Disaggregation

Overview

Oasis allows detailed hazard models to be executed in an efficient manner. However, often the inputted data is aggregated, low resolution or missing key attributes. This is particularly true in the developing world, a key focus of the Oasis model library. Currently in the Oasis platform, as of July 2018, exposure must be converted into detailed data before being imported into the system for analysis. Typically, the model builder will provide a tool to do this conversion.

This document proposes an approach to stochastic disaggregation that would be incorporated into the Oasis execution kernel. This will perform on the fly sampling from one or more hazard and/or vulnerability distributions based on pre-defined weight tables. This approach provides a full treatment of the uncertainty associated with the disaggregation and will be computationally and data efficient. Currently the disaggregation work has only been done in regards to the number of locations and split TIV (see use cases below). Further work will be done to handle financial conditions associated with aggregate exposure in the future.

Approach and Integration

In the getmodel component, the weight of the relevant combinations of areaperil_id and vulnerability_id will be applied to the hazard and vulnerability distributions respectively to create the CDF for the aggregate_areaperil_ids and aggregate_vulnerability_ids present in the aggregate_item file. If both the areaperil and vulnerability are aggregated, the weights will be applied to the damage CDFs for each detailed attribute combination. The output of getmodel will remain the same – event_id, areaperil_id, vulnerability_id and CDF results, and the expanded items file will be of the same structure as before which will allow gulcalc and all following components to run as normal. The weight for some combination of detailed attributes is found by dividing the count of buildings with those attributes by the total count of relevant buildings. getmodel can still be used for non-disaggregation cases, and disaggregation will only be done if the option -d is added.

Example Weight Calculation:

areaperil_id	vulnerability_id	count
101	101	300
101	102	200
102	101	200
102	102	300

Table 2: Weights

aggregate_areaperil_id	areaperil_id
100001	101
100001	102

Table 1: Aggregate areaperil to areaperil

Given an item with aggregate_areaperil_id 100001 and vulnerability_id 101, the weight for areaperil_id 101 is $\frac{300}{200+300} = 0.6$

Modifications:

keys server → eve 1 1 | getmodel -d | gulcalc -S100 -c - | summarycalc -g -1 - | eltcalc

GenerateTestData.py – additional -disaggregation option added to generate necessary files

ktest – additional checksums added to verify getmodel -d output with additional files in examples

Current Model Tables:

- Areaperil_dict: representation of an area for a particular peril with areaperil_id (integer values) as keys
- Vulnerability_dict : representation of vulnerability functions with vulnerability_id (integer values) as keys
- Intensity_bin_dict: represents a discretised, abstracted set of intensity measures that are specific to the peril (or perils) represented by the events
- Damage_bin_dict: represents a discretised, abstracted set of damage ratios
- Event: representation of events (synthetic or historic) with event_id (integer values) as keys

- Occurrence: information of occurrences of events over time – used for time-based calculations such as Average Annual Loss and Loss Exceedance Curves
- Footprint: describes the interaction of events with area perils, giving a probability distribution of intensity for each combination
- Vulnerability: describes the interaction of intensities with vulnerability types, giving a probability distribution of damage for each combination

Disaggregation Tables:

- Weights: mapping of risk count to vulnerability_id and areaperil_id
- Aggregate_ap_to_ap: mapping of detailed areaperil_ids included in each aggregate_areaperil_id
- Aggregate_vul_to_vul: mapping of detailed vulnerability_ids included in each aggregate_vulnerability_id

The disaggregation tables should reference existing model data i.e. areaperil_id 1 in the aggregate_ap_to_ap table refers to the same thing as areaperil_id 1 in the areaperil_dict table. Aggregate IDs must be over some threshold (i.e. larger than all detailed IDs) and will be read from the first line of the disaggregation tables.

areaperil_dict

areaperil_id	lat	lon
101	33.88	35.50
102	33.88	35.56
103	33.88	35.62
104	33.88	35.44

event

event_id
1
2

occurrence

event_id	period_no	occ_year	occ_month	occ_day
1	1	1	9	3
2	1	1	10	18
1	4	4	8	16
2	10	10	9	24

vulnerability_dict

vulnerability_id	construct_scheme	construct_code
101	ATC	1
102	ATC	2

intensity_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	50	60	55	1202
2	60	70	65	1202
3	70	80	75	1202

damage_bin_dict

bin_index	bin_from	bin_to	interpolation	interval_type
1	0	0.4	0.2	1202
2	0.4	1	0.7	1202

Figure 1: Basic windstorm model without disaggregation tables

Use Cases

A weight table showing a count of buildings with each combination of areaperil/vulnerability IDs will be required in each of the use cases. The count can be taken directly as the number of buildings with the corresponding attributes or as normalised floating point numbers. This can be provided by various models such as census or economic data. Different sets can then be selected and compared in the analysis step. Models will typically have building inventories that provide data on the occupancy and vulnerability mix by geography as well. By calculating the appropriate ratios, these weights can be used to calculate new hazard and vulnerability curves. This is already the standard approach, but going forward there will be no need to pre-compile these curves. This will decrease data volumes and weight tables will make assumptions more transparent.

Use case 1 – low resolution exposure

This is the case where the data on exposure and vulnerability are known but exact geographical location is unknown, or the original grid of area perils is of low resolution. In table for aggregate area peril to area perils will need to be provided. The weights for each area peril in the aggregate area peril is found, and is then applied to the corresponding hazard curves. These weighted curves are then combined to make a new hazard curve for this particular aggregate area peril and detailed vulnerability combination, and can be used to get the damage CDF as usual.

Figure 2: Aggregate area peril grid

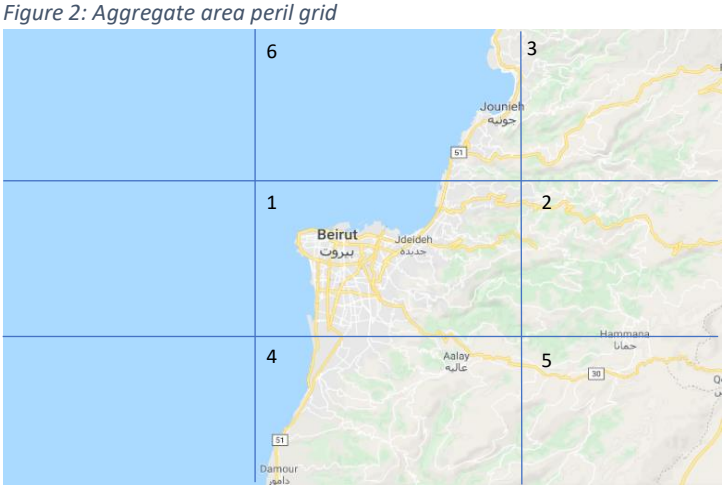
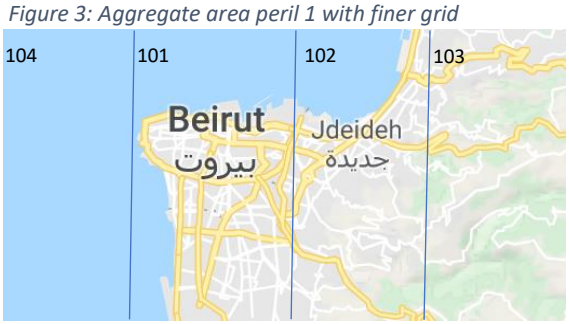


Figure 3: Aggregate area peril 1 with finer grid



Use case 2 – unknown attributes

This is the case where some of the data required for mapping an exposure to a particular vulnerability curve is unknown (e.g. Construction code, construction scheme, occupancy type). In this case an additional table is required to show the vulnerability IDs included in each aggregate vulnerability. The weights for each vulnerability curve is applied and all the weighted curves are combined to make a new aggregate vulnerability curve for the aggregate vulnerability and detailed area peril combination. This can then be used to create the damage CDF as usual.

Use case 3 – area peril and vulnerability aggregated

This is the case where both detailed geographical location and some attribute data are unknown. In this situation the damage CDF is found for each detailed attribute combination, weighted and combined. This is then written to gulcalc as usual.

Use case 4 – aggregation exposure

In this case, instead of just one property with some unknowns, a group of multiple properties with similar unknowns is given. The aggregate TIV is known but individual TIVs are not. The tables for area perils and vulnerability curves are needed and the value for individual items will be taken as the average TIV. In this situation, the aggregated exposures will be expanded into a separate item file with a changed exposure ID which is linked to the average TIV. New exposure IDs will also be appended onto the exposure file. These individual items can be treated as having the same "group_id" if the user wishes for the items to be correlated.

Examples

event_id	areaperil_id	intensity_bin_index	prob
1	101	1	1
1	102	1	1
2	101	2	1
2	102	3	1

Table 3: Basic windstorm model footprint table

vulnerability_id	intensity_bin_index	damage_bin_index	prob
101	1	1	0.9
101	1	2	0.1
101	2	1	0.8
101	2	2	0.2
101	3	1	0.7
101	3	2	0.3
102	1	1	0.8
102	1	2	0.2
102	2	1	0.5
102	2	2	0.5
102	3	1	0.1
102	3	2	0.9

Table 4: Basic windstorm model vulnerability table

item_id	coverage_id	areaperil_id	vulnerability_id	number_items	grouped
1	1	102	100001	1	0
2	1	100001	102	2	0
3	2	100002	100001	2	1

Table 5: Aggregate_items

item_id	coverage_id	areaperil_id	vulnerability_id	group_id
1	1	102	100001	1
2	3	100001	101	2
3	3	100001	101	3
4	4	100001	100001	4
5	4	100001	100001	4

Table 6: Expanded items

Low Resolution Exposure

aggregate_areaperil_id	areaperil_id
100001	101
100001	102

Table 3: Agg_ap_to_ap

areaperil_id	vulnerability_id	count
101	101	300
101	102	200
102	101	200
102	102	300

Copy of Table 2 - Weights

Item 2 is in aggregate area peril 100001 with vulnerability 101 and is worth $100k. Given the basic windstorm model, Table 2 and 6, the following is generated for Event 2:

vulnerability_id	aggregate_areaperil_id	areaperil_id	weight
101	100001	101	0.6
101	100001	102	0.4

areaperil_id	intensity_bin_index	prob
100001	2	0.6
100001	3	0.4

damage_bin_index	prob
1	0.76
2	0.24

Random numbers	*Sample Loss*
0.518701437	$20k
0.658979807	$20k
0.259967481	$20k

- Sample mean loss is $20k for this simulation
- The damage CDF for Item 3 will be the same
- But sample mean loss may differ as the items are not grouped so the random numbers used for sampling will be different

Unknown Attributes

aggregate_vulnerability_id	vulnerability_id
100001	101
100001	102

Table 4: Agg_vul_to_vul

areaperil_id	vulnerability_id	count
101	101	300
101	102	200
102	101	200
102	102	300

Copy of Table 2 - Weights

Item 1 is in area peril 102 with aggregate vulnerability 100001 and is worth $200k. Given the basic windstorm model, Table 2 and 7, the following is generated for Event 2:

areaperil_id	aggregate_vulnerability_id	vulnerability_id	weight
102	100001	101	0.4
102	100001	102	0.6

vulnerability_id	intensity_bin_index	damage_bin_index	prob
100001	1	1	0.84
100001	1	2	0.16
100001	2	1	0.62
100001	2	2	0.38
100001	3	1	0.34
100001	3	2	0.66

damage_bin_index	prob
1	0.34
2	0.66

Random numbers	*Sample Loss*
0.382659384	$140k
0.836472958	$140k
0.553480293	$140k

- Sample mean loss is $140k for this simulation

Area Peril and Vulnerability Aggregated

Item 4 is in aggregate area peril 100001 with aggregate vulnerability 100001 and is worth $200k. Given the basic windstorm model, Table 2 and 7, the following is generated for Event 2:

aggregate_areaperil_id	aggregate_vulnerability_id	areaperil_id	vulnerability_id	weight
100001	100001	101	101	0.3
100001	100001	101	102	0.2
100001	100001	102	101	0.2
100001	100001	102	102	0.3

areaperil_id	vulneability_id	damage_bin_index	prob
101	101	1	0.8
101	101	2	0.2
101	102	1	0.5
101	102	2	0.5
102	101	1	0.7
102	101	2	0.3
102	102	1	0.1
102	102	2	0.9

damage_bin_index	prob
1	0.51
2	0.49

Random numbers	*Sample Loss*
0.602934823	$140k
0.209582954	$40k
0.103948583	$40k

- Sample mean loss is $73.33k for this simulation
- Item 5 will have the same damage CDF and the same sample losses as the two items are grouped and are sampled with the same random numbers