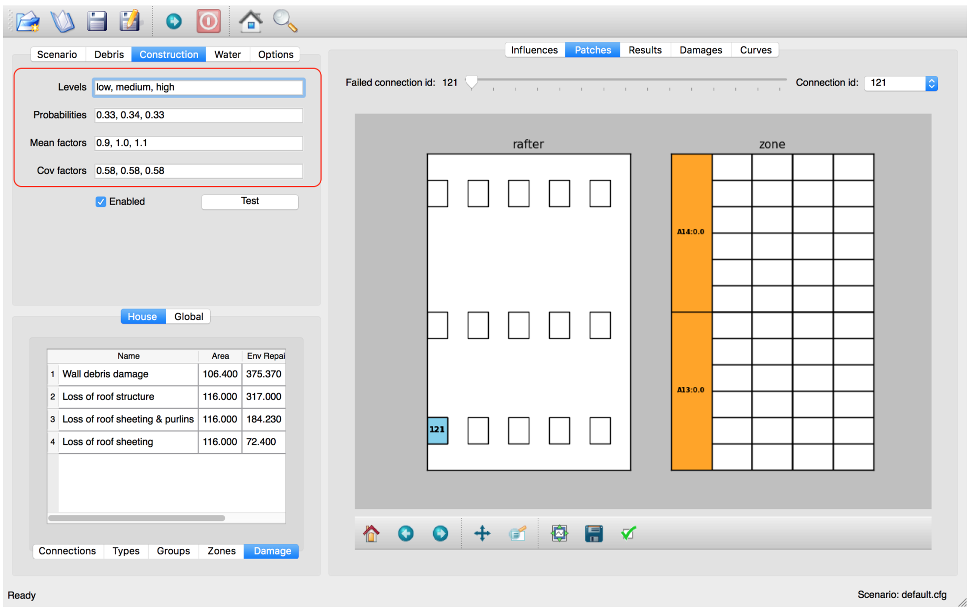

Fig. 3.3: Parameters of construction_levels section in Construction tab

3.1.5 Water_ingress section

Parameters of the water_ingress section are listed in Table 3.5. In the GUI window, they are displayed in the Water tab as box shown in Fig. 3.5. The thresholds define a lower limit of envelope damage index above which the relevant water ingress vs wind speed curve is applied. The speeds at 0% water ingress and speeds at 100% water ingress define cumulative normal distribution used to relate percentage water ingress to wind speed as shown in Fig. 3.4.

Table 3.5: Parameters of water_ingress section

Name	Name in GUI	Description
thresholds	DI thresholds	comma separated list of thresholds of damage indices (default: 0.0, 0.1, 0.2, 0.5)
speed_at_zero_wi	Speeds at 0% WI	comma separated list of maximum wind speed at no water ingress (default: 40.0, 35.0, 0.0, -20.0)
speed_at_full_wi	Speeds at 100% WI	comma separated list of minimum wind speed at full water ingress (default: 60.0, 55.0, 40.0, 20.0)

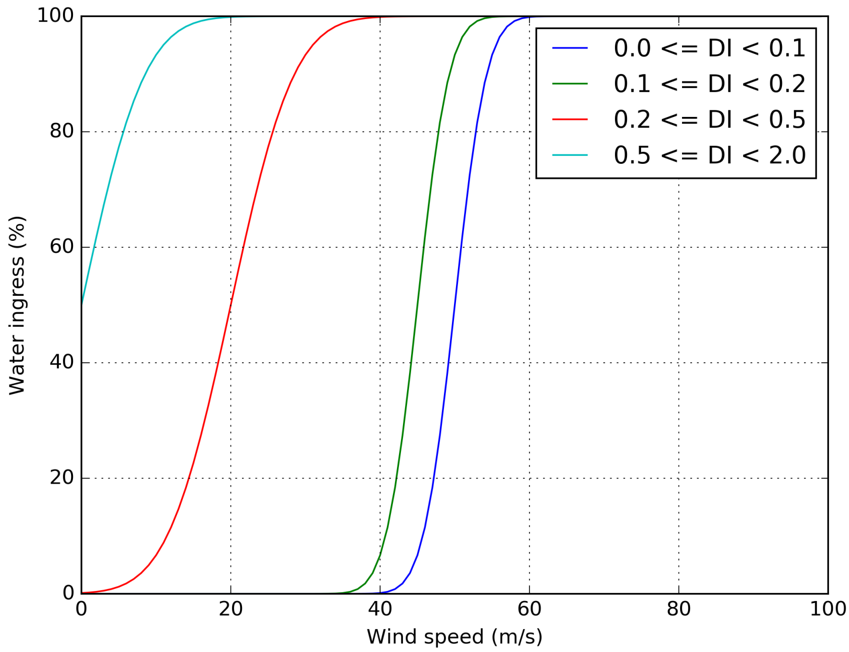

Fig. 3.4: Water ingress vs. wind speed for different ranges of damage index

3.1.6 Fragility_thresholds

Parameters of the fragility_thresholds section are listed in Table 3.6. In the GUI window, they are displayed in the Options tab as box shown in Fig. 3.6. The fragility thresholds are used as shown in (5.21).

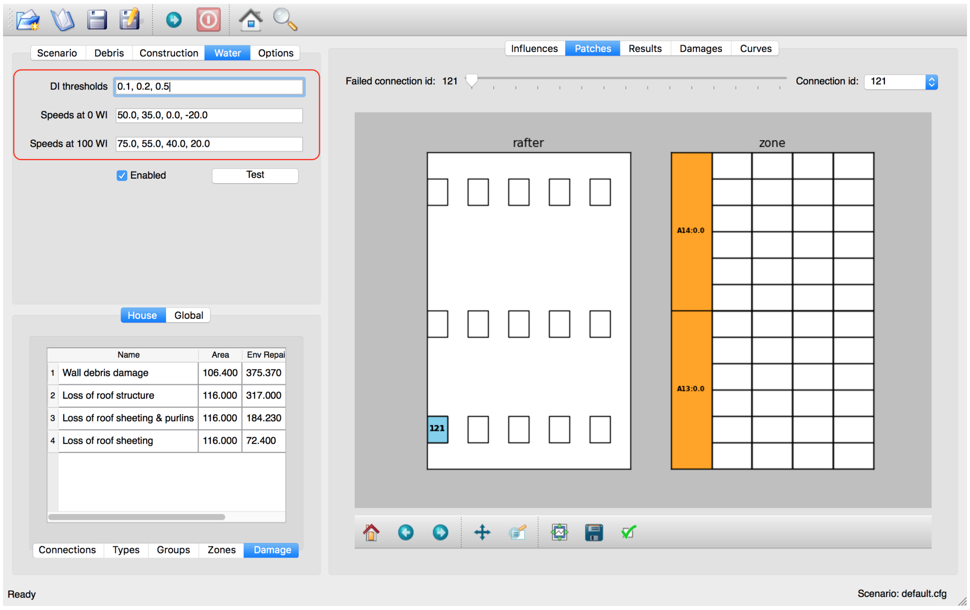

Fig. 3.5: Parameters of water_ingress section in Water tab

Table 3.6: Parameters of fragility_thresholds section

Name	Name in GUI	Description
states	Damage states	comma separated list of damage states (default: slight, medium, severe, complete)
thresholds	Thresholds	comma separated list of damage states thresholds (default: 0.02, 0.1, 0.35, 0.9)

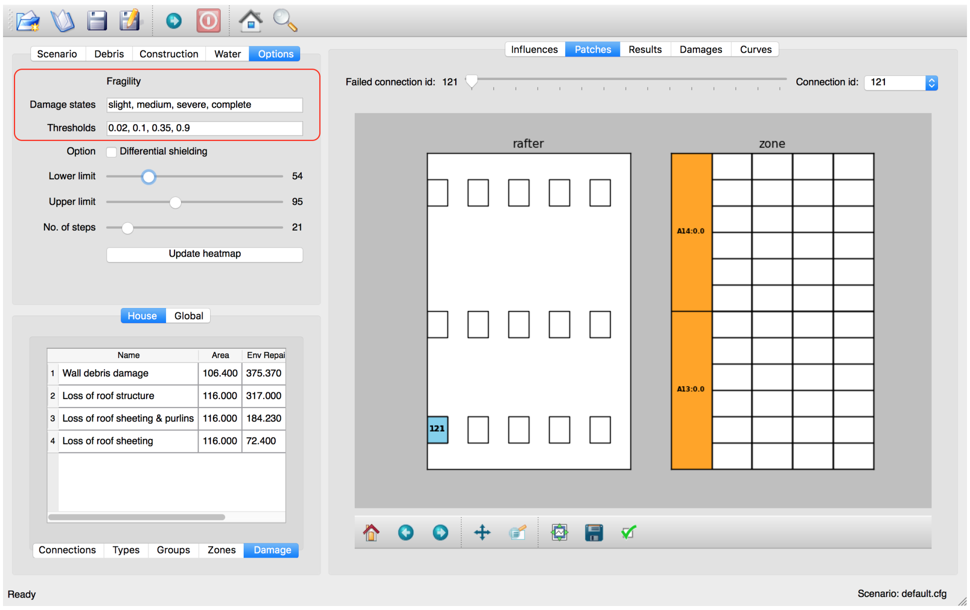

Fig. 3.6: Parameters of fragility_thresholds section in Options tab

3.1.7 Heatmap

Parameters of the heatmap section are listed in Table 3.7. In the GUI window, they are displayed in the Options tab as box shown in Fig. 3.7

Table 3.7: Parameters of heatmap section

Name	Name in GUI	Description
vmin	Lower limit	lower limit of wind speed for heatmap
vmax	Upper limit	upper limit of wind speed for heatmap
vstep	No. of steps	number of steps

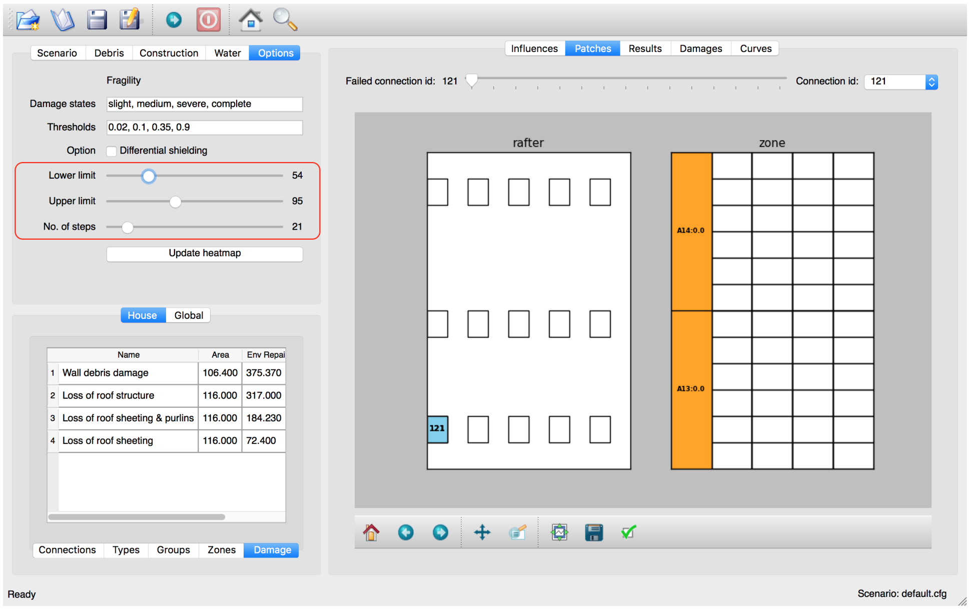

Fig. 3.7: Parameters of heatmap section in Options tab

3.2 Input file under *debris* directory

In the debris directory, *debris.csv* is located where parameter values related to windborne debris are defined. Three types of windborne debris are modelled, as listed in Table 3.8, which include *Compact*, *Rod*, and *Sheet*. Parameter values for each debris type needs to be defined by unique region name, and the defined region name should be referenced in the configuration file.

An example *debris.csv* is shown in Listing 3.3, in which debris parameters are defined for both *Capital_city* and *Tropical_town*. Note that *Capital_city* is referenced in the example configuration file Listing 3.2.

Listing 3.3: Example debris.csv

```
Region name,Capital_city,Tropical_town
Compact_ratio,20,15
Compact_mass_mean,0.1,0.1
Compact_mass_stddev,0.1,0.1
Compact_frontal_area_mean,0.002,0.002
Compact_frontal_area_stddev,0.001,0.001
```

(continues on next page)

(continued from previous page)

```
Compact_cdav,0.65,0.65
Compact_flight_time_mean, 2.0, 2.0
Compact_flight_time_stddev, 0.8, 0.8
Rod_ratio,30,40
Rod_mass_mean,4,4
Rod_mass_stddev,2,2
Rod_frontal_area_mean,0.1,0.1
Rod_frontal_area_stddev,0.03,0.03
Rod_cdav,0.8,0.8
Rod_flight_time_mean, 2.0, 2.0
Rod_flight_time_stddev, 0.8, 0.8
Sheet_ratio,50,45
Sheet_mass_mean,3,10
Sheet_mass_stddev,0.9,5
Sheet_frontal_area_mean,0.1,1
Sheet_frontal_area_stddev,0.03,0.3
Sheet_cdav,0.9,0.9
Sheet_flight_time_mean, 2.0, 2.0
Sheet_flight_time_stddev, 0.8, 0.8
```

Table 3.8: Debris types

Name	Examples
Compact	Loose nails screws, washers, parts of broken tiles, chimney bricks, air conditioner units
Rod	Parts of timber battens, purlins, rafters
Sheet	Roof cladding (mainly tiles, steel sheet, flashing, solar panels)

The parameter values should be provided for each of the debris types as set out in Table 3.9.

Table 3.9: Parameters for each debris item

Name	Note
ratio	proportion out of debris in percent
mass_mean	mean of mass (kg)
mass_stddev	standard deviation of mass (kg)
frontal_area_mean	mean of frontal area (m^2)
frontal_area_stddev	standard deviation of frontal area (m^2)
cdav	average drag coefficient
flight_time_mean	mean of flight time
frontal_area_stddev	standard deviation of flight time

3.3 Input files under *gust_envelope_profiles* directory

The gust envelope profiles are defined under *gust_envelope_profiles* directory. In the configuration file, file name of the gust envelope profile needs to be referenced as shown in Listing 3.2.

Example files are provided with respect to Australian wind design categories: cyclonic_terrain_cat2.csv, cyclonic_terrain_cat2.5.csv, cyclonic_terrain_cat3.csv, and non_cyclonic.csv, which are recommended in JDH Consulting, 2010 *[3]*.

An example of gust envelope profile is provided in Listing 3.4, and the corresponding plot is shown in Fig. 3.8.

Listing 3.4: Example of gust_envelope_profile

```
# Terrain Category 2
3,0.908,0.896,0.894,0.933,0.884,0.903,0.886,0.902,0.859,0.927
5,0.995,0.980,0.946,0.986,0.962,1.010,0.978,0.970,0.945,0.990
7,0.994,1.031,1.010,0.986,0.982,0.987,0.959,0.984,0.967,0.998
10,1.000,1.000,1.000,1.000,1.000,1.000,1.000,1.000,1.000,1.000
12,1.056,1.025,1.032,1.033,0.998,1.043,0.997,1.008,1.005,1.027
15,1.058,1.059,1.028,1.069,1.048,1.076,1.016,1.027,1.021,1.039
17,1.092,1.059,1.079,1.060,1.042,1.053,1.046,1.045,1.047,1.102
20,1.110,1.103,1.037,1.068,1.088,1.107,1.068,1.106,1.098,1.103
25,1.145,1.151,1.069,1.091,1.089,1.196,1.126,1.113,1.099,1.142
30,1.245,1.188,1.177,1.178,1.192,1.199,1.179,1.165,1.127,1.203
```

The first row is header, and heights (in metre) are listed in the first column. Profile values along the heights are listed from the second column with comma separation. One wind profile (one column) will be randomly selected for each run of the simulation.

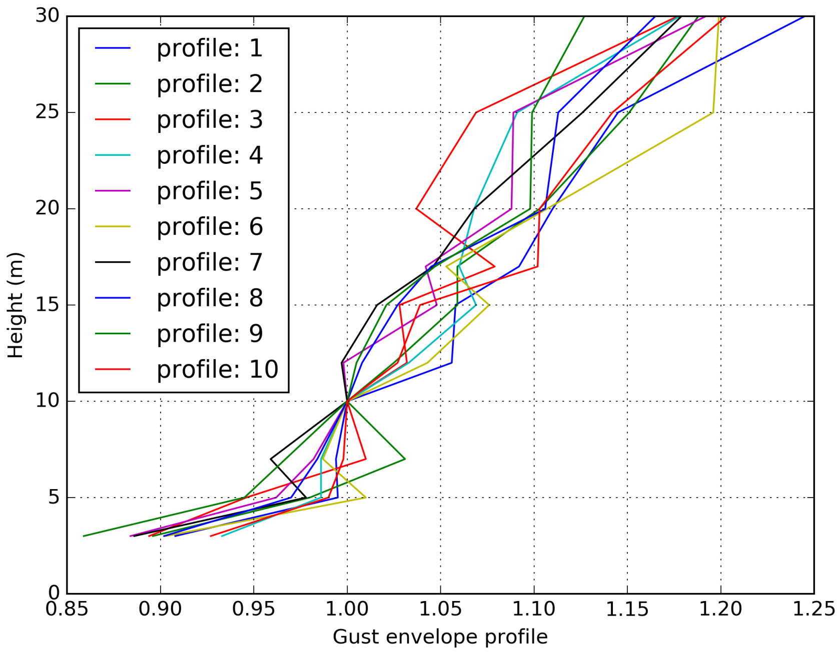

Fig. 3.8: Wind gust envelope profile along height.

3.4 Input files under *house* directory

In the house directory, a large number of files are located which are required to set parameter values of the model. The simulation model is assumed to consist of connections and zones. The connections are grouped into a number of connection types, and the connection types are further grouped into connection groups.

3.4.1 house_data.csv

This file defines parameter values for the model such as replacement cost and dimensions. An example is shown in Listing 3.5, and description of each of the parameter values are provided in Table 3.10.

Listing 3.5: Example house_data.csv

```
replace_cost,3220.93
height,4.5
length,0.9
width,9.0
cpe_cv,0.0
cpe_k,0.1
cpe_str_cv,0.0
cpe_str_k,0.1
```

Table 3.10: Parameters in the house_data.csv

Name	Type	Description
replace_cost	float	replacement cost of the model ($)
height	float	height of the model (in metre)
length	float	length of the model (in metre)
width	float	width of the model (in metre)
cpe_cv	float	CV of C_{pe} for cladding elements such as sheeting and batten
cpe_k	float	shape factor of C_{pe} for cladding elements such as sheeting and batten
cpe_str_cv	float	CV of $C_{pe,str}$ for structural elements such as rafter
cpe_str_k	float	shape factor of $C_{pe,str}$ for structural elements as rafter

3.4.2 conn_groups.csv

The model is assumed to consist of a number of connection groups. This file defines connection groups and parameter values of the each connection group. An example is shown in Listing 3.6, and description of each of the parameter values are provided in Table 3.11. Note that pre-defined group names need to be used for display of heatmap in the GUI, which are *sheeting*, *tiles*, *batten*, *rafter*, *truss*, *wallcladding*, *wallracking_cladding*, *wallracking_bracing*, *wall_collapse*, and *piersgroup*.

Listing 3.6: Example conn_groups.csv

```
group_name,dist_order,dist_dir,damage_dist,damage_scenario,trigger_collapse_at,
↪flag_pressure
sheeting,1,col,1,Loss of roof sheeting,0.0,cpe
batten,2,row,1,Loss of roof sheeting & purlins,0.0,cpe
rafter,3,patch,1,Loss of roof structure,0.0,cpe_str
```

Table 3.11: Parameters in the conn_groups.csv

Name	Type	Description
group_name	string	name of connections group
dist_order	integer	order of checking damage
dist_dir	string	direction of damage distribution; either 'col', 'row', 'patch', or 'none
damage_dist	integer	1 if load distribution is applied when connection is damaged otherwise 0
damage_scenario	string	damage scenario name defined in *damage_costing_data.csv*
trigger_collapse_at	float	proportion of damaged connections of the group at which a model is deemed to be collapsed. 0 if ignored
flag_pressure	string	type of C_{pe} for pressure calculation; either 'cpe' or 'cpe_str'

3.4.3 conn_types.csv

A connection group may consists of a number of connection types which have different parameter values for strength, dead load, and costing area. This file defines connection types and parameter values of the each connection type. An example is shown in Listing 3.7, and description of each of the parameter values are provided in Table 3.12.

Listing 3.7: Example conn_types.csv

```
type_name,strength_mean,strength_std,dead_load_mean,dead_load_std,group_name,
↪costing_area
sheetinggable,1.54,0.16334,0.02025,0.0246,sheeting,0.405
sheetingeave,4.62,0.28292,0.02025,0.0246,sheeting,0.405
sheetingcorner,2.31,0.2,0.01013,0.0246,sheeting,0.225
sheeting,2.695,0.21608,0.0405,0.0246,sheeting,0.81
batten,3.6,1.26,0.089,0.0708,batten,0.81
battenend,3.6,1.26,0.089,0.0708,batten,0.405
batteneave,3.6,1.26,0.089,0.0708,batten,0.405
battencorner,3.6,1.26,0.089,0.0708,batten,0.225
endraftertopplate,19.5,5.85,0.84,0.063,rafter,1.238
endrafterridge,16.5,4.95,1.8,0.135,rafter,1.665
collarraftertopplate,19.5,5.85,1.68,0.126,rafter,1.845
collarrafterridge,16.5,4.95,1.13,0.08475,rafter,1.26
collarraftercollar,2.4,0.48,3.95,0.29625,rafter,1.665
plainraftertopplate,19.5,5.85,1.68,0.126,rafter,2.475
plainrafterridge,16.5,4.95,3.6,0.27,rafter,3.33
weakbatten,3.6,1.26,0.089,0.0708,batten,0.81
```

Table 3.12: Parameters in the conn_types.csv

Name	Type	Description
type_name	string	name of connection type
strength_mean	float	mean strength (kN)
strength_std	float	standard deviation of strength
dead_load_mean	float	mean dead load (kN)
dead_load_std	float	standard deviation of dead load
group_name	string	name of connections group
costing_area	float	costing area (m^2)

3.4.4 connections.csv

This file defines connections and parameter values of the each connection. An example is shown in Listing 3.8, and description of each of the parameter values are provided in Table 3.13.

Listing 3.8: Example connections.csv

```
conn_name,type_name,zone_loc,section,coords
1,sheetingcorner,A1,1,0,0,0.2,0,0.2,0.5,0,0.5
2,sheetinggable,A2,1,0,0.5,0.2,0.5,0.2,1,0,1
3,sheetinggable,A3,1,0,1,0.2,1,0.2,1.5,0,1.5
4,sheetinggable,A4,1,0,1.5,0.2,1.5,0.2,2,0,2
5,sheetinggable,A5,1,0,2,0.2,2,0.2,2.5,0,2.5
```

Table 3.13: Parameters in the connections.csv

Name	Type	Description
conn_name	string	name of connection
type_name	string	name of connection type
zone_loc	integer	zone name corresponding to connection location
section	integer	index of section in which damage distribution occurs
coords	float	comma separated values of x, y coordinates for plotting purpose. e.g., 4 sets for a rectangular shape, 3 sets for a triangular shape.

3.4.5 zones.csv

This file defines zones and parameter values of the each zone. An example is shown in Listing 3.9, and description of each of the parameter values are provided in Table 3.14.

Listing 3.9: Example zones.csv

```
name,area,cpi_alpha,coords,
A1,0.2025,0,0,0,0.2,0,0.2,0.5,0,0.5
A2,0.405,0.5,0,0.5,0.2,0.5,0.2,1,0,1
A3,0.405,1,0,1,0.2,1,0.2,1.5,0,1.5
A4,0.405,1,0,1.5,0.2,1.5,0.2,2,0,2
A5,0.405,1,0,2,0.2,2,0.2,2.5,0,2.5
```

Table 3.14: Parameters in the zones.csv

Name	Type	Description
name	string	name of zone
area	float	area of zone (m^2)
cpi_alpha	float	proportion of the zone's area to which internal pressure is applied
coords	float	comma separated list of x, y coordinates for plotting purpose. e.g., 4 sets for a rectangular shape, 3 sets for a triangular shape.

3.4.6 zones_cpe_mean.csv

This file defines mean cladding C_{pe} of each zone with regard to the eight wind directions. An example is shown in Listing 3.10, and description of each of the parameter values are provided in Table 3.15.

Listing 3.10: Example zones_cpe_mean.csv

```
name,S,SW,W,NW,N,NE,E,SE
A1,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A3,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A4,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A5,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A6,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2
A7,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A8,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A9,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A10,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A11,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A12,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5
A13,0,0,0,0,0,0,0,0
A14,0,0,0,0,0,0,0,0
```

Table 3.15: Parameters in the zones_cpe_mean.csv

Name	Type	Description
name	string	name of zones
S	float	mean cladding C_{pe} value in South direction
SW	float	mean cladding C_{pe} value in South West direction
W	integer	mean cladding C_{pe} value in West direction
NW	float	mean cladding C_{pe} value in North East direction
N	float	mean cladding C_{pe} value in North direction
NE	float	mean cladding C_{pe} value in North East direction
E	integer	mean cladding C_{pe} value in East direction
SE	float	mean cladding C_{pe} value in South East direction

3.4.7 zones_cpe_str_mean.csv

Like zones_cpe_mean.csv, mean $C_{pe,str}$ values for zones associated with structural component (e.g., rafter) need to be provided in zones_cpe_str_mean.csv. An example is shown in Listing 3.11.

Listing 3.11: Example zones_cpe_str_mean.csv

```
name,S,SW,W,NW,N,NE,E,SE
A1,0,0,0,0,0,0,0,0
A2,0,0,0,0,0,0,0,0
A3,0,0,0,0,0,0,0,0
A4,0,0,0,0,0,0,0,0
A5,0,0,0,0,0,0,0,0
A6,0,0,0,0,0,0,0,0
A7,0,0,0,0,0,0,0,0
A8,0,0,0,0,0,0,0,0
A9,0,0,0,0,0,0,0,0
A10,0,0,0,0,0,0,0,0
A11,0,0,0,0,0,0,0,0
A12,0,0,0,0,0,0,0,0
A13,-1,-1,-1,-1,-1,-1,-1,-1
A14,-0.4,-0.4,-0.4,-0.4,-0.4,-0.4,-0.4,-0.4
```

3.4.8 zones_cpe_eave_mean.csv

Like zones_cpe_mean.csv, mean C_{pe} values for zones at eave need to be provided in zones_cpe_eave_mean.csv. An example is shown in Listing 3.12.

Listing 3.12: Example zones_cpe_eave_mean.csv

```
name,S,SW,W,NW,N,NE,E,SE
A1,0.7,0.7,0.7,0.7,0.7,0.7,0.7,0.7
A2,0.35,0.35,0.35,0.35,0.35,0.35,0.35,0.35
A3,0,0,0,0,0,0,0,0
A4,0,0,0,0,0,0,0,0
A5,0,0,0,0,0,0,0,0
A6,0,0,0,0,0,0,0,0
A7,0,0,0,0,0,0,0,0
A8,0,0,0,0,0,0,0,0
A9,0,0,0,0,0,0,0,0
A10,0,0,0,0,0,0,0,0
A11,-0.1,-0.1,-0.1,-0.1,-0.1,-0.1,-0.1,-0.1
A12,-0.2,-0.2,-0.2,-0.2,-0.2,-0.2,-0.2,-0.2
A13,0.07,0.07,0.07,0.07,0.07,0.07,0.07,0.07
A14,-0.02,-0.02,-0.02,-0.02,-0.02,-0.02,-0.02,-0.02
```

3.4.9 zones_edge.csv

In zones_edge.csv, for each of the eight direction, 1 is provided for zone within the region of a roof edge, otherwise 0. Zones in the edge region are considered to be subjected to differential shielding if enabled by user. An example is shown in Listing 3.13.

Listing 3.13: Example zones_edge.csv

```
name,S,SW,W,NW,N,NE,E,SE
A1,1,1,1,0,0,0,0,0
A2,1,1,1,0,0,0,0,0
A3,1,1,1,0,0,0,0,0
A4,0,1,0,0,0,0,0,0
A5,0,1,0,0,0,0,0,0
A6,0,1,0,0,0,0,0,0
```

(continues on next page)

(continued from previous page)

```
A7,0,0,0,1,0,0,0,0
A8,0,0,0,1,0,0,0,0
A9,0,0,0,1,0,0,0,0
A10,0,0,1,1,1,0,0,0
A11,0,0,1,1,1,0,0,0
A12,0,0,1,1,1,0,0,0
A13,1,1,1,0,0,0,0,0
A14,0,0,1,1,1,0,0,0
```

3.4.10 coverages.csv

This file defines coverages making up the wall part of the envelope of the model. An example is shown in Listing 3.14, and description of each of the parameter values are provided in Table 3.16. The wall name is defined in Listing 3.23. Coverages are assessed for debris impact damage and failure by direct wind pressure. The area of coverage is used in the calculation of C_{pi}.

Listing 3.14: Example coverages.csv

```
name,description,wall_name,area,coverage_type
1,window,1,3.6,Glass_annealed_6mm
2,door,1,1.8,Timber_door
3,window,1,1.89,Glass_annealed_6mm
4,window,1,1.89,Glass_annealed_6mm
```

Table 3.16: Parameters in the coverages.csv

Name	Type	Description
name	integer	coverage index
description	string	description
wall_name	integer	wall name
area	float	area (m^2)
coverage_type	string	name of coverage type

3.4.11 coverage_types.csv

This file defines types of coverages referenced in the Listing 3.14. An example is shown in Listing 3.15, and description of each of the parameter values are provided in Table 3.17.

Listing 3.15: Example coverage_types.csv

```
name,failure_momentum_mean,failure_momentum_std,failure_strength_in_mean,failure_
↪strength_in_std,failure_strength_out_mean,failure_strength_out_std
Glass_annealed_6mm,0.05,0.0,100,0.0,-100,0.0
Timber_door,142.2,28.44,100,0.0,-100,0.0
```

Table 3.17: Parameters in the coverage_types.csv

Name	Type	Description
name	string	name of coverage type
failure_momentum_mean	float	mean failure momentum (kg · m/s) for debris impact
failure_momentum_std	float	standard deviation of failure momentum
failure_strength_in_mean	float	mean failure strength inward direction (positive) for failure due to wind pressure (kN)
failure_strength_in_std	float	standard deviation of failure strength inward direction
failure_strength_out_mean	float	mean failure strength outward direction (negative) for failure due to wind pressure (kN)
failure_strength_out_std	float	standard deviation of failure strength outward direction

3.4.12 coverages_cpe.csv

Like zones_cpe_mean.csv, mean C_{pe} values for coverages are provided in coverages_cpe.csv. An example is shown in Listing 3.16.

Listing 3.16: Example coverages_cpe.csv

```
ID,S,SW,W,NW,N,NE,E,SE
1,2.4,2.4,2.4,2.4,2.4,2.4,2.4,2.4
2,1.69,1.69,1.69,1.69,1.69,1.69,1.69,1.69
3,-1.14,-1.14,-1.14,-1.14,-1.14,-1.14,-1.14,-1.14
4,-1.45,-1.45,-1.45,-1.45,-1.45,-1.45,-1.45,-1.45
5,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9
6,-0.55,-0.55,-0.55,-0.55,-0.55,-0.55,-0.55,-0.55
```

3.4.13 influences.csv

This file defines influence coefficients relating a connection with either another connection(s) or zone(s). The wind load acting on a connection can be computed as the sum of the product of influence coefficient and either wind load on zone or load on another connection. An example is shown in Listing 3.17, and description of each of the parameter values are provided in Table 3.18. In this example, connection 1 is related to the zone A1 with coefficient 1.0, and connection 61 is related to the connection 1 with coefficient 1.0. Similarly, connection 121 is related to the zone A13 with coefficient 0.81 and the zone A14 with coefficient 0.19.

Listing 3.17: Example influences.csv

```
Connection,Zone,Coefficent
1,A1,1.0
2,A2,1.0
61,1,1
62,2,1
```

(continues on next page)

(continued from previous page)

```
63,3,1
121,A13,0.81,A14,0.19
```

Table 3.18: Parameters in the influences.csv

Name	Description
Connection	name of connection
Zone	name of either zone or connection associated with the Connection
Coefficient	coefficient value

3.4.14 influence_patches.csv

This file defines influence coefficients of connections when associated connection is failed. An example is shown in Listing 3.18, and description of each of the parameter values are provided in Table 3.19. In the example, when connection 121 is failed, influence coefficients of connection 121, 122, 123 are re-defined.

Listing 3.18: Example influence_patches.csv

```
Damaged connection,Connection,Zone,Coefficient
121,121,A13,0,A14,0
121,122,A13,1,A14,0
121,123,A13,1,A14,1
122,121,A13,1,A14,0
122,122,A13,0,A14,0
122,123,A13,0,A14,1
```

Table 3.19: Parameters in the influence_patches.csv

Name	Description
Damaged Connection	name of damaged connection
Connection	name of connection for which the influence coefficients are to be updated
Zone	name of either zone or connection associated with the connection to be updated
Coefficient	new influence coefficient value

3.4.15 damage_costing_data.csv

This file defines damage scenarios referenced in Listing 3.6. An example is shown in Listing 3.19, and description of each of the parameter values are provided in Table 3.20. The damage cost for each damage scenario C is calculated as

$$C = x \times (A \times C_{\text{env}} \times R_{\text{env}} + C_{\text{int}} \times R_{\text{int}}) \tag{3.1}$$

where x: proportion of damaged area ($0 \leq x \leq 1$), A: surface area, C_{env}: costing function for envelope, R_{env}: repair rate for envelope, C_{int}: costing function for internal, and R_{int}: repair rate for internal. Two types of costing functions are defined as:

$$\begin{aligned} f_1 &= c_1 \times x^2 + c_2 \times x + c_3 \\ f_2 &= c_1 \times x^{c_2} \end{aligned} \tag{3.2}$$

Listing 3.19: Example damage_costing_data.csv

```
name,surface_area,envelope_repair_rate,envelope_factor_formula_type,envelope_
↪coeff1,envelope_coeff2,envelope_coeff3,internal_repair_rate,internal_factor_
↪formula_type,internal_coeff1,internal_coeff2,internal_coeff3,water_ingress_order
Loss of roof sheeting,116,72.4,1,0.3105,-0.8943,1.6015,0,1,0,0,0,6
Loss of roof sheeting & purlins,116,184.23,1,0.3105,-0.8943,1.6015,0,1,0,0,0,7
Loss of roof structure,116,317,1,0.3105,-0.8943,1.6015,8320.97,1,-0.4902,1.4896,0.
↪0036,3
```

Table 3.20: Parameters in the damage_costing_data.csv

Name	Description
name	name of damage scenario
surface_area	surface area (m^2)
envelope_repair_rate	repair rate for envelope damage ($/m^2)
envelope_factor_formula_type	type index of costing function for envelope
envelope_coeff1	c_1 in costing function for envelope
envelope_coeff2	c_2 in costing function for envelope
envelope_coeff3	c_3 in costing function for envelope
internal_repair_rate	repair rate for internal damage ($)
internal_factor_formula_type	type index of costing function for internal
internal_coeff1	c_1 in costing function for internal
internal_coeff2	c_2 in costing function for internal
internal_coeff3	c_3 in costing function for internal
water_ingress_order	order in applying cost induced by water ingress

3.4.16 damage_factorings.csv

This file defines a hierarchy of costings, where each row has a parent and child connection type group. When costing the parent group, all child costings will be factored out of the parent. This mechanism avoids double counting of repair costs. An example is shown in Listing 3.20.

Listing 3.20: Example damage_factorings.csv

```
ParentGroup,FactorByGroup
batten,rafter
sheeting,rafter
sheeting,batten
```

3.4.17 water_ingress_costing_data.csv

This file contains costing information of damage induced by water ingress for various scenarios of structural damage. Each row contains coefficients that are used by costing functions. An example is shown in Listing 3.21, and description of each of the parameter values are provided in Table 3.21. The water ingress cost WC is calculated as

$$WC = B \times C(x) \tag{3.3}$$

where x: envelope damage index prior to water ingress ($0 \leq x \leq 1$), B: base cost, and C: costing function. Like the damage costing functions, two types of costing functions are defined as (3.2).

Listing 3.21: Example water_ingress_costing_data.csv

```
name,water_ingress,base_cost,formula_type,coeff1,coeff2,coeff3
Loss of roof sheeting,0,0,1,0,0,1
Loss of roof sheeting,5,2989.97,1,0,0,1
Loss of roof sheeting,18,10763.89,1,0,0,1
Loss of roof sheeting,37,22125.78,1,0,0,1
Loss of roof sheeting,67,40065.59,1,0,0,1
Loss of roof sheeting,100,59799.39,1,0,0,1
```

Table 3.21: Parameters in the water_ingress_costing_data.csv

Name	Description
name	name of damage scenario
water_ingress	water ingress in percent
base_cost	base cost B
formula_type	type index of costing function
coeff1	c_1 in costing function
coeff2	c_2 in costing function
coeff3	c_3 in costing function

3.4.18 footprint.csv

This file contains information about footprint of the model. Each row contains x and y coordinates of the vertices of the footprint. An example is shown in Listing 3.22.

Listing 3.22: Example footprint.csv

```
footprint_coord
-6.5, 4.0
6.5, 4.0
6.5, -4.0
-6.5, -4.0
```

3.4.19 front_facing_walls.csv

This file contains wall information with respect to the eight wind direction. Each row contains wall name(s) for a wind direction. An example is shown in Listing 3.23.

Listing 3.23: Example front_facing_walls.csv

```
wind_dir,wall_name
S,1
SW,1,3
W,3
NW,3,5
N,5
NE,5,7
E,7
SE,1,7
```

3.5 Output file

After simulation output file named *results.h5* is created, which is in HDF5 format. Its content can be accessed via Python or HDF Viewer. Table 3.22 lists attributes in the output file. Note that time invariant attribute is one whose value is set when the model is created, and is kept the same over the range of wind speeds.

Table 3.22: Attributes saved in the output file

Attribute	Description	Note
profile_index	wind profile index	per model (time invariant)
wind_dir_index	wind direction index	per model (time invariant)
construction_level	construction quality level	per model (time invariant)
terrain_height_multiplier	terrain height multiplier	per model (time invariant)
shielding_multiplier	shielding multiplier	per model (time invariant)
mean_factor	mean factor of construction quality	per model (time invariant)
cv_factor	cv factor of construction quality	per model (time invariant)
qz	free stream wind pressure	per model
cpi	internal pressure coefficient	per model
collapse	1 if model collapse otherwise 0	per model
di	damage index	per model
di_except_water	damage index except water ingress induced damage	per model
repair_cost	repair cost	per model
water_ingress_cost	repair cost induced by water ingress	per model
window_breached_by_debris	1 if any window breached by debris otherwise 0	per model
no_items	total number of generated debris items	per debris model
no_impacts	total number of debris impacts	per debris model
damaged_area	total damaged area by debris impact	per debris model
damaged_area	total damaged area by group	per group
damaged	1 if connection is damaged otherwise 0	per connection
capacity	wind speed at which damage occurred	per connection
load	wind load	per connection
strength	strength	per connection (time invariant)
dead load	dead load	per connection (time invariant)
pressure_cpe	wind pressure for zone component related to sheeting and batten	per zone
pressure_cpe_str	wind pressure for zone component related to rafter	per zone
cpe	external pressure coefficient for zone component related to sheeting and batten	per zone (time invariant)

Continued on next page

Table 3.22 – continued from previous page

Attribute	Description	Note
cpe_str	external pressure coefficient for zone component related to rafter	per zone (time invariant)
cpe_eave	external pressure coefficient for zone component related to eave	per zone (time invariant)
strength_negative	strength in one direction	per coverage (time invariant)
strength_positive	strength in the other direction	per coverage (time invariant)
load	wind load	per coverage
breached	1 if coverage is damaged otherwise 0	per coverage
breached_area	cumulative area breached by debris	per coverage
capacity	wind speed at which breach occurred	per coverage

Fig. 3.9 shows the structure of the results when opened in the HDFView, a visual tool for browsing HDF5 files. There are 8 groups consisting of connection, coverage, debris, fragility, group, house, vulnerability, and zone. Fig. 3.10 shows a list of sub-groups under the connection: capacity, damaged, dead_load, load, and strength. Fig. 3.11 shows a dataset of capacity of the selected connection.

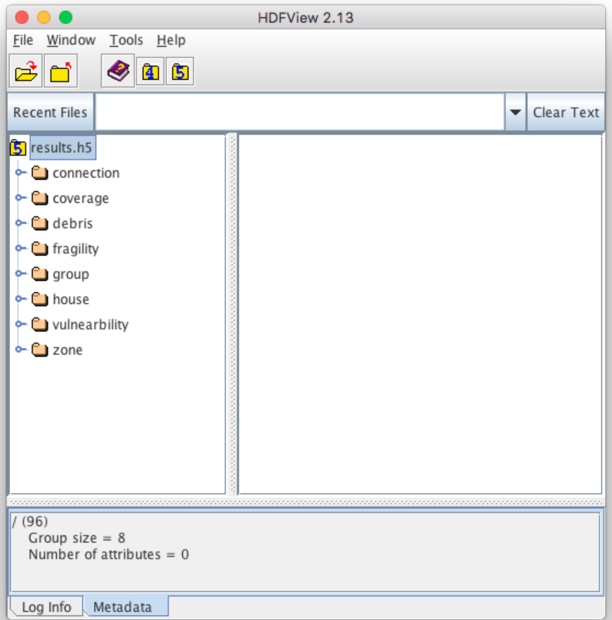

Fig. 3.9: Structures of output in the HDFView

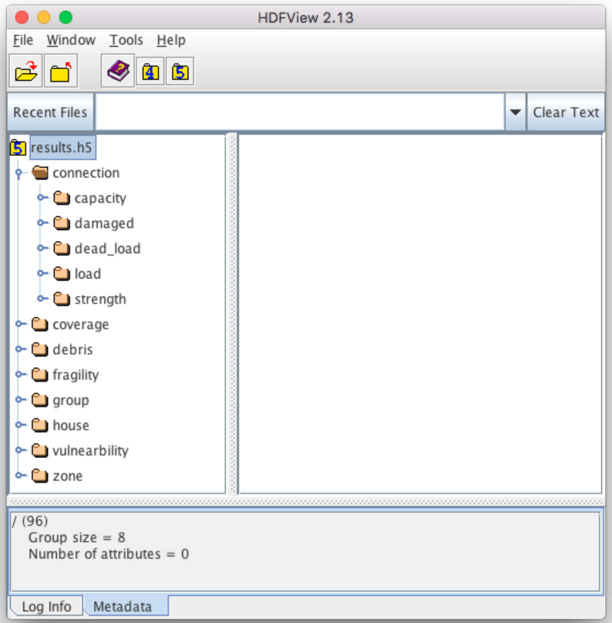

Fig. 3.10: Attributes under connection tab in the HDFView

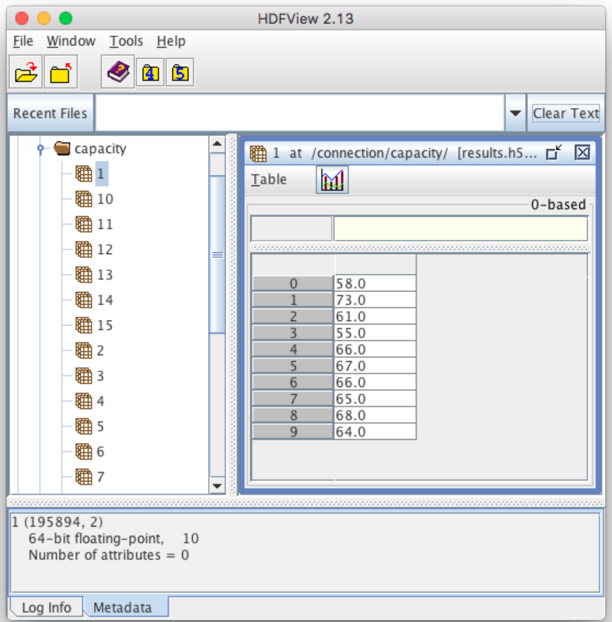

Fig. 3.11: Values of capacity of the selected connection in the HDFView

CHAPTER

FOUR

USE OF THE GUI

This chapter provides an overview of the GUI and instructions on how to run simulations using the GUI.

4.1 Structure

The main window is logically separated into distinct areas of functionality as shown as Fig. 4.1.

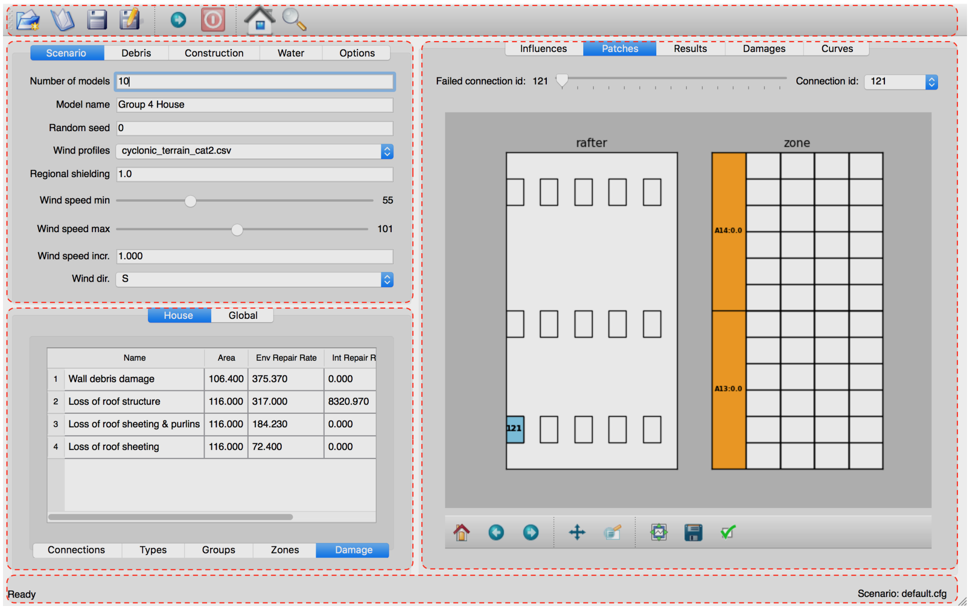

Fig. 4.1: Program main window consisting of five areas by functionality as shown as dotted box

4.1.1 Top

At the top of the main window, tool bar with buttons is located. The file menu and action corresponding to each of the buttons is set out in the Table 4.1.

Table 4.1: Buttons in the toolbar

Button	Menu	Action
	File -> New	Create a new scenario with default setting
	File -> Open Scenario	Open an existing configuration file
	File -> Save	Save current scenario
	File -> Save As	Save current scenario to a new configuration file
	Simulator -> Run	Run the scenario
	Simulator -> Stop	Stop the simulation when in progress
	Model -> House Info	Show the current house information including wall coverages

4.1.2 Top left

The top left panel contains simulation settings across five tabs: Scenario, Debris, Construction, Water, and Options, where parameter values for a simulation can be set. The details of each of the tab can be found in *3.1*.

There are three test button across the tabs.

The Test button in the Debris tab demonstrates debris generation function at a selected wind speed. Once the wind speed is determined, then a window showing debris traces from sources are displayed as shown as Fig. 4.2. The vulnerability curve used in the test function is selected from the curves shown in Fig. 4.3, whereas other parameter values can be changed through the GUI.

Table 4.2: Parameter values for vulnerability curves (4.2) used in the debris test

name	α	β
Capital_city	0.1585	3.8909
Tropical_town	0.1030	4.1825

The Test button in the Construction tab shows distribution of connection strength of the selected connection type. Example of sampled strength of batten type is shown in Fig. 4.4.

The Test button in the Water tab shows relationship between percentage of water ingress and wind speed for a range of damage index as shown in Fig. 4.5.

4.1.3 Bottom left

The bottom left panel contains data browser of house and global data. This panel contains two tabs at the top: House and Global. The House tab has five tabs at the bottom: Connections, Types, Groups, Zones, and Damage, as shown in Fig. 4.6. Table 4.3 sets out corresponding input file and section for each of the tabs.

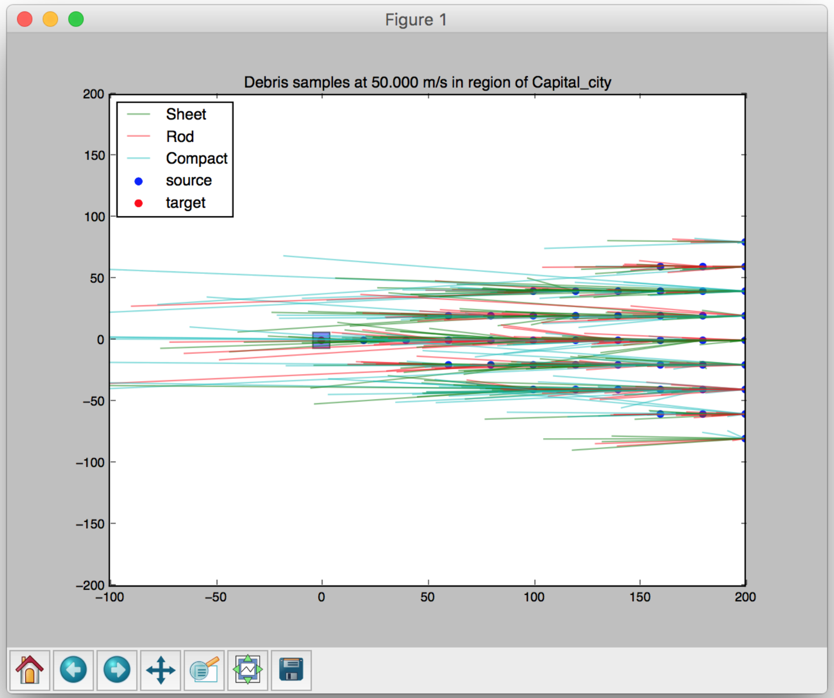

Fig. 4.2: Test of debris generation function: debris generated at 50 m/s in region of Capcital_city

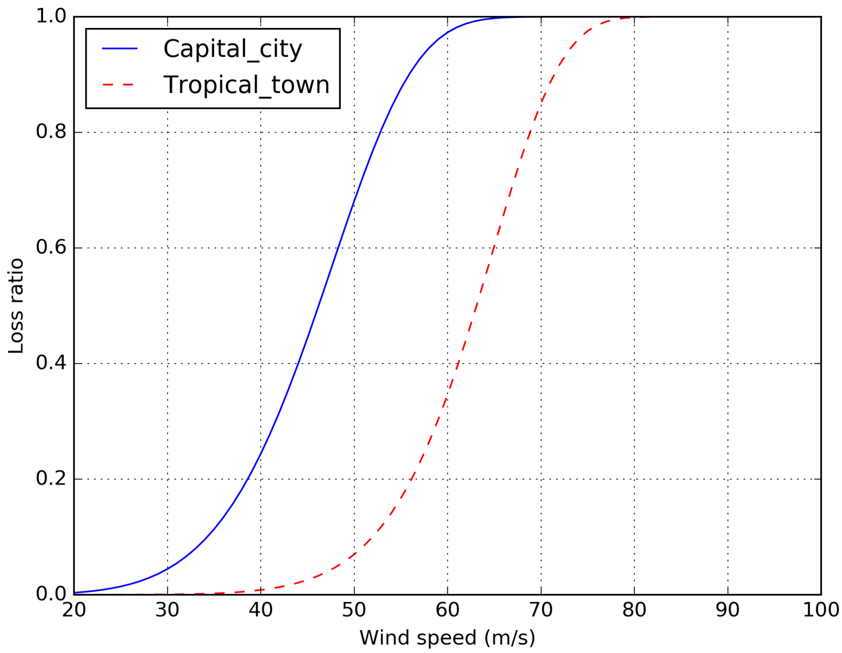

Fig. 4.3: Vulnerability curves implemented in the debris test function using the parameter values listed in Table 4.2.

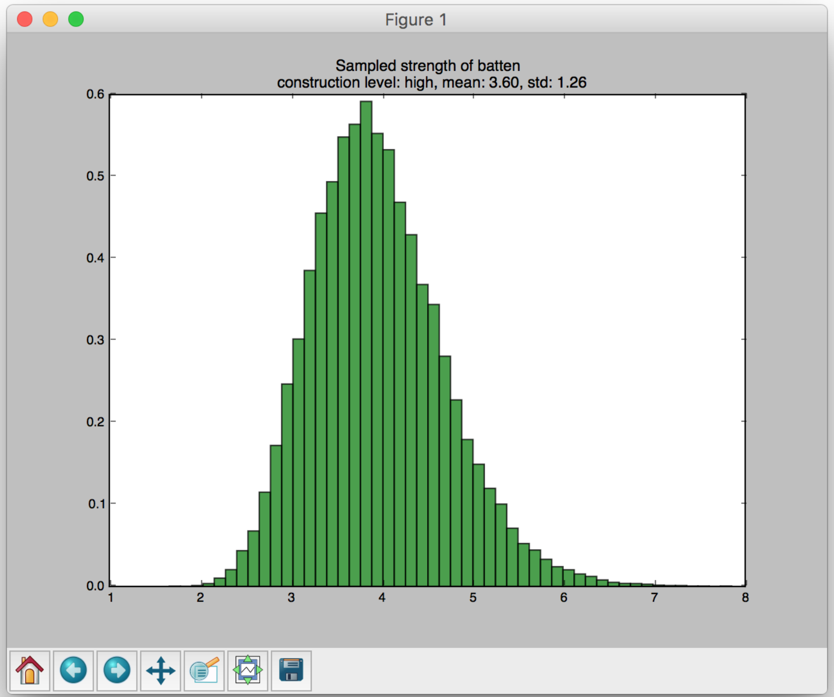

Fig. 4.4: Distribution of sampled strength of the selected connection type

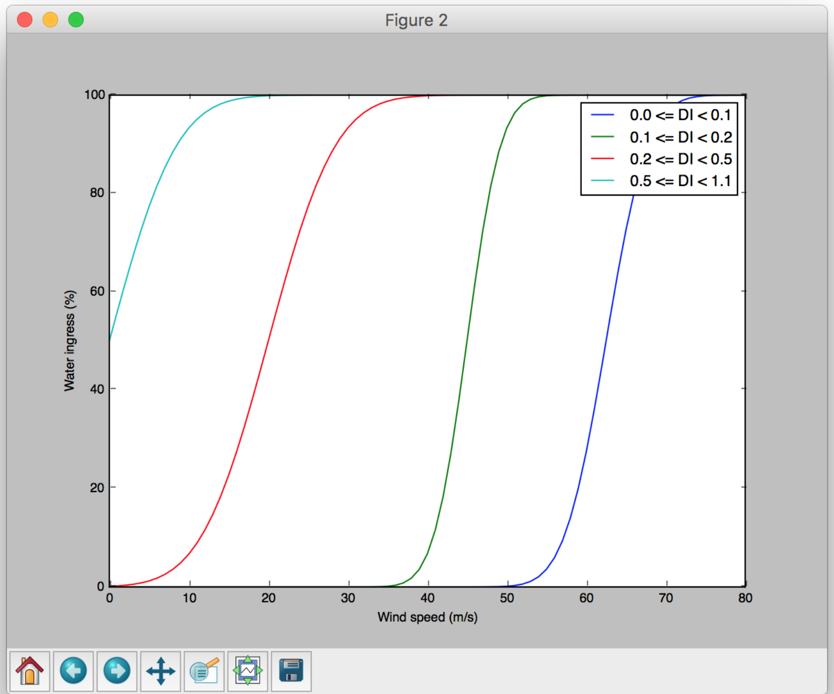

Fig. 4.5: Relationship between percentage of water ingress and wind speed

Table 4.3: House data

Tab name	Input file	Section
Connections	connections.csv	*3.4.4 connections.csv*
Types	conn_types.csv	*3.4.3 conn_types.csv*
Groups	conn_groups.csv	*3.4.2 conn_groups.csv*
Zones	zones.csv	*3.4.5 zones.csv*
Damage	damage_costing_data.csv	*3.4.15 damage_costing_data.csv*

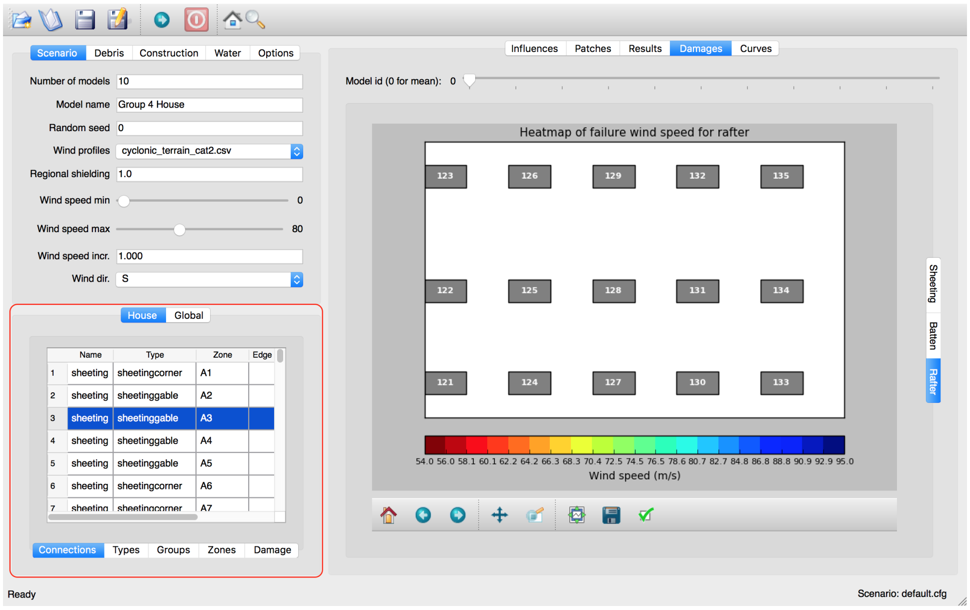

Fig. 4.6: House data tab showing connections information

The Global tab has two tabs at the bottom: Boundary Profile and Debris, as shown in Fig. 4.7. In the Boundary Profile tab, gust envelope profiles of selected wind profiles is displayed. Details about the gust envelope profiles can be found in *3.3*. In the Debris tab, parameter values for debris model listed in the debris.csv (*3.1.3*) is displayed. Note that the contents of both tabs are to be changed dynamically upon different selection of wind profile file (Wind Profiles) and debris region (Region).

4.1.4 Bottom right

The bottom right panel shows input data of influence coefficients and simulation results. This panel consists of five tabs: Influences, Patches, Results, Damages, and Curves, among which Results, Damages, and Curves are empty until a simulation is completed.

Influences tab

Once connection id is set by the slider at the top, then the selected connection (coloured in skyblue) and its associated either zone or connection (coloured in orange) with influence coefficient are shown as Fig. 4.8.

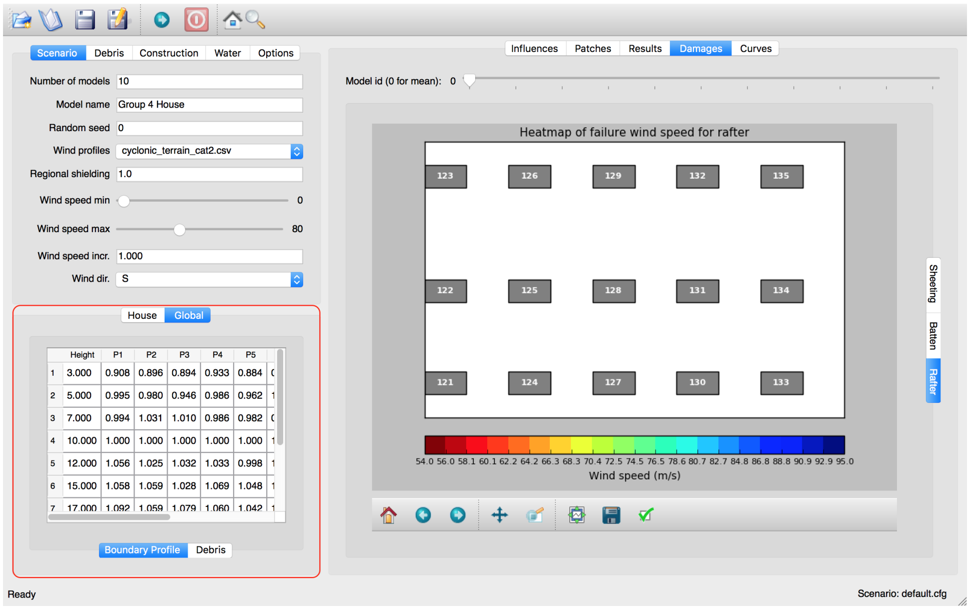

Fig. 4.7: Global data tab showing boundary profiles information

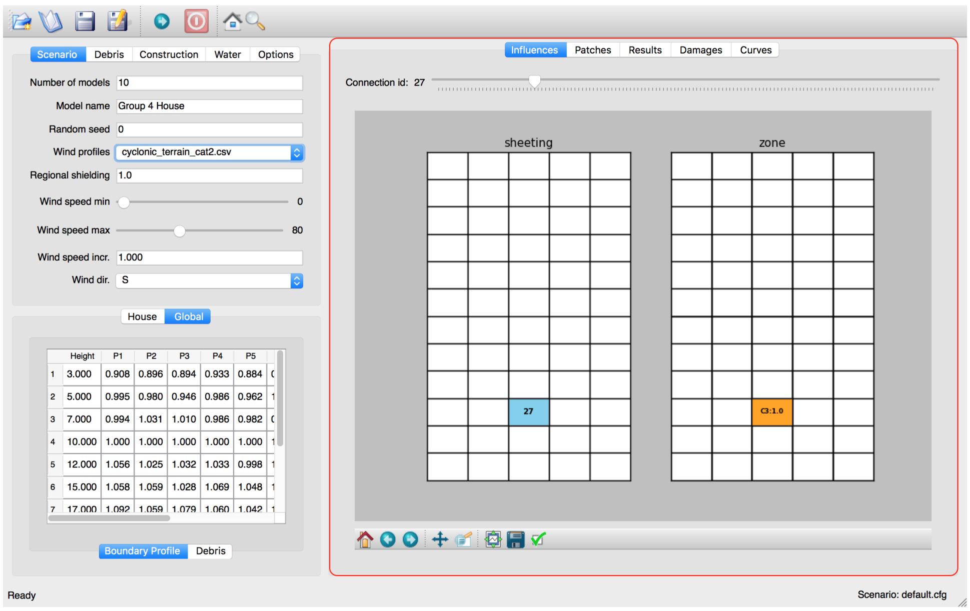

Fig. 4.8: Display of influence coefficient of connection id 27, which is 1.0 with Zone C3.

Patches tab

The Patches tab shows the influence coefficient of connection when associated connection is failed. Once failed connection (coloured in gray) and connection id (coloured in skyblue) are set, then associated either zone or connection (coloured in orange) with influence coefficient is shown as Fig. 4.9.

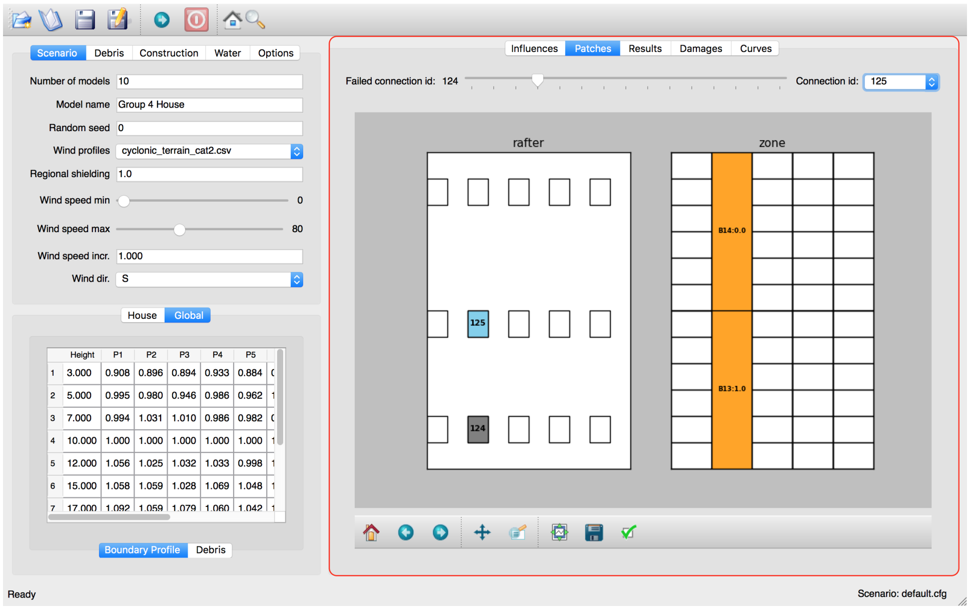

Fig. 4.9: Display of influence coefficient of connection 125 when connection 124 is failed.

Results tab

The Results tab shows the results of simulation in four sub-windows: Zones, Connections, Type Strengths, and Type Damage.

The Zones window shows sampled Cpe values for each of the zones for each realisation of the simulation models as shown as Fig. 4.10. The first string at the *House* column refers to model index, and the string before and after slash refer to wind direction and construction quality level, respectively.

Likewise, the Connections window shows the results of each connections such as sampled strength and dead load as shown as Fig. 4.11.

The Type Strengths window show distribution of connection strength by connection type as shown as Fig. 4.12.

The Type Damage window shows distribution of speeds at which connection fails by connection type as shown as Fig. 4.13.

Damages tab

The Damages tab shows heatmap by connection type group such as Sheeting, Batten, and Rafter as shown in Fig. 4.14. The heatmap averaged across models is shown by default, and heatmap for individual model can be displayed by moving the slide at the top.

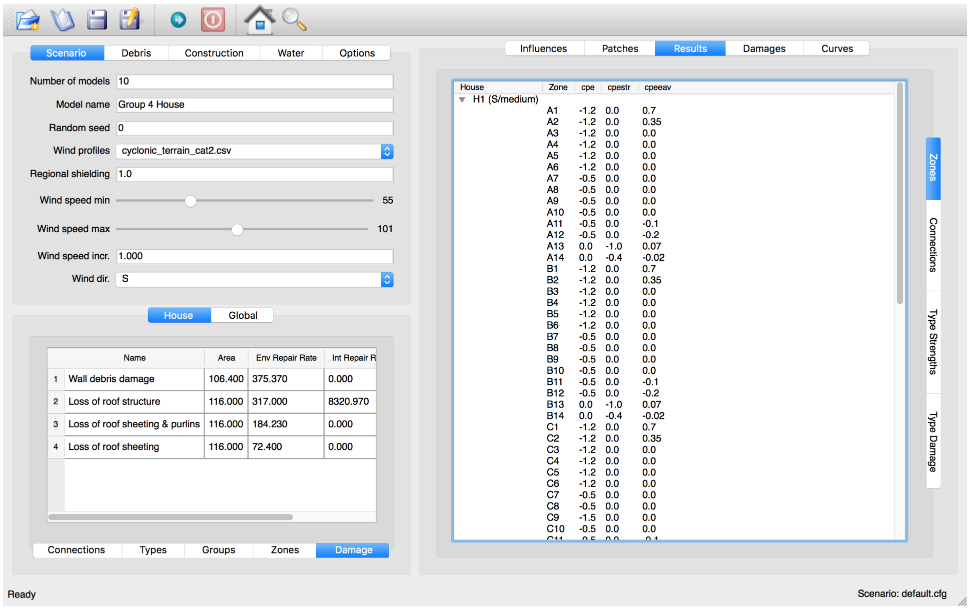

Fig. 4.10: Display of Cpe values for each zone

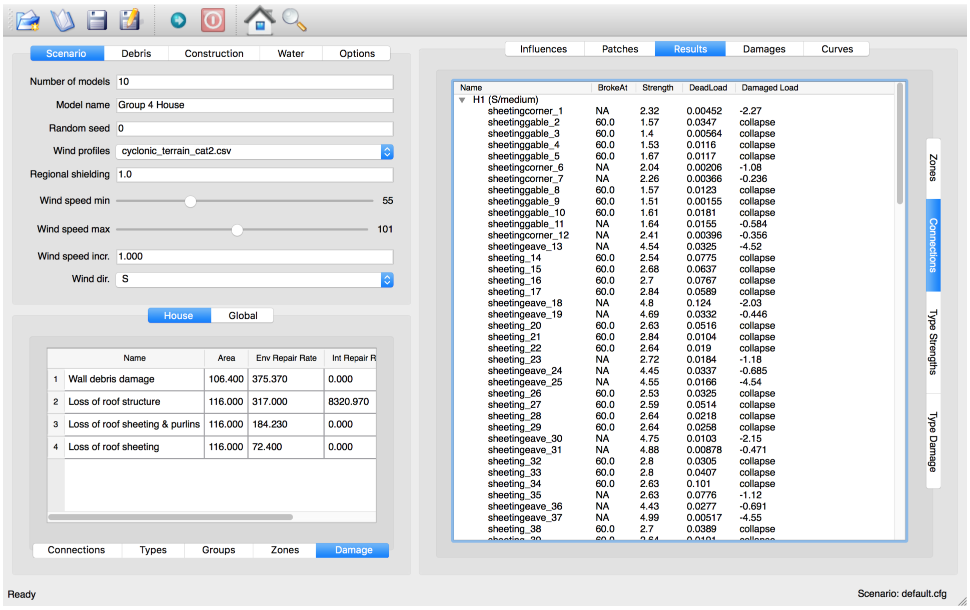

Fig. 4.11: Display of strength, dead load, and failure wind speed for each connection

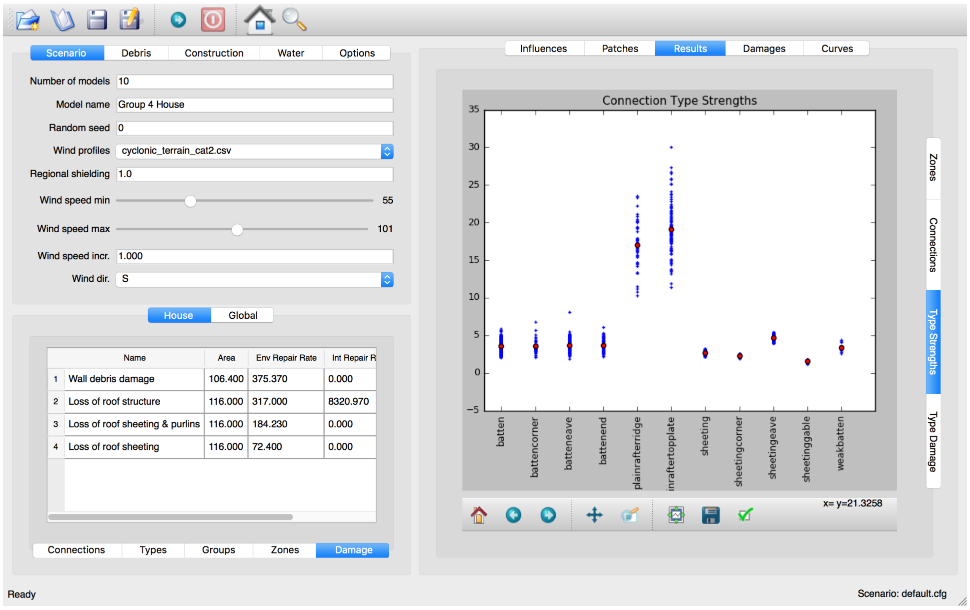

Fig. 4.12: Display of distribution of sampled connection strength by connection type

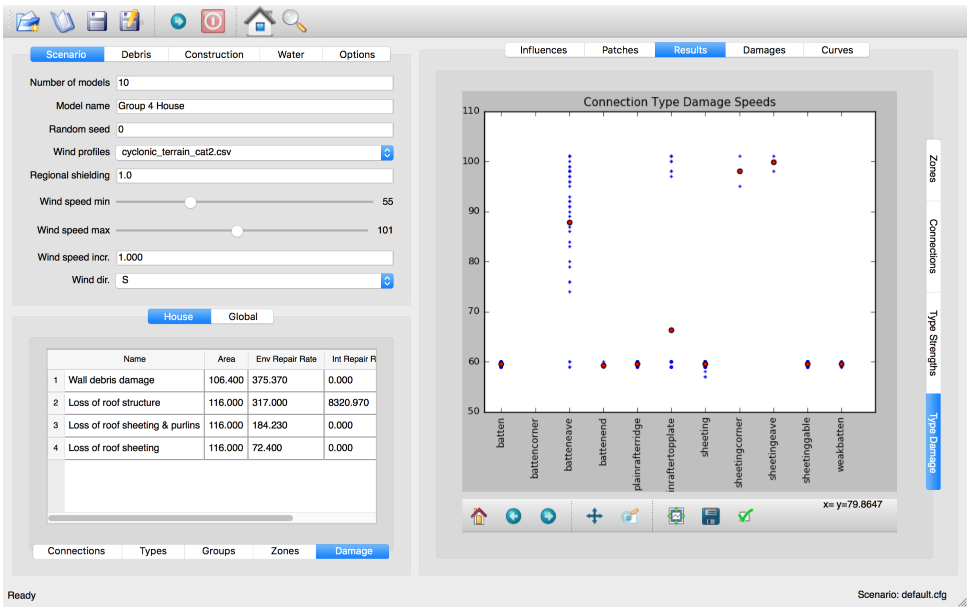

Fig. 4.13: Display of distribution of failure wind speed by connection type

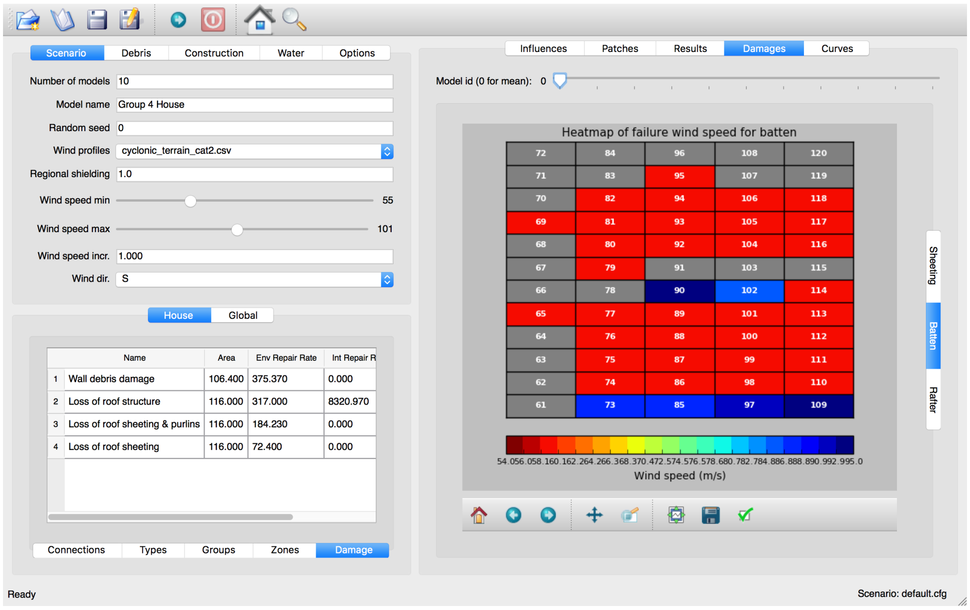

Fig. 4.14: Heatmap of failure wind speed averaged across models for batten group

Curves tab

The Curves tab shows curves in four sub-windows: Vulnerability, Fragility, Water Ingress, and Debris. The Vulnerability window shows a scatter plot of damage indices at each wind speed along with two fitted vulnerability curves, one of which is cumulative lognormal distribution function as (4.1) and the other one is cumulative Weibull distribution as (4.2). The estimated parameter values are displayed at the top.

$$F_X(x; m, \sigma) = \Phi\left(\frac{\ln(x/m)}{\sigma}\right) \tag{4.1}$$

where Φ: the cumulative distribution function of the standard normal distribution, m: median, and σ: logarithmic standard deviation.

$$F(x; \alpha, \beta) = 1 - \exp\left[-\left(\frac{x}{e^{\beta}}\right)^{\frac{1}{\alpha}}\right] \tag{4.2}$$

An example plot is shown in Fig. 4.15.

The Fragility window shows fragility curves for discrete damage states, which are fitted to cumulative lognormal distribution function, as shown in Fig. 4.16.

The Water Ingress window shows a scatter plot of the costing associated with water ingress along wind speed, as shown in Fig. 4.17.

The Debris window shows 1) number of generated debris items, 2) number of impacted debris items, and 3) proportion of models breached by debris along the range of wind speed, as shown in Fig. 4.18.

4.1.5 Bottom

At the bottom of the main window, configuration file name and status of current simulation are displayed as shown in Fig. 4.19.

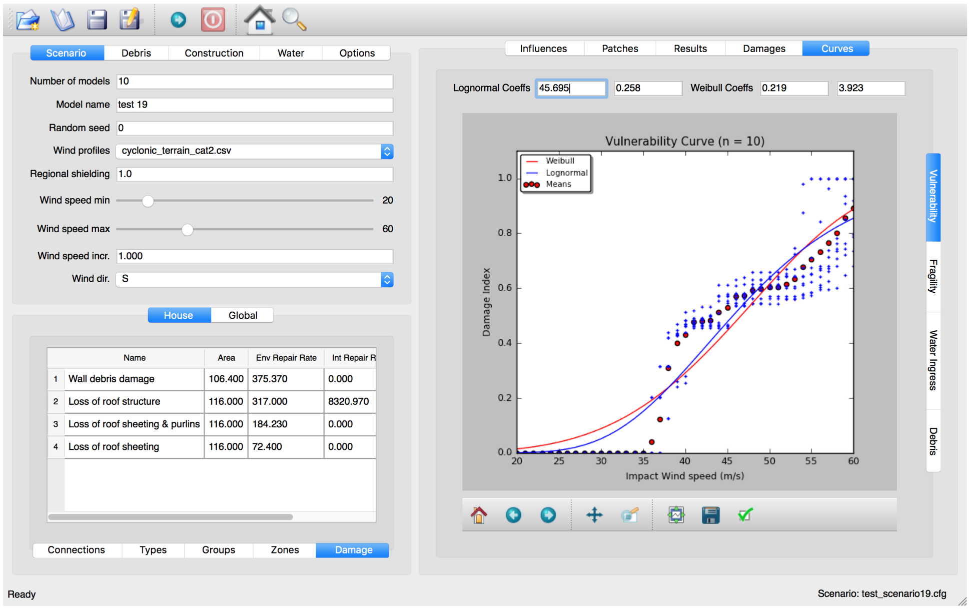

Fig. 4.15: Plot in the Vulnerability window

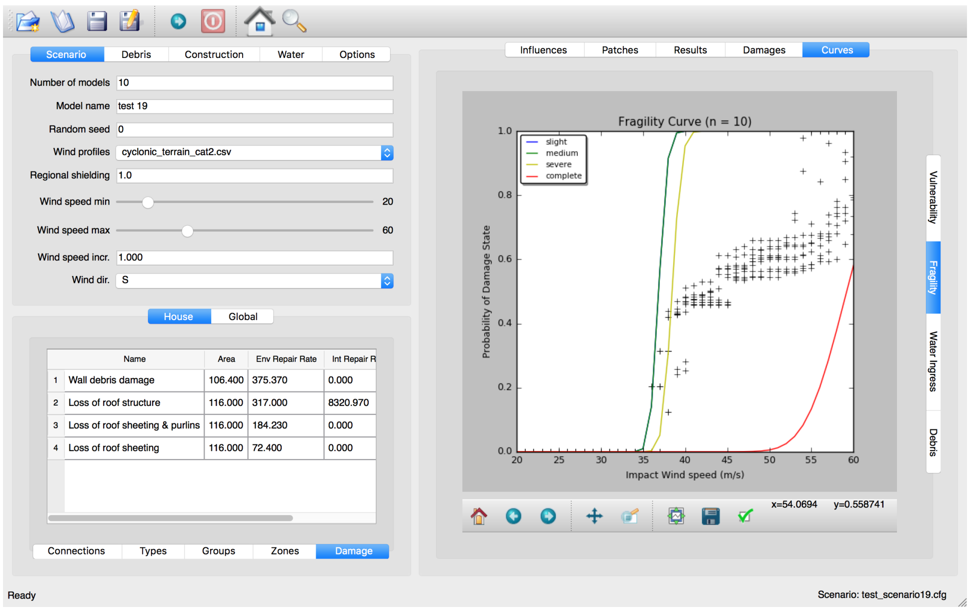

Fig. 4.16: Plot in the Fragility window

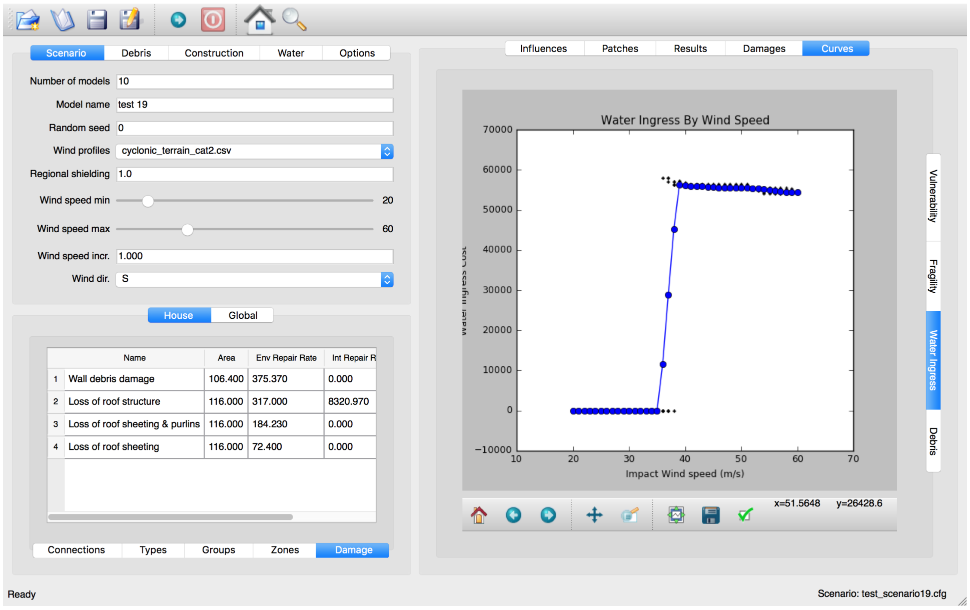

Fig. 4.17: Plot in the Water Ingress window

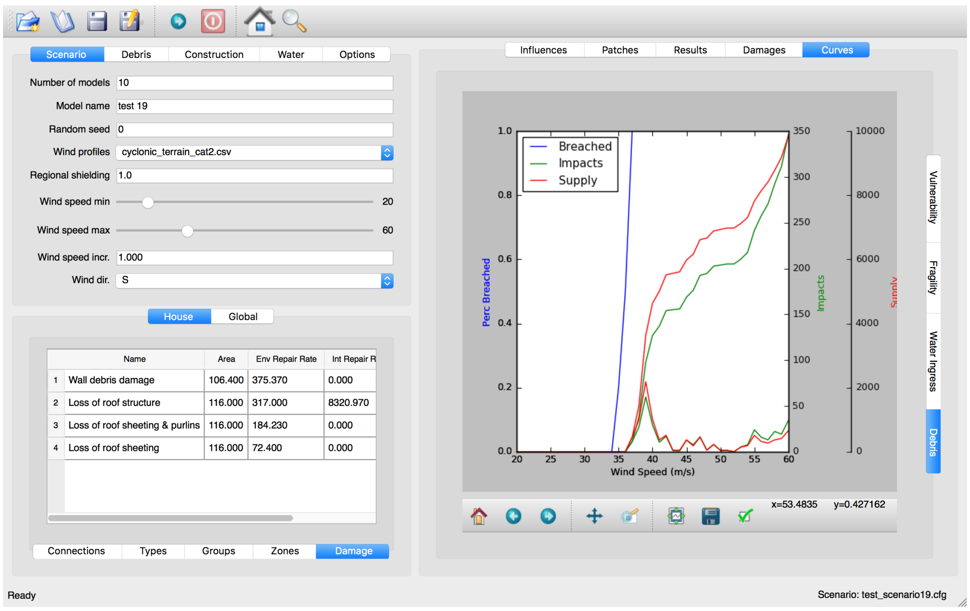

Fig. 4.18: Plot in the Debris window

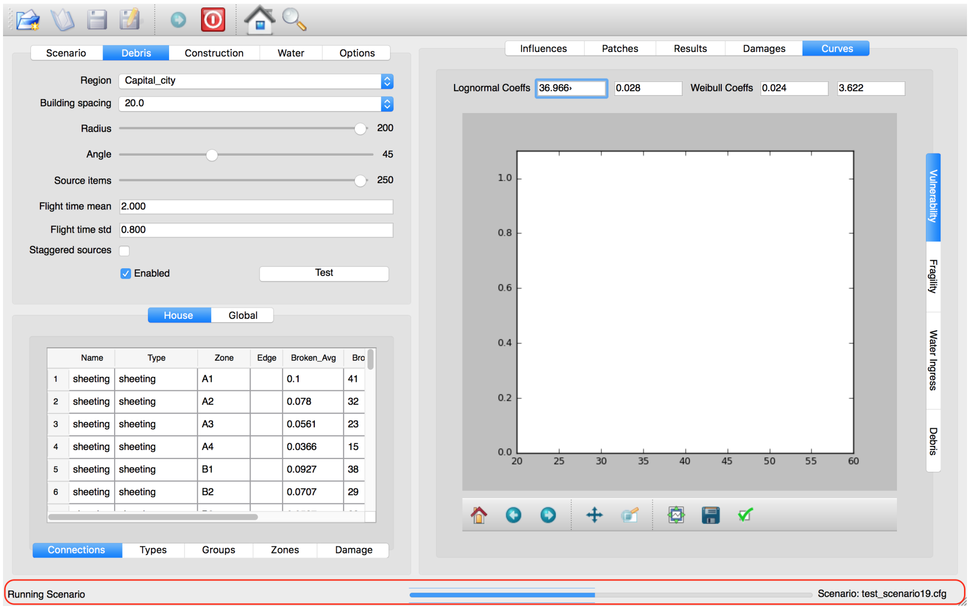

Fig. 4.19: Display of configuration file and status of simulation

4.2 Running simulations

The simulation can be run by either 1) creating a new scenario or 2) loading a saved scenario.

4.2.1 Creating a new scenario

User can create a new scenario by clicking the *New* button, as shown in Table 4.1. The new scenario comes with a set of input files as an template. Once all the setting are set, then user can save the configuration file to a folder where the template input files will be saved. User need to make changes to each of the input files as required.

4.2.2 Loading a saved scenario

User can load a saved scenario file (e.g., default.cfg). A collection of input files should be located in the directory with the folder structure described in Listing 3.1. User may make some changes on the settings through GUI. Once all the settings are set, then simulation can be run by clicking the *Run* button, as shown in Table 4.1. Once the simulation is completed, user can either exit the program or save the current setting to a different scenario, as shown in Fig. 4.20.

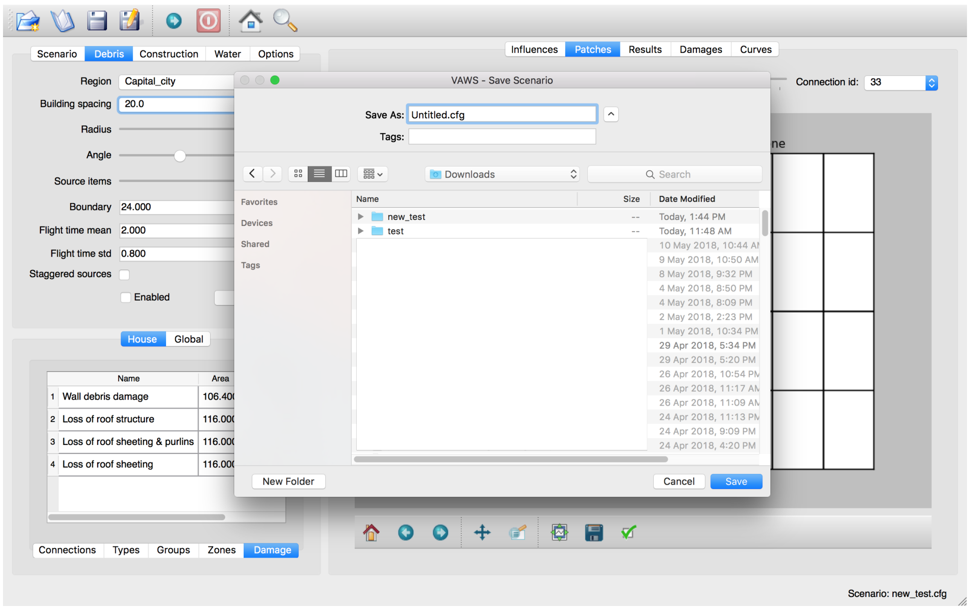

Fig. 4.20: Save as a new scenario

CHAPTER

FIVE

PROGRAM LOGIC

This chapter describes the logic of the program.

5.1 Overall logic

The program is built around the following high level sequence:

1. Create a group of models

- For each model

  - *sample wind direction*
  - *sample construction quality level*
  - *sample wind profile*
  - *set terrain height multiplier*
  - *set shielding multiplier*
  - *set up coverages*
  - *set up zones*
  - *set up connections*

2. Calculate damage indices of the models over a range of wind speeds

- For each wind speed

  - simulate damage for each model

    - *compute free stream wind pressure*
    - *compute zone pressures*
    - *compute coverage load and check damage*
    - *compute connection loads*
    - *check damage of each connection by connection group*
    - *update influence by connection group*
    - *check model collapse*
    - *run debris model and update Cpi*
    - *compute damage index*

- *compute damage index increment*

3. *Fit fragility and vulnerability curves and save outputs*

5.2 Detailed logic

Detailed description of the logic is explained in the following sections by module.

5.2.1 Main module

compute damage index increment (`compute_damage_increment()`)

The increment of mean damage index for the group of models is computed. If the computed increment is less than zero, then zero value is returned.

fit fragility and vulnerability curves and save outputs (`save_results_to_files()`)

Based on the simulation results, fragility (see *fit fragility*) and vulnerability curves (see *fit vulnerability*) are fitted. The output file *results.h5* is also created, and the values of the selected attributes are saved. See *output file* for the list of attributes.

5.2.2 House module

sample wind direction (`House.wind_dir_index`)

The wind direction is set up at the time of model creation, and kept constant during the simulation over a range of wind speeds. If *wind_direction* (Table 3.1) is 'RANDOM', then wind direction is randomly sampled among the eight directions.

sample construction quality level (`House.construction_level`)

A set of mean and coefficient of variation (CV) factors for connection strength is defined for each construction quality level with likelihood as listed in Table 3.4. Construction level for each model is determined from a random sampling, and the corresponding mean and CV factors are later multiplied to arithmetic mean and standard deviation of connection strength as (5.1):

$$
\begin{aligned}
\mu_{adj} &= \mu \times f_{\mu} \\
\sigma_{adj} &= \sigma \times f_{\mu} \times f_{\mathrm{cv}}
\end{aligned}
\tag{5.1}
$$

where μ_{adj} and σ_{adj}: adjusted mean and standard deviation of connection strength reflecting construction quality level, respectively, μ and σ: mean and standard deviation of connection strength, f_{μ} and f_{cv}: mean and CV factors for connection strength.

sample wind profile (`House.profile_index`)

A set of gust envelope wind profiles is read from *wind_profiles* (Table 3.1). Note that each profile is a normalized profile whose value is normalized to 1 at 10 metres height. An example profile is shown in Fig. 3.8. One profile is randomly chosen for each model and kept constant during the simulation over a range of wind speeds.

set terrain height multiplier (`House.terrain_height_multiplier`)

The terrain height multiplier ($M_{z,cat}$) value at the model height is calculated by the interpolation using the selected wind profile over height.

set shielding multiplier (`House.shielding_multiplier`)

The shielding multiplier (M_s) value is determined based on the location. If the value of *regional_shielding_factor* is less or equal to 0.85, which means that the model is located in Australian urban areas, then M_s value is sampled based on the proportion of each type of shielding listed in Table 5.1. Otherwise, M_s value is set to be 1.0, which corresponds to *No shielding*. The proportion of shielding type is adopted following the recommendation in JDH Consulting, 2010 *[3]*.

Table 5.1: Proportion of shielding type

Type	M_s value	Proportion
Full shielding	0.85	63%
Partial shielding	0.95	15%
No shielding	1.0	22%

set action combination factor (`House.combination_factor`)

In the AS/NZS 1170.2 *[1]*, the action combination factor, K_c is defined to reduce wind pressure when wind pressures from more than one building surfaces, for example walls and roof, contribute significantly to a peak load effect. When C_{pi} is between -0.2 and +0.2, then the effect is ignored. Otherwise 0.9 is used.

set up coverages (`House.set_coverages()`)

The coverages make up the wall part of the envelope of the model. Two failure mechanism are implemented: 1) failure by wind load and 2) failure by windborne debris.

A set of coverage components (`Coverage`) is defined using the information provided in the input files of *coverages.csv*, *coverage_types.csv* and *coverages_cpe.csv*. The C_{pe} and strength values for each coverage component are sampled when it is defined. The windward direction for each coverage component is assigned from among *windward*, *leeward*, *side1*, or *side2*, which is used in determining the windward direction of dominant opening due to coverage failure.

set up zones (`House.set_zones()`)

A set of zone components (`Zone`) is defined using the information provided in the input files of *zones.csv*, *zones_cpe_mean.csv*, *zones_cpe_str_mean.csv*, *zones_cpe_eave_mean.csv*, and *zones_edges.csv*. The C_{pe} value for each zone component is sampled when it is defined.

set up connections (`House.set_connections()`)

A set of connection components (`Connection`) is defined using the information provided in the input files of *conn_groups.csv*, *conn_types.csv*, *connections.csv*, *influences.csv*, and *influence_patches.csv*. The strength and dead load values for each connection component are sampled and influence and influence patch for each connection are also defined with reference to either zone or another connection components.

A set of connection type group (`ConnectionTypeGroup`) is also defined, and reference is created to relate a connection component to a connection type group. A connection type group is further divided into sub-group by section in order to represent load distribution area within the same group. For instance roof sheetings on a hip roof are divided into a number of sheeting sub-groups to represent areas divided by roof ridge lines.

set footprint for debris impact (`House.footprint`)

Once the wind direction of the model is determined, the footprint for debris impact is created by rotating the model footprint with regard to the wind direction as set out in Table 5.2 (`House.footprint`). The boundary for debris impact assessment is also defined with the radius of boundary (`Config.impact_boundary`). Note that all the debris sources are assumed to be located in the East of the model when debris impact to the model is simulated.

Table 5.2: Rotation angle by wind direction

Wind direction	Rotation angle (deg)
S or N	90
SW or NE	45
E or W	0
SE or NW	-45

set up coverages for debris impact (`House.debris_coverages`)

Once the wind direction of the model is determined, the coverages for debris impact are selected based on the wind direction.

calculate free stream wind pressure (`House.compute_qz()`)

The free stream wind pressure, q_z is calculated as (5.2):

$$q_z = \frac{1}{2} \times \rho_{air} \times (V \times M_{z,cat} \times M_s)^2 \times 1.0\text{e-}3 \tag{5.2}$$

where ρ_{air}: air density (=1.2 $\mathrm{kg/m^3}$), V: 3-sec gust wind speed at 10m height, $M_{z,cat}$: terrain-height multiplier, M_s: shielding multiplier. Note that 1.0e-3 is multiplied to convert the unit of the wind pressure from Pa to kPa.

check model collapse (`House.check_collapse()`)

The model is deemed to be collapsed if the proportion of damaged components out of the total components is greater than the value of *trigger_collapse_at*, which is listed in Table 3.11, for any group with non-zero value of *trigger_collapse_at*.

run debris model and update C_{pi} (`House.run_debris_and_update_cpi()`)

If the value of *debris* is *True* (see Table 3.2), then debris impact to the model is simulated. See *Debris module* for more details.

The internal pressure coefficient, C_{pi} is determined based on Table 5.3 and Table 5.4 depending on the existence of dominant opening by either coverage failure or debris breach, which are revised from Tables 5.1(A) and 5.1(B) of AS/NZS 1170.2 *[1]*, respectively.

Table 5.3: C_{pi} for buildings without dominant openings

Condition	C_{pi}
All walls equally breached	-0.3
Two or three windward walls equally breached	0.2
Two or three non-windward walls equally breached	-0.3

Table 5.4: C_{pi} for buildings with dominant openings

Ratio of dominant opening to total open area (r)	Dominant opening on windward wall	Dominant opening on leeward wall	Dominant opening on side wall
$r < 0.5$	-0.3	-0.3	-0.3
$0.5 \leq r < 1.5$	0.2	-0.3	-0.3
$1.5 \leq r < 2.5$	0.7 C_{pe}	C_{pe}	C_{pe}
$2.5 \leq r < 6.0$	0.85 C_{pe}	C_{pe}	C_{pe}
$r \geq 6.0$	C_{pe}	C_{pe}	C_{pe}

compute damage index (`House.compute_damage_index()`)

The damage index is calculated over the following steps:

1. calculate sum of damaged area by connection group (`House.compute_area_by_group()`)
2. Apply damage factoring (`House.apply_damage_factoring()`)

In order to avoid double counting of repair cost, damage cost associated with child group(s) will be factored out of damage cost of the parent group as explained in *3.4.16*.

3. Calculate sum of damaged area by damage scenario (`House.compute_area_by_scenario()`)

A damage scenario is assigned to each connection type group as explained in *3.4.2*.

4. calculate total damage cost and damage index prior to water ingress (DI_p) as (5.3):

$$DI_p = \frac{\sum_{i=1}^{S} C_i}{R} \tag{5.3}$$

where S: number of damage scenario, C_i: damage cost for i th damage scenario, and R: total replacement cost.

5. Calculate cost by water ingress damage, C_{wi} if required as explained in *damage due to water ingress*.

6. calculate damage index as (5.4):

$$DI = \frac{\sum_{i=1}^{S} C_i + C_{wi}}{R} \tag{5.4}$$

5.2.3 Zone module (`Zone`)

sample Cpe (`Zone.cpe`)

The external pressure coefficient, C_{pe} is used in computing zone pressures, and is sampled from Type III extreme value distribution (`stats.sample_gev()`) which has the cumulative distribution function and probability density as (5.5) and (5.6), respectively.

$$F(s; k) = \exp(-(1 - ks)^{1/k}) \tag{5.5}$$

$$f(s; a, k) = \frac{1}{a}(1 - ks)^{1/k-1} \exp(-(1 - ks)^{1/k}) \tag{5.6}$$

where $s = (x - u)/a$, u: location factor ($\in \mathbb{R}$), a: scale factor (> 0), and k: shape factor ($k \neq 0$).

The mean and standard deviation are calculated as (5.7):

$$\begin{aligned} \mathrm{E}(X) &= u + \frac{a}{k}\left[1 - \Gamma(1 + k)\right] \\ \mathrm{SD}(X) &= \frac{a}{k}\sqrt{\Gamma(1 + 2k) - \Gamma^2(1 + k)} \end{aligned} \tag{5.7}$$

The u and a can be estimated given $c_v \left(= \frac{SD}{E}\right)$ and k values as (5.8):

$$\begin{aligned} a &= \mathrm{E}\,\frac{c_v}{B} \\ u &= \mathrm{E} - a \times A \end{aligned} \tag{5.8}$$

where $A = (1/k)\left[1 - \Gamma(1 + k)\right]$ and $B = (1/k)\sqrt{\Gamma(1 + 2k) - \Gamma^2(1 + k)}$.

calculate zone pressure (`Zone.calc_zone_pressure()`)

Two kinds of zone pressure, p_z for zone component related to sheeting and batten and $p_{z,str}$ for zone component related to rafter, are computed as (5.9):

$$\begin{aligned} p_z &= q_z \times (C_{pe} - C_{pi,\alpha} \times C_{pi}) \times D_s \times K_c \\ p_{z,str} &= q_z \times (C_{pe,str} - C_{pi,\alpha} \times C_{pi} - C_{pe,eave}) \times D_s \times K_c \end{aligned} \tag{5.9}$$

where q_z: free stream wind pressure, C_{pe}: external pressure coefficient, C_{pi}: internal pressure coefficient, $C_{pi,\alpha}$: proportion of the zone's area to which internal pressure is applied, $C_{pe,str}$: external pressure coefficient for zone component related to rafter, $C_{pe,eave}$: external pressure coefficient for zone component related to eave, D_s: differential shielding, and K_c: action combination factor. The value of differential shielding is determined as explained in *set differential shielding*. The value of action combination factor is determined as explained in *set action combination factor*.

set differential shielding (`Zone.differential_shielding`)

If the value of *differential_shielding* (see Table 3.2) is *True*, then differential shielding effect is considered in calculating zone pressure. Based on the recommendations from JDH Consulting, 2010 *[3]*, adjustment for shielding multiplier is made as follows:

- **For outer suburban situations and country towns (*building_spacing*=40m),** adjust M_s to 1.0 except for the leading edges of upwind roofs
- **For inner suburban buildings (*building_spacing* =20m) with full shielding (M_s=0.85),** adjust M_s to 0.7 for the leading edges of upwind roofs
- **For inner suburban buildings (*building_spacing* =20m) with partial shielding (M_s=0.95),** adjust M_s to 0.8 for the leading edges of upwind roofs
- Otherwise, no adjustment is made.

5.2.4 Coverage module (`Coverage`)

calculate coverage load and check damage (`Coverage.check_damage()`)

The load applied for each of coverages are calculated as (5.10):

$$L = q_z \times (C_{pe} - C_{pi}) \times A \times K_c \tag{5.10}$$

where q_z: free stream wind pressure, C_{pe}: external pressure coefficient, C_{pi}: internal pressure coefficient, A: area, and K_c: action combination factor.

If the calculated load exceeds either positive or negative strength, which represents strength in either direction, then it is deemed to be damaged.

5.2.5 Connection module (`Connection` and `ConnectionTypeGroup`)

calculate connection load (`Connection.check_damage()`)

The load applied for each of connections are calculated as (5.11):

$$L_i = D_i + \sum_{j=1}^{N_z} (I_{ji} \times A_j \times P_j) + \sum_{j=1}^{N_c} (I_{ji} \times L_j) \tag{5.11}$$

where L_i: applied load for i th connection, D_i: dead load of i th connection, N_z: number of zones associated with the i th connection, N_c: number of connections associated with the i th connection, A_j: area of j th zone, P_j: wind pressure on j th zone, I_{ji}: influence coefficient from j th either zone or connection to i th connection.

If the load applied for a connection is less than the negative value of its strength, then the connection is considered damaged.

check connection damage by connection type group (`ConnectionTypeGroup.check_damage()`)

Damage of each connection is checked by connection type group. If the load applied for a connection is less than the negative value of its strength, then the connection is considered damaged. Then damage grid of the connection type group (`ConnectionTypeGroup.damage_grid`) is updated with the index of the damaged connection, which is later used in updating influence of intact components (`ConnectionTypeGroup.update_influence()`).

update influence by connection group (`ConnectionTypeGroup.update_influence()`)

The influence coefficient is used to associate one connection with another either zone or connection with regard to load distribution. For instance, if connection 1 has influences of connection 2 and 3 with coefficient 0.5 and 0.5, respectively, then the load on connection 1 is equal to the sum of 0.5 times load on connection 2 and 0.5 times load on connection 3, as shown in (5.11).

Once a connection is damaged, then load on the damaged connection needs to be distributed to other intact connections accordingly, which means that influence set of the connections needs to be updated.

Two types of influence update are implemented:

1. update influence coefficients of the next intact connections for the distribution of load on the damaged connection, when *dist_dir* is either *col* or *row* (`ConnectionTypeGroup.update_influence()`)

Given the damage of connection of either sheeting and batten connection type group, the influence coefficient will be distributed evenly to the next intact connections of the same type to the distribution direction (*dist_dir* listed in Table 3.11). If both the next connections, which are left and right if *dist_dir* is 'row' or above and below if 'col', of the damaged connection are intact, then the half of the load is distributed to the each of next intact connection. Otherwise, the full load of the damaged connection is distributed to the intact connection.

2. replace the existing influence set with new one, when *dist_dir* is *patch* (`ConnectionTypeGroup.update_influence_by_patch()`)

Unlike sheeting and batten, a connection of rafter group fails, then influence set of each connection associated with the failed connection are replaced with a new set of influence, which is termed "patch". In the current implementation, the patch is defined for a single failed connection. Thus the failure order of the connections may make difference in the resulting influences as shown in Table 5.5.

Table 5.5: Example of how patch works

Failed connection	Connection	Patch (connection: influence coeff.)
1	3	1:0.0, 2:0.5, 3:0.5
2	3	1:0.5, 2:0.0, 3:1.0
1 and then 2	3	1:0.0, 2:0.0, 3:1.0
2 and then 1	3	1:0.0, 2:0.0, 3:0.5

5.2.6 Debris module (`Debris`)

The methdology of modelling damage from wind-borne debris implemented in the code is described in Holmes et al., 2010 *[4]* and Wehner et al., 2010 *[6]*. The debris damage module consists of four parts: 1) debris source generation, 2) debris generation, 3) debris trajectory, and 4) debris impact.

debris source generation

The debris sources are generated by calling `create_sources()`, which requires a number of parameters as shown in the Fig. 5.1.

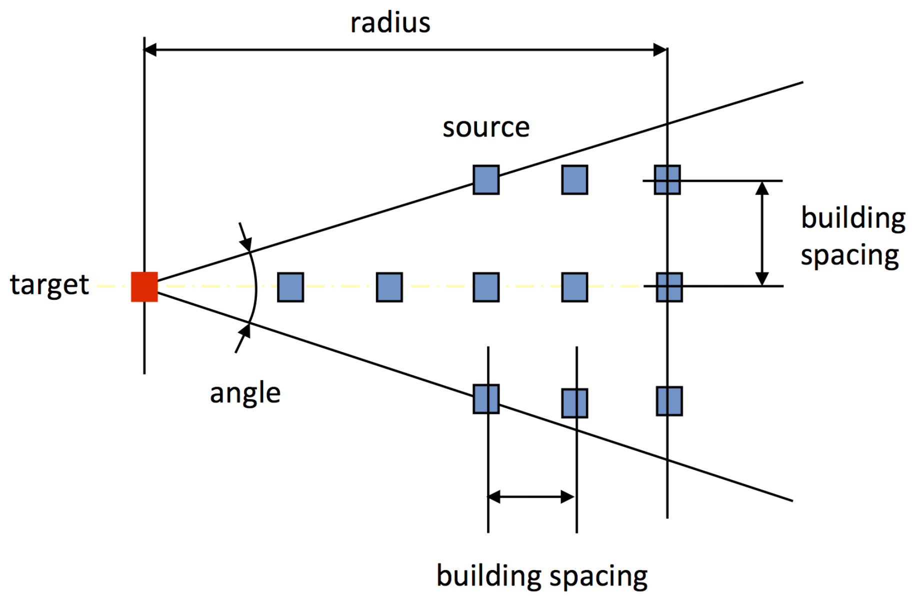

Fig. 5.1: Distribution of debris sources with parameters

Depending on the value of *staggered_sources*, different layout of debris sources can be generated as shown in Fig. 5.2 and Fig. 5.3.

debris generation

For each wind speed, a group of debris items are generated by calling `generate_debris_items()`. The mean number of debris items to be generated (N_{mean}) is calculated by (5.12).

$$N_{mean} = \mathrm{nint}\left(\Delta DI \times N_{items}\right) \tag{5.12}$$

where N_{items}: number of debris items per source defined in *3.1.3*, ΔDI: increment in damage index from previous wind step, and nint: nearest integer function.

The number of generated debris items is assumed to follow the Poisson distribution with parameter $\lambda = N_{mean}$. For each debris source, the number of generated debris items is randomly sampled from the distribution, and debris type is randomly chosen as many as number of items with probability proportional to the ratio of each type defined in Table 3.9. The debris types are provided in the section of *3.2 debris.csv*.

debris trajectory

For each generated debris item, mass (`Debris.mass`), frontal area ((`Debris.frontal_area`), and flight time (`Debris.flight_time`) are sampled from the lognormal distribution with parameter values provided in *3.1.3* and *3.2*. The flight distance (`Debris.flight_distance`) is calculated based on the methodology presented in the Appendix of Lin and Vanmarcke, 2008 *[5]*. Note that the original fifth polynomial functions are replaced with quadratic one with the coefficients as listed in Table 5.6. The computed flight distance by debris type using the fifth and quadratic polynomials is shown in Fig. 5.4.

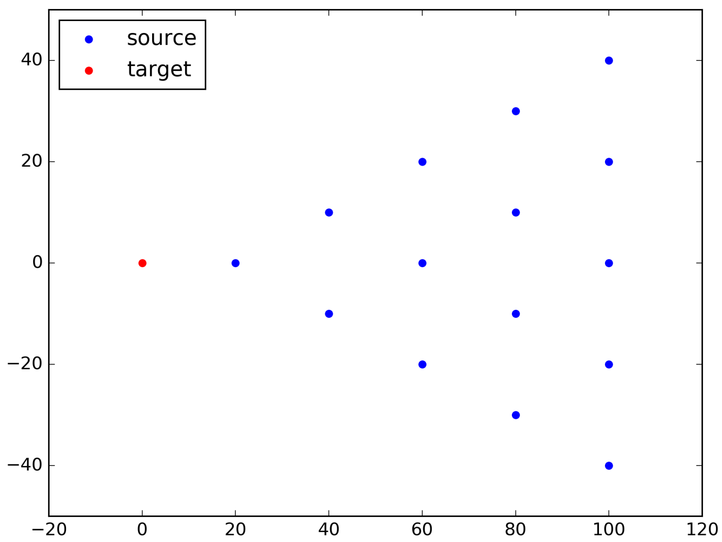

Fig. 5.2: Distribution of debris sources generated with debris_radius = 100.0 (m), debris_angle = 45.0 (deg), debris_space = 20.0 (m), and staggered_sources = *True*.

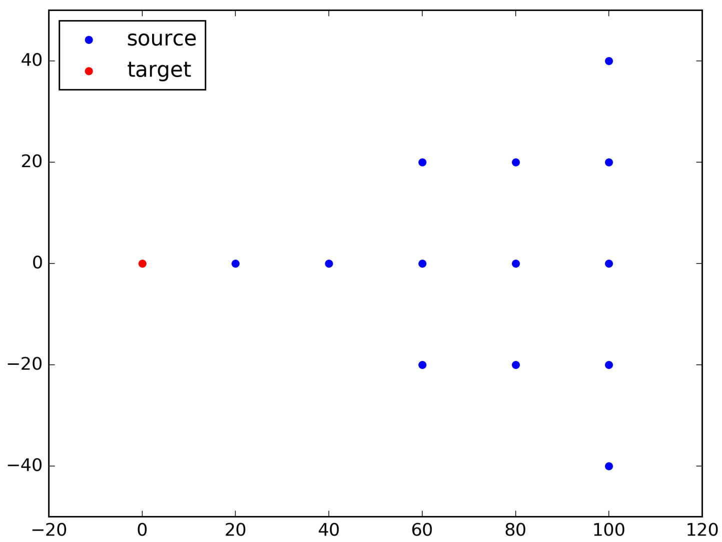

Fig. 5.3: Distribution of debris sources generated with debris_radius = 100.0 (m), debris_angle = 45.0 (deg), debris_space = 20.0 (m), and staggered_sources = *False*.

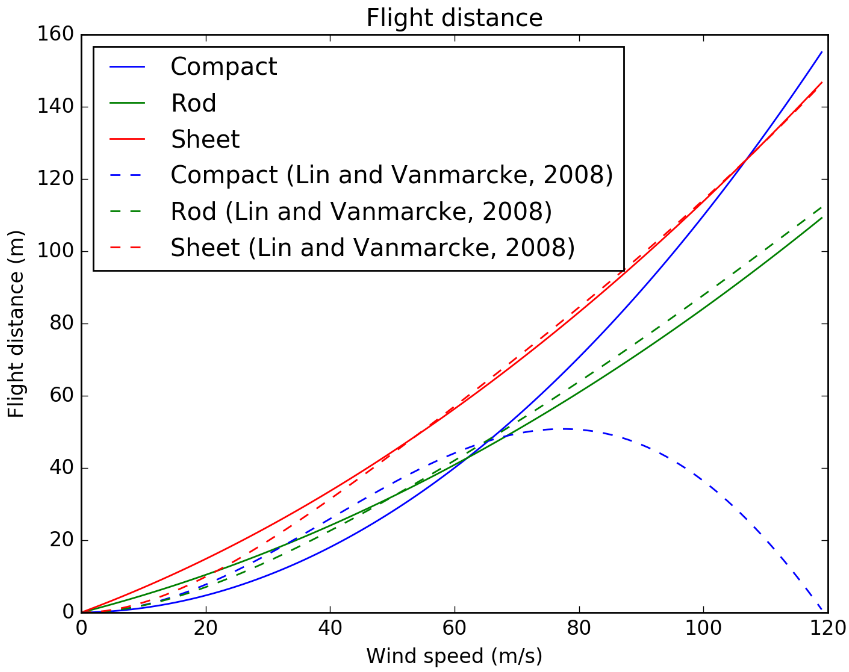

Fig. 5.4: Flight distance of debris item

Table 5.6: Coefficients of quadratic function for flight distance computation by debris type

Debris type	Linear coeff.	Quadratic coeff.
Compact	0.011	0.2060
Rod	0.2376	0.0723
Sheet	0.3456	0.072

The probability distribution of point of landing of the debris in a horizontal plane is assumed to follow a bivariate normal distribution as (5.13).

$$f_{x,y} = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{(x-d)^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right] \tag{5.13}$$

where x and y are the coordinates of the landing position of the debris, σ_x and σ_y: standard deviation for the coordinates of the landing position, and d: expected flight distance. The value of σ_x and σ_y are set to be $d/3$ and $d/12$, respectively.

Following Lin and Vanmarcke 2008, the ratio of horizontal velocity of the windborne debris object to the wind gust velocity is modelled as a random variable with a Beta distribution as (5.14).

$$\frac{u_m}{V_s} \sim Beta(\alpha, \beta) \tag{5.14}$$

where u_m: the horizontal velocity of the debris object, V_s: the local (gust) wind speed, α and β are two parameters of the Beta distribution and estimated as (5.15).

$$\begin{aligned}\alpha &= E \times \nu \\ \beta &= \nu \times (1 - E)\end{aligned} \tag{5.15}$$

where E: the expected value and $\nu = \alpha + \beta$.

The expected value (E) and the parameter (ν) are assumed to be as (5.16).

$$E = 1 - \exp\left(-b\sqrt{x}\right)$$
$$\nu = \max\left[\frac{1}{E}, \frac{1}{1-E}\right] + 3.0 \tag{5.16}$$

where x: the flight distance, b: a dimensional parameter calucalted as (5.17). If E is 1, then α and β are assigned with 3.996 and 0.004, respectively.

$$b = \sqrt{\frac{\rho_a C_{D,av} A}{m}} \tag{5.17}$$

where ρ_a: the air density, $C_{D,av}$: an average drag coefficient, A: the frontal area, and m: the mass of the object.

The momentum ξ (`Debris.momentum`) is calculated using the sampled value of the ratio, $\frac{u_m}{V_s}$ as (5.18).

$$\xi = \left(\frac{u_m}{V_s}\right) \times m \times V_s \tag{5.18}$$

debris impact

Either if the landing point is within the footprint of the model or if the line linking the source to the landing point intersects with the footprint of the model and the landing point is within the boundary, then it is assumed that an impact has occurred. The criteria of debris impact is illustrated in the Fig. 5.5 where blue line represents debris trajectory with impact while red line represents one without impact.

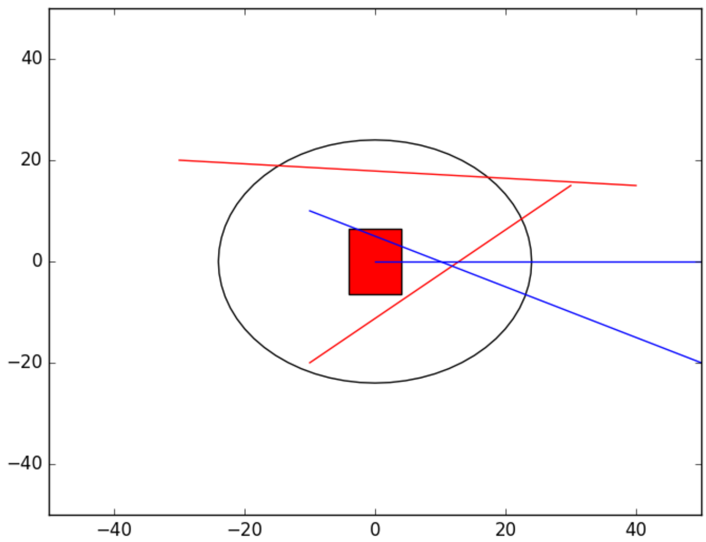

Fig. 5.5: Graphical presentation of debris impact criteria

Based on the methodology presented in HAZUS and Lin and Vanmacke (2008), the number of impact N is assumed to follow a Poisson distribution as (5.19).

$$N \sim \text{Pois}(\lambda)$$
$$\lambda = N_v \cdot q \cdot F_{\xi}(\xi > \xi_d) \tag{5.19}$$

where N_v: number of impacts at a single wind speed, q: proportion of coverage area out of the total area of envelope, F_ξ: the cumulative distribution of momentum, and ξ_d: threshold of momentum or energy for damage of the material of the coverage.

The probability of damage can be calculated based on the Poisson distribution as (5.20).

$$P_D = 1 - P(N = 0) = 1 - \exp\left[-\lambda\right] \tag{5.20}$$

q and $F_\xi(\xi > \xi_d)$ are estimated for each coverage.

If the material of the coverage is glass, then P_D is computed and compared against a random value sampled from unit uniform distribution to determine whether the coverage is damaged or not. If the coverage is damaged, then damaged area is set to be equal to the coverage area. For coverage with non-glass material, a random value of number of impact is sampled from the Poisson distribution with λ, and damaged coverage area is then computed assuming that the area requiring repairs due to debris impact is 1.

Since version 2.2 of the code, a Monte Carlo based approach is implemented (`Debris.check_coverages()`). For each debris item, a coverage component is chosen based on the ratio of the area out of the total area of the coverages, once debris impact is assumed to occur. If the computed momentum exceeds the momentum capacity of the coverage, then damaged coverage area is computed depending on the material of the coverage as explained above.

5.2.7 damage_costing module (`Costing`)

damage due to water ingress

The damage cost induced by water ingress is estimated over the following three steps:

1. estimate amount of water ingress (`compute_water_ingress_given_damage()`)

The amount of water ingress is estimated based on the relationship between water ingress and wind speed, which is listed in Table 3.5. The estimated damage index prior to water ingress is used to choose the right curve as shown in Fig. 3.4.

2. determine damage scenario (`House.determine_scenario_for_water_ingress_costing()`)

The damage scenario for water ingress is determined based on the order of damage scenario as listed in Table 3.20. One damage scenario is selected by the order among the damage scenarios with which damage area of connection associated is greater than zero. When the damage index is zero (or no connection damage yet), then damage scenario of 'WI only' is used.

3. calculate cost for water ingress damage (`House.compute_water_ingress_cost()`)

The cost for water ingress damage is estimated using the data provided in *3.4.17*. The example plot for the scenario of *loss of roof sheeting* is shown in Fig. 5.6. The cost for water ingress damage is estimated using the curve for water ingress closest to the estimated amount of water ingress.

5.2.8 Curve module

fit fragility (`curve.fit_fragility_curves()`)

The probability of exceeding a damage state ds at a wind speed x is calculated as (5.21):

$$P\left(DS \geq ds|x\right) = \frac{\sum_{i=1}^{N}\left[DI_{i|x} \geq t_{ds}\right]}{N} \tag{5.21}$$

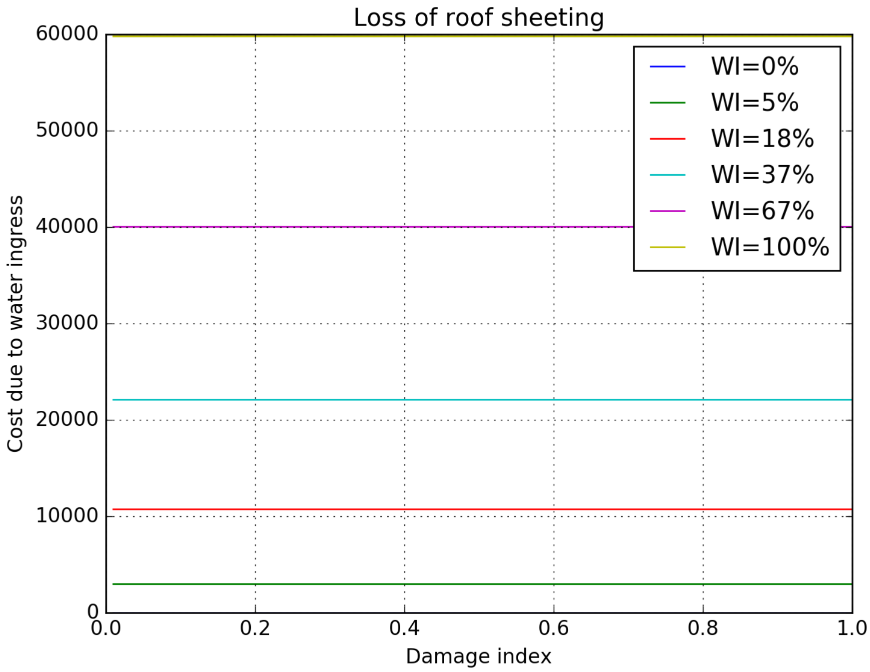

Fig. 5.6: Relationship between cost due to water ingress damage and damage index

where N: number of models, $DI_{i|x}$: damage index of i th model at the wind speed x, and t_{ds}: threshold for damage state ds.

Then for each damage state, a curve of cumulative lognormal distribution (4.1) is fitted to the computed probabilities of exceeding the damage state.

fit vulnerability (`curve.fit_vulnerability_curve()`)

Two types of curves are used to fit the results of damage indices of models: a cumulative lognormal distribution ((4.1), `curve.vulnerability_lognormal()`) and Weibull distribution ((4.2), `curve.vulnerability_weibull()`).

CHAPTER

SIX

GLOSSARY

connection models a physical structural connection that will bear a simulated load and when broken cause load to be distributed and generate damage outcomes.

connection type a collection of connections sharing the same strength and dead load statistical characteristics and costing area

connection group, connection type group a collection of connection types sharing the same load distribution and costing scenario

coverage a component making up the wall part of the envelope of the model

C_{pe} external pressure coefficient

C_{pi} internal pressure coefficient

$C_{pi,\alpha}$ proportion of the zone's area to which internal pressure is applied

CV coefficient of variation, the ratio of the standard deviation to the mean

damage index The total cost of repairing the building fabric of a group of buildings exposed to severe natural hazard divided by the total cost of fully rebuilding the same assets in the existing locality to current local building regulations.

differential shielding incremental adjustments to be applied to envelope surface pressures to account for different degrees of shielding between envelope surfaces on a single shielded structure.

fragility, fragility function, fragility curve Fragility describes the probability of discrete damage states for a specific hazard. Fragility function or curve is referred to a damage model which describes the likelihood of a building of a particular type being damaged to a defined degree for a given level of natural hazard exposure.

influence coefficient coefficient relating a connection to either zone or connection with regard to load distribution

K_c action combination factor. This factor is devised to reduce wind pressure when wind pressures from more than one building surfaces, for example walls and roof, contribute significantly to a peak load effect.

M_s shielding multiplier. This multiplier represents the reduction in peak 3-second gust velocity at a given height and terrain, caused by the presence of buildings and other obstructions upwind of the site of interest.

$M_{z,cat}$ terrain height multiplier

patch a set of revised influence coefficients for a connection

q_z free stream wind pressure

vulnerability, vulnerability function, vulnerability curve A damage model, or curve, which describes the average severity of physical economic loss to a group of buildings of a particular type in terms of a damage index with increasing natural hazard exposure.

zone an area of building envelope on which wind pressure acts. Zone is linked with connection with influence coefficient so the wind pressure is transformed to wind load on a connection.

BIBLIOGRAPHY

[1] AS/NZS 1170.2:2011. Structural design actions part 2: wind actions. Technical Report, Standards Australia, 2011.

[2] Geoscience Australia. Wind vulnerability model development for adaptation studies on specific residential and industrial buildings. Technical Report, Report to the Department of Climate Change and Energy Efficiency, 2011.

[3] JDH Consulting. Contributions to development of a wind vulnerability model - final report 2009-2010. Technical Report, Report to Geoscience Australia, 2010.

[4] J.D. Holmes, M. Wehner, C. Sandland, and M. Edwards. Modelling damage to residential buildings from wind-borne debris – part 1. methodology. In *Proceedings of 14th Australasian Wind Engineering Society Workshop*. 2010.

[5] Ning Lin and Erik Vanmarcke. Windborne debris risk assessment. *Probabilistic Engineering Mechanics*, 23(4):523–530, October 2008.

[6] M. Wehner, C. Sandland, J. Holmes, D. Henderson, and M. Edwards. Modelling damage to residential buildings from wind-borne debris – part 2. implementation. In *Proceedings of 14th Australasian Wind Engineering Society Workshop*. 2010.

[7] Martin Wehner, John Ginger, John Holmes, Carl Sandland, and Mark Edwards. Development of methods for assessing the vulnerability of australian residential building stock to severe wind. *IOP Conference Series: Earth and Environmental Science*, 11(1):12–17, 2010. URL: http://stacks.iop.org/1755-1315/11/i=1/a=012017.