Contents

[Introduction 2](#_Toc47105494)

[Additional Features: 2](#_Toc47105495)

[Using the Zhong Implementation 3](#_Toc47105496)

[Important Conceptual Note 3](#_Toc47105497)

[Model Parameters 3](#_Toc47105498)

[**age\_group\_population:** 3](#_Toc47105499)

[**susceptible:** 4](#_Toc47105500)

[**fatalities:** 4](#_Toc47105501)

[**infected:** 4](#_Toc47105502)

[**recovered:** 5](#_Toc47105503)

[**correlation:** 6](#_Toc47105504)

[**disobedient:** 6](#_Toc47105505)

[**virulence\_rates:** 6](#_Toc47105506)

[**recovery\_rates:** 7](#_Toc47105507)

[**mobility\_rates:** 8](#_Toc47105508)

[**fatality\_rates:** 8](#_Toc47105509)

[**infection\_correction\_factors:** 9](#_Toc47105510)

[**hospital\_infected\_capacity:** 10](#_Toc47105511)

[**over\_capacity\_fatality\_modifier:** 10](#_Toc47105512)

[**cells and neighbourhood:** 11](#_Toc47105513)

[**Example Input File** 12](#_Toc47105514)

# Introduction

The Zhong model is a SIRS-like model. The population of a region can be divided into:

* Susceptible: Those who can become infected
* Infected: those who are currently sick and pass on the infection to others
* Recovered: Those who have recovered from the infection and are immune from further infection for a certain period of time

After being in the infected state for some time, individuals move onto the recovered phase; after some time being classified recovered, those individuals are now susceptible again.

Both the infection and recovery stages have sub-stages; an individual enters either the infection or recovery in the first respective sub-stage, then moves onto the next sub stage until they are in the last sub stage. After the last sub stage they move onto the next state stage, such as infection from recovery.

The reference is at:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7088542/pdf/11430_2009_Article_44.pdf>

# Additional Features:

The aforementioned Zhong model was used as a base to implement a new model, henceforth called GeographyModel. The implementation of the *name\_of\_new\_model* adds additional features to more closely model what is seen in the real world:

* Fatalities
* Travel restrictions due to lockdown
* Hospital capacity
* Age groups

Furthermore, the original Zhong model calculates a number representing connectivity between cells by taking into account using their border lengths as well as interconnecting roads. However, GeographyModel assumes the connectivity, also called correlation, is computed beforehand and used in the input file for the model. The correlation between cells specified in the input file can be based off of the user’s choice. This means that the correlation number between two cells in the input file could have taken interconnecting roads into account like the Zhong model, or could have completely ignored them, or used other geographical data to calculate that correlation factor.

# Using the Zhong Implementation

The input to the Zhong model is controlled by a JSON file that controls the parameters for the simulation. This file is specified as a program argument; for example, to launch the Zhong implementation with an input file “parameters.json”:

***(On Linux)* ../compiled\_GeographyModel\_implementation directory\_with\_input\_file/parameters.json**

# Important Conceptual Note

Population percentage are specified in the range of [0, 1], with 1 being equal to 100%.

The simulation has support for age groups; these age groups are independent except when calculating new infections, which takes in the entire infected population into account. This can be viewed as:

All calculations for each age group are done independently, and so is the information stated in the input file for each age group.

For example, suppose there are two age groups. For each age group:

*Susceptible + Infected + Recovered + Fatalities = 1*

However, if each age group represents half of the total population, then the total number of:

* Susceptible: ***Susceptible Age Group 1 \* 0.5 + Susceptible Age Group 2 \* 0.5***
* Infected: ***Infected Age Group 1 \* 0.5 + Infected Age Group 2 \* 0.5***
* Recovered: ***Recovered Age Group 1 \* 0.5 + Recovered Age Group 2 \* 0.5***
* Fatalities: ***Fatalities Age Group 1 \* 0.5 + Fatalities Age Group 2 \* 0.5***

# Model Parameters

## **age\_group\_population:**

Defines the number of age segments the population is divided into, and how much of the population is a represented by a certain age group.

Example:

"age\_group\_proportions": [0.35, 0.30, 0.25, 0.1, 0],

In this array, there are five numbers, representing five age groups. The first age group represents 35% of the total population, the second age group 30% of the population, and so on.

It should be noted that specific age range that each group specifies is not explicit- it is up to the person running the models. In the above example, the first age group can represent ages of [0, 20] or [40, 80]. To reflect this fact, the appropriate configuration for each group is modified to change the age range that group is representing.

The valid values in each sub array is [0, 1]. The sum of all elements in the array must not exceed 1.

## **susceptible:**

Represents the proportion of an age group that is currently in the susceptible age group.

Example:

"susceptible": [0.50, 0.50, 1],

In the above example the last age group starts with all of its individuals susceptible. In other words, there are no infected or recovered individuals for the last age group; the first two age groups have half of their population susceptible, with the other half either infected, recovered, or casualties.

The valid values for each element in the array is [0, 1].

## **fatalities:**

Represents the proportion of an age group that has succumbed to the disease.

Example:

"fatalities": [0, 0.15, 0],

The above array indicates that for all age groups except for the second one there are no fatalities. However, for the second age group 15% of its population has succumbed to the disease.

The valid values for each element in the array is [0, 1].

## **infected:**

Represents the proportion of an age group that is in each stage of infection.

Example:

“infected”: [

[0.05, 0.13, 0.01, 0, 0],

[0, 0, 0, 0, 0]

[0, 0, 0, 0, 0]

]

Notice how this is an array of arrays. Each sub array represents an age group, meaning the number of sub arrays must equal the number of age groups.

The length of a sub array denotes the number of infection stages. All sub arrays must be of equal length depicting an equal number of infection stage regardless of the age group.

In the first age group, represented by the first sub array, there are non-zero values. This means that there are infected individuals in the first age group. More specifically, 5% of the population in the age group is in the first infection stage; 13% in the second infection stage, and 1% in the third infection stage.

The sum of all of the values in a sub array is the percentage of the population of an age group that is infected. In this case, the first age a group has (5 + 13 + 1) 19% of its population infected; the other age groups have no infected individuals.

There must be an equal number of sub arrays as there are age groups. The valid values in each sub array is [0, 1]. The sum of all elements in each sub array must not exceed 1.

## **recovered:**

Represents the proportion of an age group that is in each stage of recovery.

For example:

“recovered”: [

[0.1, 0.1, 0, 0, 0],

[0.2 0.1, 0.05, 0, 0]

[0, 0, 0, 0, 0]

]

This format is similar to the infected array of arrays. The only difference is what is being represented.

The first age group has (10 + 10) 20% of its population recovered, while the second age group has (20 + 10 + 5) 35% of its population recovered.

There must be an equal number of sub arrays as there are age groups. The valid values in each sub array is [0, 1]. The sum of all elements in each sub array must not exceed 1.

## **correlation:**

Represents how much of an influence a neighbouring cell has on a different cell.

Example:

1. correlation = 1
2. correlation = 0

In part a) of the above example, a cell has a lot of influence on a neighbouring cell. Meanwhile in part b), a cell has no influence on a different cell.

The valid values for a correlation are [0, 1].

## **disobedient:**

The percentage of the population for each age group that ignores any restrictions on movement.

Example:

“disobedient”: [1.0, 0.5, 0.0]

The above example shows that the first age group is completing ignoring any restrictions on travel. As a result any mobility restrictions have no effect on that age group, resulting in the infection rates likely not decreasing.

The second age group is partially adhering to the mobility restrictions, while the last age group is being fully compliant.

The valid values for disobedience are [0, 1].

## **virulence\_rates:**

Specifies how infectious the virus is (how much it spreads). Used in the calculation for new infections. This used for all age groups.

Example:

“virulence rates”: [

[0.15, 0.18, 0.15],

[0, 0, 0],

[0.7, 0.7, 0.7]

]

Each value in the sub array represents how infectious a population is at each stage of infection. Thus for the first age group, the second stage of infection is when they are the most infectious.

Each sub array represents a different age group. Thus the second age group, represented by the second sub array does not spread the infection at all, while the first age group spreads it a little bit while the third age group spreads the infection a lot.

There must be an equal number of sub arrays as there are age groups, and each value in each sub array must be greater than 0.

## **recovery\_rates:**

Specifies how much of the population at a certain infection stage recovers from the infection before going through all of the infection stages. This is applied for each age group.

Example:

“recovery rates”: [

[0.2, 0.1, 0.4, 0.0],

[0, 0, 0, 0],

[0.4, 0.2, 0.1, 0.4]

]

The first sub array shows for those in the first age group that are in the first stage of infection, 20% recover. Those that are still infected move onto the next stage, where 10% recover. Those still infected move onto the next stage of infection, where 40% recover. This continues for all of the recovery rates and for all of the age groups.

It should be noted that the number of recovery rates in each sub array is one less than the number of infection stages. This is because on the last stage of infection, all those still infected are now considered recovered. Thus no recovery rate is needed for that last stage of infection (can be thought of as an implicit recovery rate of 1 after the last number in each sub array).

Each recovery rate must be in the range of [0, 1]; having duplicate values of 1 for the same sub array is redundant.

## **mobility\_rates:**

Specifies how free individuals are free to travel during each stage of infection. This applies to all age groups.

Example:

“mobility\_rates”: [

[0, 0.4, 0.6, 1, 1],

[0, 0, 0, 0, 0],

[0.4, 0.2, 0.1, 0.4, 0.7]

]

For the first age group, again represented by the first sub array, those on the first stage of infection are not allowed to travel at all. Then, people move onto the next infection stage, they are become freer to travel. Contrast this with the second age group, where no matter the stage of infection they are in, they are not allowed to travel.

There must be an equal number of sub arrays as there are age groups. Each value in the sub arrays must be in the range of [0, 1], and each sub array must have as many values as there are infection stages.

## **fatality\_rates:**

Specifies how many of those infected succumb to the disease for each stage of infection. This applies to all age groups.

Example:

“fatality\_rates”: [

[0.005, 0.002, 0.001, 0.006],

[0, 0, 0, 0, 0],

[0.1, 0.05, 0.02, 0.009]

]

It can be seen above that those infected that are a part of the second age group never succumb to the disease, regardless of the infection stage they are in. The other two age groups cannot say the same thing. Half a percent (0.05%) of those in the first age group that are in the first stage infection succumb to the disease for example.

There must be an equal number of sub arrays as there are age groups. Each value in the sub arrays must be in the range of [0, 1], and each sun array must have as many values are there are infection stages. Having duplicate values of 1 for the same sub array is redundant.

## **infection\_correction\_factors:**

Determines how much to restrict movement above a certain infected population threshold, and for how long.

Example:

“infection\_correction\_factors”: {“0.2”: [0.7, 0.1], “0.5”: [0.5, 0.2]}

The format of each infection threshold is of the format:

“total\_infected\_population”: [additional\_mobility\_restriction\_factor, hysteresis]

The example shown above has two such threshold. Looking at the first one:

total\_infected\_population: 0.2.

*The associated restriction factor applies when 20% of the entire population is infected. Note that this not 20% of an age group!*

additional\_mobility\_restriction\_factor: 0.7

*The additional travel restriction on top of any number specified in the mobility\_rates arrays. Say the default mobility rate is 0.6; then the new travel freedom is no longer 0.6, but (0.6 \* 0.7) 0.42. This is the case as soon as at least 20% of the population is infected.*

hysteresis: 0.1

*This specifies how long to apply the additional travel restriction even after the entire infected population drops below the initial threshold. Thus the travel restriction above would start when the entire infection population is 20%, but continue on until the entire infected population is still (20% - 10%) 10% of the entire population.*

*This is implemented as generally speaking it is probably a good idea to be more cautious about travelling a lot again if such travel caused an upswing of infection.*

There are few other things to note:

* The infection threshold must be in the range of (0, 1). These two values are implicitly specified. For the above example, the infection threshold [0, 0.2) is implicit, and the last infection threshold implicitly goes to 1 (so the last infection range actually applies to an entire infected population range of [0.5, 1] / [50%, 100%].
* The restriction factor must be in the range of [0, 1]
* The hysteresis factor must be in the range of [0, infection\_threshold). If the infection threshold is 0.2 as in the example above, then the valid range of values are [0, 0.2].
* The infection thresholds specify the lower bound of the next infection threshold. For example, in situation given earlier the next threshold is when the entire population is 50% infected. Thus the restriction factor of 0.7, associated with the infected threshold 0.2, applies to the range [0.2, 0.5), or [0.1, 0.5) with the hysteresis specified.
* No infection threshold can be specified at all, indicated an implicit infection threshold of [0, 1] with no restriction factor.

## **hospital\_infected\_capacity:**

Specifies how much of the entire population can get infected before the medical system collapses. Closely tied with *over\_capacity\_fatality\_modifier.*

Example:

“hospital\_infected\_capacity”: 0.1

The value range of this variable is [0, 1].

Thus the medical system can only treat infected individuals if less than (or equal to) 10% of the entire population is infected.

## **over\_capacity\_fatality\_modifier:**

Specifies how many more fatalities there will be if the medical system collapses. This is only applied if the total proportion of the entire population is above the *hospital\_infected\_capacity*.

Example:

“over\_capacity\_fatality\_modifier”: 1.2

There will be 1.2x, or 20% more fatalities if the medical system collapses.

The value range of this variable is [1, some\_value\_greater\_than\_one]. Any increase by the multiplier will be capped to the number of people infected minus those who recovered. However, on the last stage of infection, the number of fatalities is limited to the number of infected on the last stage. This is due to a limitation of the model implementation.

## **cells and neighbourhood:**

This is where the neighbourhood for the simulation is defined. To specify a neighbour, these two pieces of information need to be specified for each cell:

* State: initial susceptible, initial infected, initial recovered, etc. See the “default\_state” section in the example input file.
* Vicinity: how much influence a neighbouring cell has on this cell. See the “default\_vicinity” section in the example input file.

If the default values for either state or vicinity are sufficient, then there is no need to provide that information; otherwise all elements of the above pieces of information are required.

The general structure to specify a cell is like so (*italics represent comments)*:

{

“cell\_id”: someID

“state”: { …state information… } *If the default state is fine, omit this*

“neighbourhood”: [

{“cell\_id”: “someID” }

*A cell’s neighbour always include itself. No vicinity specified, so the default vicinity is used.*

*Optional neighbouring cells (if there are none, omit the following section)*

{

*This cell must be specified in the neighborhood*

“cell\_id”: “someOtherCell”,

*Default vicinity not valid to represent relationship between these two cells.*

“vicinity”: { “correlation”: 0.2 }

}

]

},

*Specify the other cells used in the neighbourhood*

{

“cell\_id”: *someOtherCell,*

*State information if non-default is required*

“neighbourhood” [

{“cell\_id”: someOtherCell },

{

“cell\_id”: “someCell”,

*This vicinity’s correlation factor should be the same as mentioned in the neighbourhood of “someCell”*

“vicinity”: { “correlation”: 0.2 }

}

]

},

… other cells….

}

## **Example Input File**

Due to the size of this example file, it is split across three pages.

{

"scenario": {

"default\_delay": "inertial",

"default\_cell\_type": "zhong",

"default\_state": {

"age\_group\_proportions": [0.3, 0.3, 0.4],

"susceptible": [1, 1, 1],

"fatalities": [0, 0, 0],

"infected": [

[0, 0, 0, 0, 0],

[0, 0, 0, 0, 0],

[0, 0, 0, 0, 0],

],

"recovered": [

[0, 0, 0, 0, 0],

[0, 0, 0, 0, 0,

[0, 0, 0, 0, 0],

]

},

"default\_vicinity": {

"correlation": 1

},

"default\_config": {

"zhong": {

"disobedient": 0.0,

"precision": 100000000,

"virulence\_rates": [

[0.15, 0.15, 0.15, 0.15, 0.15],

[0.15, 0.15, 0.15, 0.15, 0.15],

[0.15, 0.15, 0.15, 0.15, 0.15],

],

"recovery\_rates":

[

[0.07, 0.07, 0.07, 0.07],

[0.07, 0.07, 0.07, 0.07],

[0.07, 0.07, 0.07, 0.07],

],

"mobility\_rates": [

[0.6, 0.6, 0.6, 0.6, 0.6],

[0.6, 0.6, 0.6, 0.6, 0.6],

[0.6, 0.6, 0.6, 0.6, 0.6],

],

"fatality\_rates": [

[0.005, 0.005, 0.005, 0.005, 0.005],

[0.005, 0.005, 0.005, 0.005, 0.005],

[0.005, 0.005, 0.005, 0.005, 0.005],

],

"infection\_correction\_factors": {"0.2": [0.7, 0.1], "0.4": [0.5, 0.2]},

"hospital\_infected\_capacity": 0.1,

"over\_capacity\_fatality\_modifier": 1.2

}

}

},

"cells": [

{

"cell\_id": "Country1",

"state": {

"age\_group\_proportions": [0.60, 0.3, 0.1],

"susceptible": [0.5, 0.6, 0.4, 0.3, 0.2],

"fatalities": [0, 0, 0, 0, 0],

"infected": [

[0.5, 0, 0, 0, 0],

[0.4, 0, 0, 0, 0],

[0.6, 0, 0, 0, 0],

],

"recovered": [

[0, 0, 0, 0, 0],

[0, 0, 0, 0, 0],

[0, 0, 0, 0, 0],

]

},

"neighborhood": [

{ "cell\_id": "Country1" },

{

"cell\_id": "Country2",

"vicinity": { "correlation": 0.2 }

},

{

"cell\_id": "Country3",

"vicinity": { "correlation": 0.3 }

}

]

},

{

"cell\_id": "Country2",

"neighborhood": [

{

"cell\_id": "Country1",

"vicinity": { "correlation": 0.2 }

},

{ "cell\_id": "Country2" },

{

"cell\_id": "Country3",

"vicinity": { "correlation": 0.4 }

}

]

},

{

"cell\_id": "Country3",

"neighborhood": [

{

"cell\_id": "Country1",

"vicinity": { "correlation": 0.3 }

},

{

"cell\_id": "Country2",

"vicinity": { "correlation": 0.4 }

},

{ "cell\_id": "Country3" }

]

}

]

}