VAWS user manual

Release 2.0

Geoscience Australia

Jan 16, 2018

CONTENTS

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. 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, 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.

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 of 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 zone and coverage

  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

  Multiple construction levels can be defined with mean and cov factors which will be used to adjust the mean and cov 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 each 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 2.7 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 2.7. This step can be skipped if either Miniconda or Anaconda with Python 2.7 is already installed.

- Windows
  - Double-click the downloaded *Miniconda2-latest-Windows-x86_64.exe* file.
  - When installation is finished, from the *Start* menu, open the *Anaconda Prompt*.
- Linux
  - In Terminal window, run

    ```
    $ bash Miniconda2-latest-Linux-x86_64.sh
    ```
- Mac
  - In Terminal window, run

    ```
    $ bash Miniconda2-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 python=2.7
```

3. Activate the environment.

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

- Windows

```
activate vaws_env
```

- Linux/Mac

```
source activate vaws_env
```

4. Install the code from conda channel

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

```
conda install -c crankymax vaws
```

In case you see *PackageNotFoundError: Packages missing in current channels:* then enter the following in the terminal client and try above command again.

```
conda config --add channels conda-forge
```

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 in the terminal to remove the old version and re-install the new version of the code.

```
conda remove vaws
conda install -c crankymax vaws
```

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.

2.2 Instructions for developers

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

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.

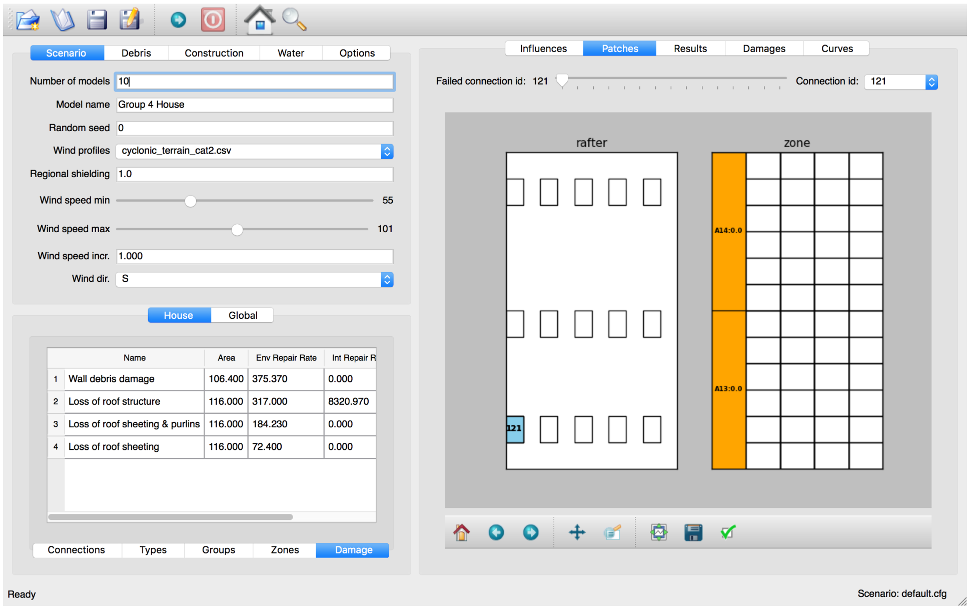

Fig. 2.1: Program main window with default scenario loaded

- 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 2.7 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_win.yml
```

- Linux/Mac

```
cd <vaws dir>
conda env create --name vaws_env --file vaws_env.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.gui.main -c ./vaws/scenarios/default/default.cfg  # for␣
↪default scenario
python -m vaws.gui.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 *anaaconda-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
# $ 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.

```
nosetests -v vaws
```

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_is_wall (vaws.model.tests.test_zone.MyTestCase) ... ok
test_str2num (vaws.model.tests.test_zone.MyTestCase) ... ok

----------------------------------------------------------------------
Ran 93 tests in 56.053s

OK
```

CHAPTER

THREE

INPUT DATA

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).

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
```

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
house_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
diff_shielding = False
water_ingress = True
construction_levels = True
save_heatmaps = True

[debris]
region_name = Capital_city
staggered_sources = False
source_items = 250
building_spacing = 20.0
debris_radius = 200
debris_angle = 45
flight_time_mean = 2.0
flight_time_stddev = 0.8

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

[water_ingress]
thresholds = 0.1, 0.2, 0.5
speed_at_zero_wi = 50.0, 35.0, 0.0, -20.0
speed_at_full_wi = 75.0, 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
house_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)

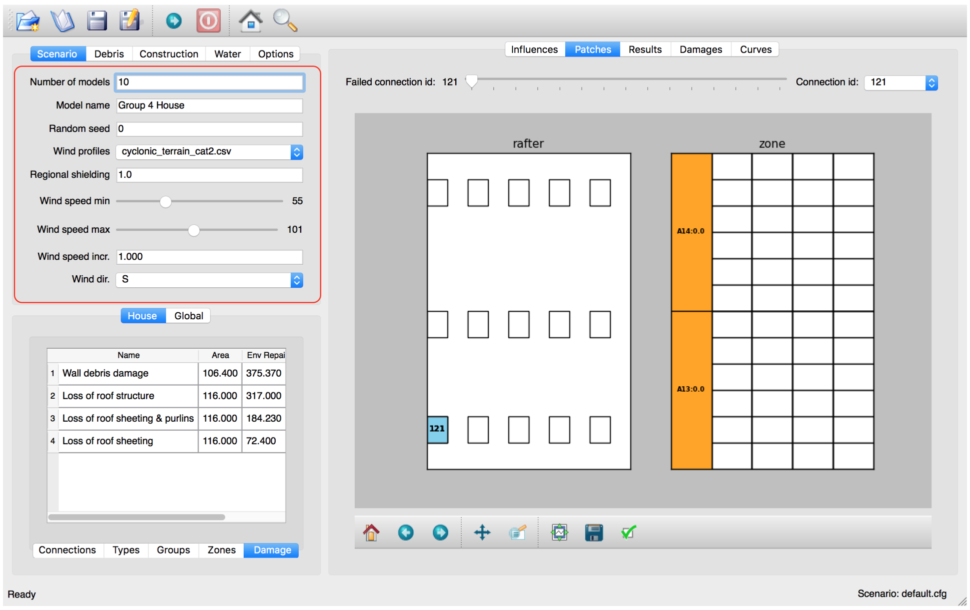

Fig. 3.1: Parameters of main section in the Scenario tab

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.
diff_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	angle (in degree) of debris sources
source_items	Source items	number of debris items per debris sources
flight_time_mean	Flight time mean	mean flight time of debris items
flight_time_stddev	Flight time std	standard deviation of flight time of debris items
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.

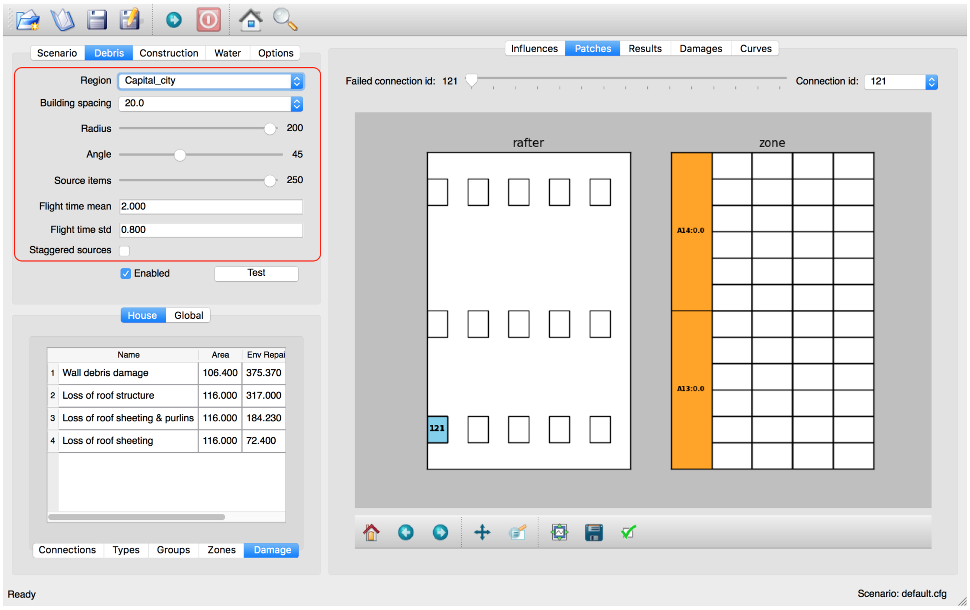

Fig. 3.2: Parameters of debris section in Debris tab

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)
cov_factors	Cov factors	comma separated list of cov factors of construction levels (default: 0.58, 0.58, 0.58)

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 ? 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)

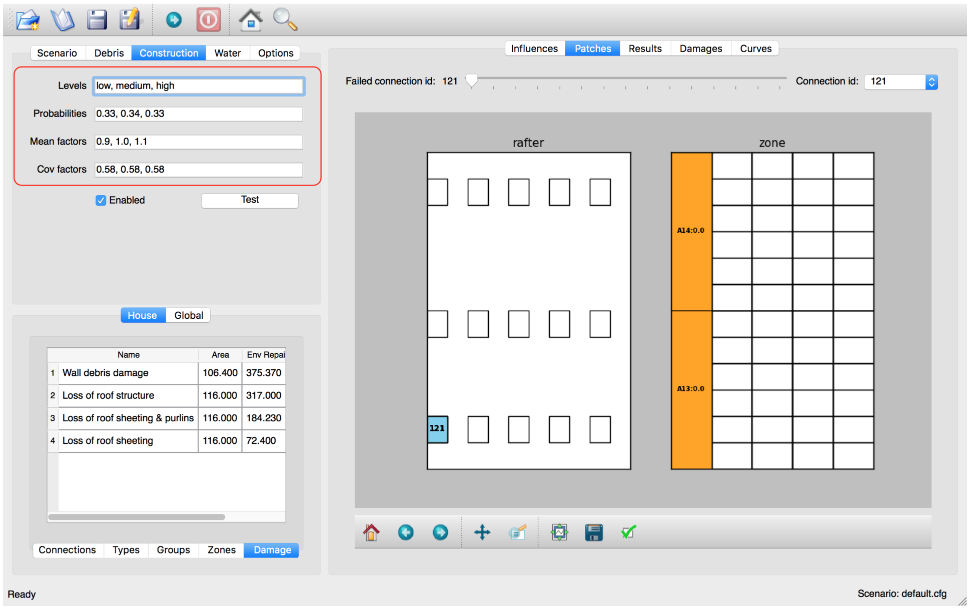

Fig. 3.3: Parameters of construction_levels section in Construction tab

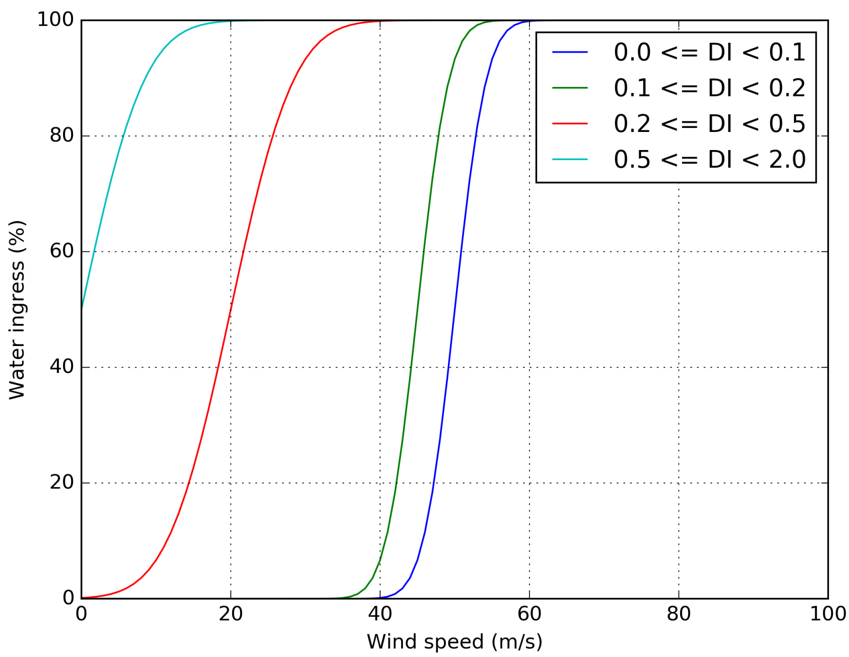

Fig. 3.4: Water ingress vs. wind speed for different ranges of damage index

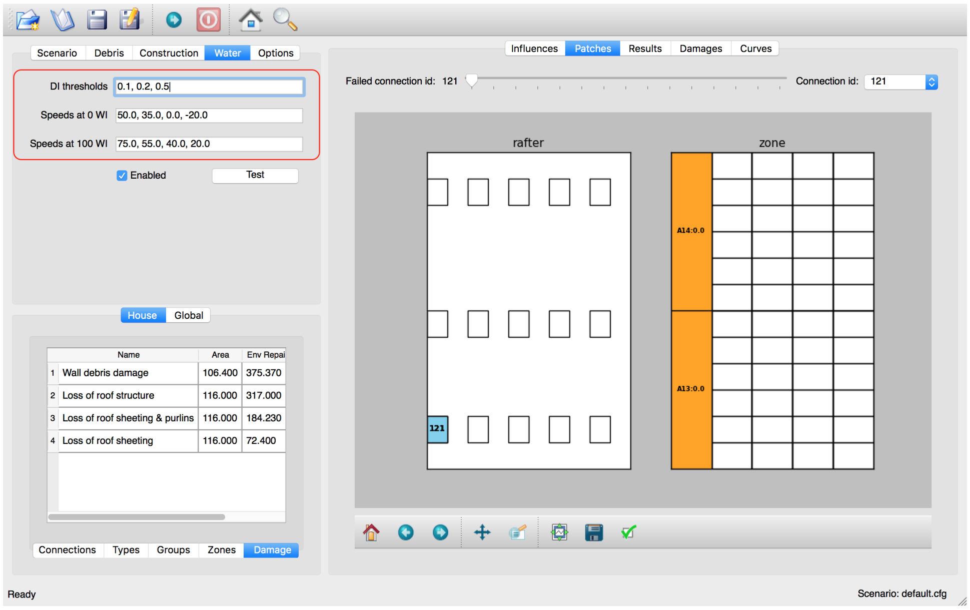

Fig. 3.5: Parameters of water_ingress section in Water tab

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 probability of exceeding a damage state ds at a wind speed x is calculated as (3.1):

$$P\left(DS \geq ds|\, x\right) = \frac{\sum_{i=1}^{N}\left[DI_{i|x} \geq t_{ds}\right]}{N} \tag{3.1}$$

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.

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)

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

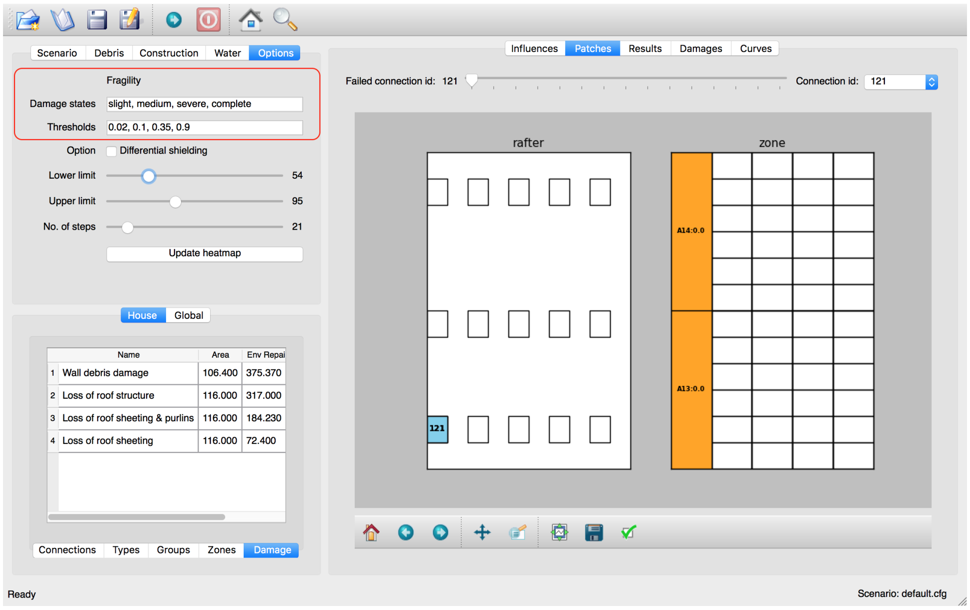

Fig. 3.6: Parameters of fragility_thresholds section in Options tab

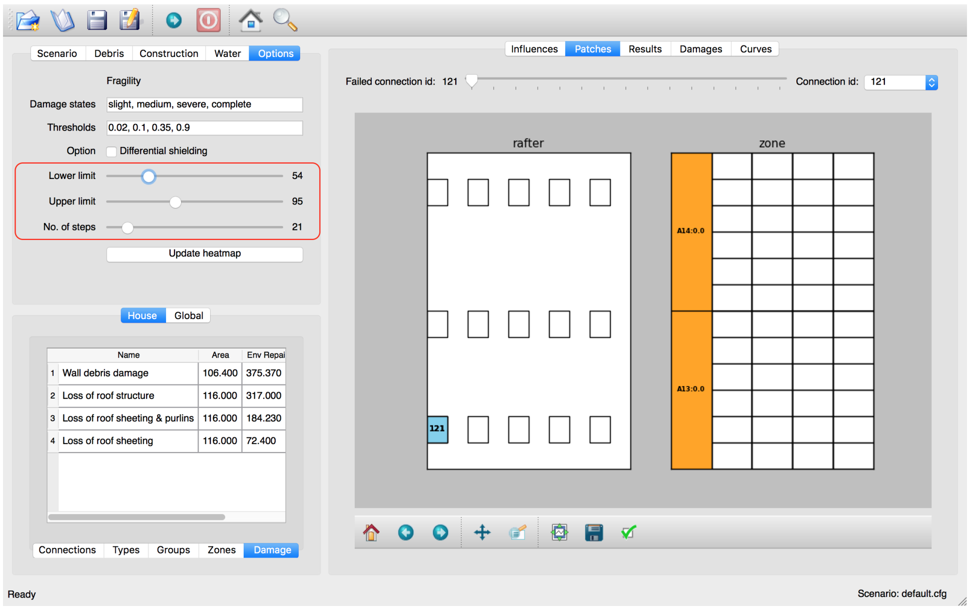

Fig. 3.7: Parameters of heatmap section in Options tab

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
Compact_cdav,0.65,0.65
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
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
```

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
mass_stddev	standard deviation of mass
frontal_area_mean	mean of frontal area (m^2)
frontal_area_stddev	standard deviation of frontal area (m^2)
cdav	average drag coefficient

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 in the default sceanrio with respect to Australian wind design categories: cyclonic_terrain_cat2.csv, cyclonic_terrain_cat2.5.csv, cyclonic_terrain_cat3.csv, and non_cyclonic.csv

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.

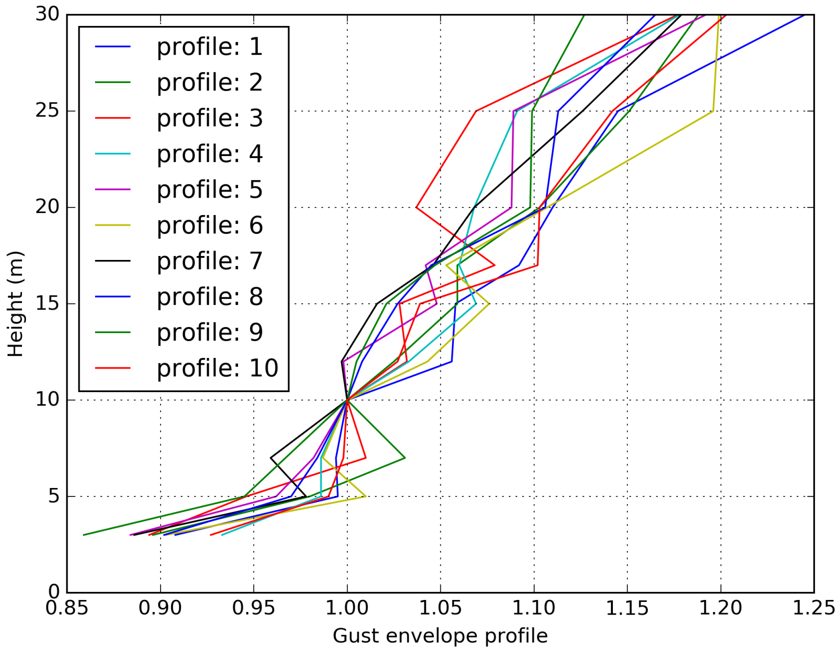

Fig. 3.8: Wind gust envelope profile along height.

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_cov,0.0
cpe_k,0.1
cpe_str_cov,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_cov	float	cov of Cpe for sheeting and batten
cpe_k	float	shape factor of Cpe for sheeting and batten
cpe_str_cov	float	cov of Cpe for rafters and eaves
cpe_str_k	float	shape factor of Cpe for rafters and eaves

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.

Listing 3.6: Example conn_groups.csv

```
group_name,dist_order,dist_dir,damage_scenario,trigger_collapse_at,patch_dist
sheeting,1,col,Loss of roof sheeting,0.0,1
batten,2,row,Loss of roof sheeting & purlins,0.0,1
rafter,3,col,Loss of roof structure,0.0,1
```

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	integer	direction of damage distribution; either 'col', 'row', or ''
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
patch_dist	integer	1 if influence patch is applied when connection is damaged otherwise 0

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. Provide 4 sets of x, y coordinates for a rectangular 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,wall_dir,coords,
A1,0.2025,0,0,0,0,0.2,0,0.2,0.5,0,0.5
A2,0.405,0.5,0,0,0.5,0.2,0.5,0.2,1,0,1
A3,0.405,1,0,0,1,0.2,1,0.2,1.5,0,1.5
A4,0.405,1,0,0,1.5,0.2,1.5,0.2,2,0,2
A5,0.405,1,0,0,2,0.2,2,0.2,2.5,0,2.5
```

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. Provide 4 sets of x, y coordinates for a rectangular shape.

3.4.6 zones_cpe_mean.csv

This file defines mean cladding Cpe 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 Cpe value in South direction
SW	float	mean cladding Cpe value in South West direction
W	integer	mean cladding Cpe value in West direction
NW	float	mean cladding Cpe value in North East direction
N	float	mean cladding Cpe value in North direction
NE	float	mean cladding Cpe value in North East direction
E	integer	mean cladding Cpe value in East direction
SE	float	mean cladding Cpe value in South East direction

3.4.7 zones_cpe_str_mean.csv

Like zones_cpe_mean.csv, mean Cpe 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 Cpe 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
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.

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 tge 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)
failure_momentum_std	float	standard deviation of failure momentum
failure_strength_in_mean	float	mean failure strength inward direction (kN)
failure_strength_in_std	float	standard deviation of failure strength inward direction
failure_strength_out_mean	float	mean failure strength outward direction (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 Cpe 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
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

$$C = x \times (A \times C_{\text{env}} \times R_{\text{env}} + C_{\text{int}} \times R_{\text{int}}) \tag{3.2}$$

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:

$$\begin{aligned} f_1 &= c_1 \times x^2 + c_2 \times x + c_3 \\ f_2 &= c_1 \times x^{c_2} \end{aligned} \tag{3.3}$$

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 ($/$m^2$)
envelope_factor_formula_type	type index of costing function for envelope
envelope_coeff1	c_1 in costing function for envelope
envelope_coeff2	c_2 in costing function for envelope
envelope_coeff3	c_3 in costing function for envelope
internal_repair_rate	repair rate for internal ($)
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

$$WC = x \times B \times C \tag{3.4}$$

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 (3.3).

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
```

CHAPTER

FOUR

USE OF THE GUI

4.1 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, 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.

4.2 Key features

- Component-based approach:

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

4.3 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 zone and coverage

  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

  Multiple construction levels can be defined with mean and cov factors which will be used to adjust the mean and cov of distribution of connection strength.

- Strength and dead load

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

4.4 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 each individual house for a specific wind event.

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 by random sampling

- For each model (`House`)
  - sample wind direction (`House.set_wind_orientation()`)
  - sample wind profile (`House.set_wind_profile()`)
  - sample construction quality level (`House.set_construction_level()`)
  - set up coverages given the wind direction (`House.set_coverages()`)
  - set up connections (`House.set_connections()`)
  - set up zones (`House.set_zones()`)
  - set up debris generation model (`House.set_debris()`)

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

- For each wind speed
  - calculate damage index for each model (`HouseDamage.run_simulation()`)
    * assign the increment in the mean damage index from the previous wind step as an input to the debris generation model (`Debris.no_items_mean`)
    * calculate free stream wind pressure (qz), optionally applying a regional shielding factor (`HouseDamage.compute_qz_ms()`)
    * calculate zone pressures from qz (`Zone.calc_zone_pressure()`)
    * check damage of envelope coverages by wind load (`Coverage.check_damage()`)
    * calculate connection loads (`Connection.compute_load()`)
    * check damage of each connection by connection group (`ConnectionTypeGroup.check_damage()`)
    * check damage and compute damaged area by connection group (`ConnectionTypeGroup.compute_damaged_area()`)

    * update influence by connection group (`ConnectionTypeGroup.update_influence()`)
    * check for total house collapse event (`HouseDamage.check_house_collapse()`)
    * compute damage index of the model (`HouseDamage.compute_damage_index()`)
    * compute damage index of the model (`HouseDamage.compute_damage_index()`)
    * generate debris and update Cpi in case of internal pressurisation event (`HouseDamage.check_internal_pressurisation()`)

  – calculate increment in mean damage index of the group of models (`update_bucket()`)

3. Fit fragility and vulnerability curves and save outputs (`save_results_to_files()`)

5.2 Detailed logic

-. Generate debris and check any breach by debris -. Update cpi in case of internal pressurization event

-. determine footprint and coverages given the wind direction -. sample cpe and calculate pressure -. sample construction quality level and determine factors applied to mean and cov of strength -. sample connection strength and dead load

5.2.1 Debris damage module

The methdology of modelling damage from wind-borne debris implemented in the code is described in the [**JDH2010d_**] and [**JDH2010d_**]. The debris damage module consists of four parts: 1) debris generation, 2) debris trajectory, 3) debris impact, and 4) debris damage costing.

Debris generation

The mean number of debris items to be generated $(N_{mean}) is calculated by : eq : `number_{o}f_{d}ebris_{i}tems_{e}q.$

$$N_{mean} = \text{nint}\left(\Delta DI \times N_{items}\right) \tag{5.1}$$

where ΔDI:, ΔDI:, ΔDI: increment in damage index from previous wind step, B: base cost, and C: costing function. Like the damage costing functions, two types of costing functions are defined as (3.3).

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

Depending on the value for *staggered_sources*, Fig. 5.2 and Fig. 5.3 are displayed.

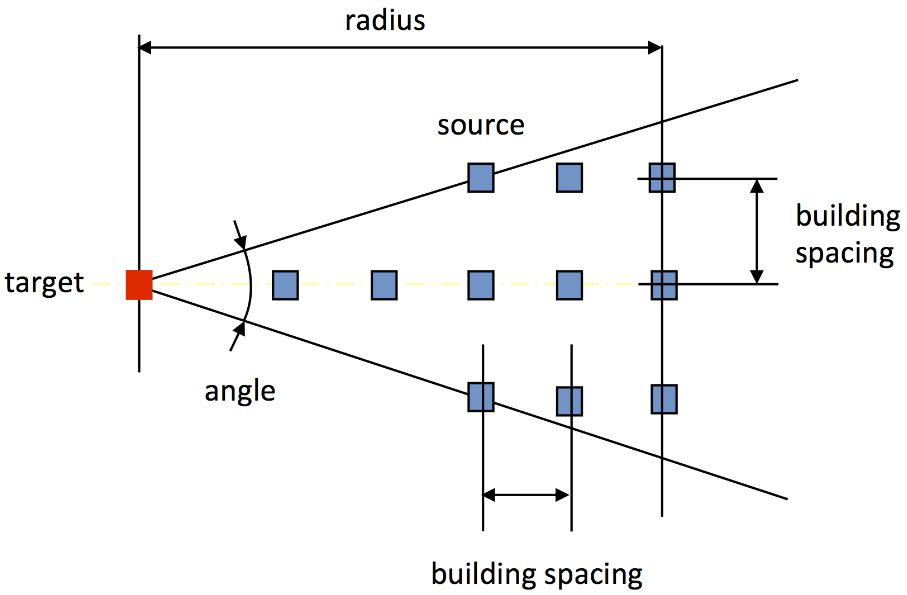

Fig. 5.1: Distribution of debris sources with parameters

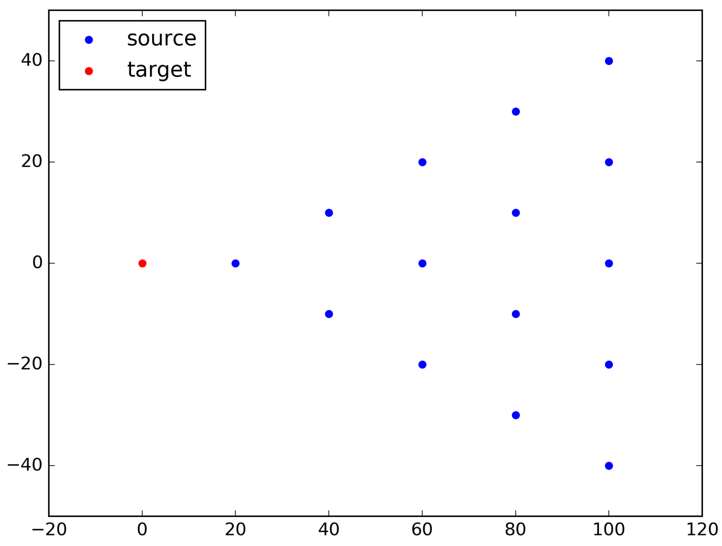

Fig. 5.2: Distribution of debris source buildings generated with debris_radius = 100.0 (m), debris_angle = 45.0 (deg), debris_space = 20.0 (m), and staggered_sources = *True*.

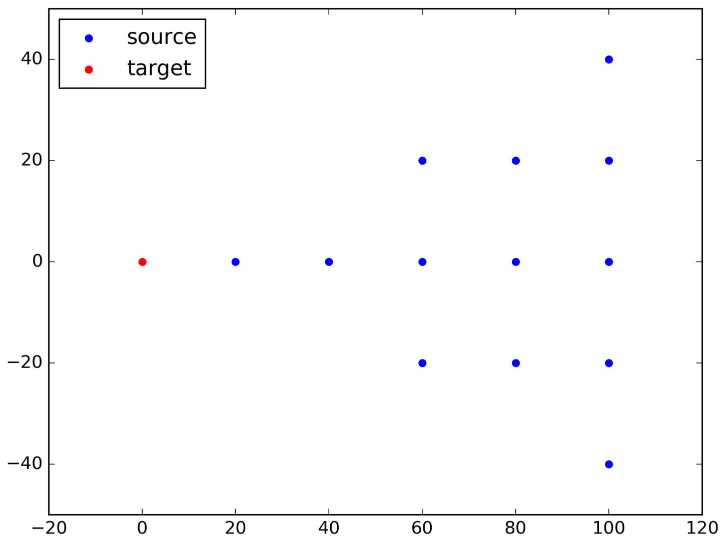

Fig. 5.3: Distribution of debris source buildings generated with debris_radius = 100.0 (m), debris_angle = 45.0 (deg), debris_space = 20.0 (m), and staggered_sources = *False*.

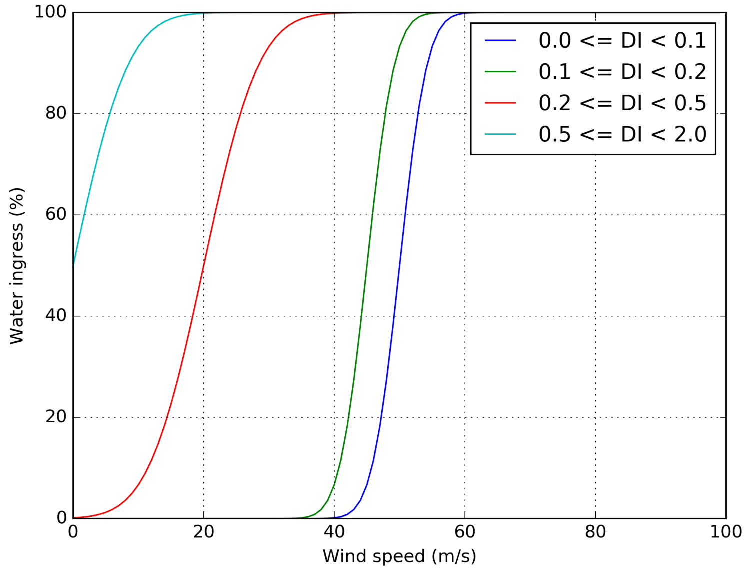

Fig. 5.4: water ingress vs. wind speed for various damage index values.

5.2.2 Water ingress

5.2.3 Cpe

Note that Cpe is used in computing zone pressures, and is sampled from Type III extreme value distribution (or Weibull) (`stats.sample_gev()`)

$$F(x; u, a, k) = \exp\left[-\left(1 - k\frac{x-u}{a}\right)^{\frac{1}{k}}\right]$$

where u: location factor ($\in \mathbb{R}$), a: scale factor (> 0), and k: shape factor (> 0). The mean and variance are calculated as follows:

$$\text{mean} = u + \frac{a}{k}\left[1 - \Gamma(1+k)\right]$$
$$\text{variance} = (\frac{a}{k})^2\left[\Gamma(1+2k) - \Gamma^2(1+k)\right]$$

The u and a are estimated given cov and k values:

$$a = \text{mean} \times \frac{\text{cov}}{B}$$
$$u = \text{mean} - a \times A$$

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

5.2.4 Costing damage

ix. cost damage 1. Calculate percentage of damaged area per connection group. 2. Translate percentage into repair cost via damage scenarios. 3. cost damage from water ingress if required 4. calculate damage index

fragility Fragility “implies easily damaged or broken, but is often used to describe the probability of a stated level of damage for a specific hazard, e.g. an earthquake.” Fragility measures probability.

fragility function

fragility curves Fragility function is a mathematical function that expresses the relationship between the probability of occurrence of some undesirable event and some measure of environmental excitation. In our case the the undesirable event is a facility or component reaching or exceeding some clearly defined limit state, and the environmental excitation is an earthquake of a defined intensity measure.

vulnerability Vulnerability refers to the concept of susceptibility of damage for a given entity. The entity can be a civil structure, a critical infrastructure facility, a component within such a facility, or a subset of population within a defined geographical area, etc. Vulnerability measures loss.

vulnerability function Vulnerability function is a mathematical function that depicts loss as a function of environmental excitation. It has several synonyms: vulnerability curves, damage functions, loss functions.

BIBLIOGRAPHY

[JDH2010a] JDH Consulting, Randomized wind gust profiles, Report to Geoscience Australia, 2010.

[GA2011] Geoscience Australia, 2011. Wind Vulnerability Model Development for Adaptation Studies on Specific Residential and Industrial Buildings. Report to the Department of Climate Change and Energy Efficiency, Geoscience Australia.

[JDH2010b] JDH Consulting, 2010. Contributions to Development of a Wind Vulnerability Model - Final Report 2009-10. Report to Geoscience Australia.

[SA2011] Standards Australia, 2011. Australian/New Zealand Standard AS/NZS 1170.2:2011 Structural design actions Part 2: Wind actions.

[TuTo2006] Turner & Townsend Rawlinsons (T&TR), 2006. Wind Damage Cost Module Development for North Queensland Building Types. Report to Geoscience Australia in four volumes.

[JDH2010c] Holmes, J.; Wehner, M.; Sandland, C. & Edwards, M. Modelling damage to residential buildings from wind-borne debris – Part 1. Methodology 14th Australasian Wind Engineering Society Workshop, 2010

[JDH2010d] Holmes, J.; Wehner, M.; Sandland, C. & Edwards, M. Modelling damage to residential buildings from wind-borne debris – Part 2. Implementation 14th Australasian Wind Engineering Society Workshop, 2010

INDEX