

SYSC 5104 – Methodologies for Discrete Event Modelling and Simulation

Assignment 1

Cadmium SEIRD Neighborhoods

Aidan Fahlman

101212919

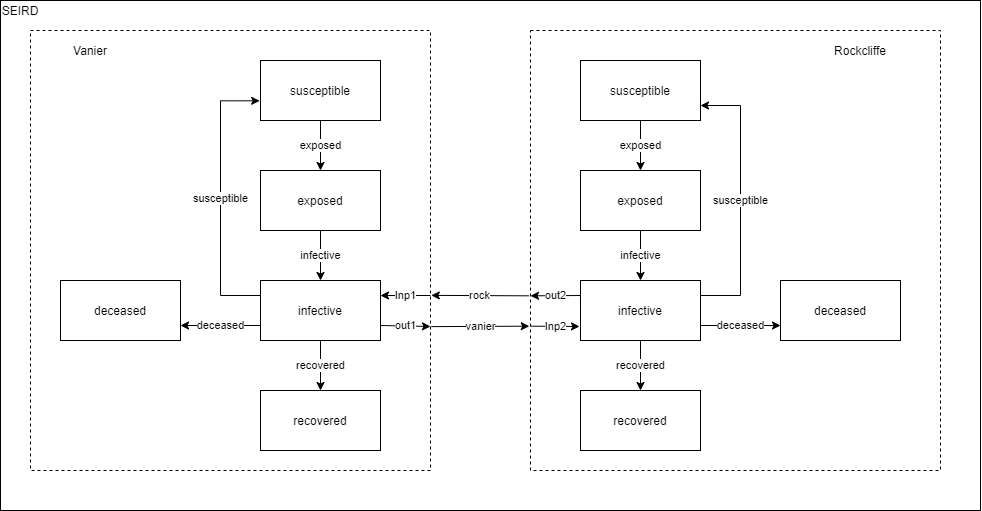
**Course Instructor:** Dr. Gabriel Wainer

October 25th, 2020

**The Problem:**

Typical SEIRD models are developed using the SEIRD format. SEIRD stands for susceptible, exposed, infective, recovered and deceased. We generally look at this model for a given neighborhood, for example the Vanier neighborhood in Ottawa. The problem with this is it does not account for people moving back and forth between multiple neighborhoods. People come and go to work, school and bars and many children with divorced families may have multiple homes. My model is going to focus on these cases where people may be moving from one neighborhood to another. To start my model will focus on first joining the two models to run side by side as well as output both neighborhoods information. Then, cases where children are infected and then move from one home to the other with their infection. My model will assume 0.5% of the infected population of either city are children moving from one neighborhood to the other. These children will be moved after each ta(). The goal will be to continue development on this model to include more in-depth transfers between the two neighborhoods. The figure below shows how my model will couple two SEIRD models together. A typical SEIRD model can be described by looking at either the Vanier, or Rockcliffe sub model below.

The Coupled SEIRD Model I am developing looks as follow:



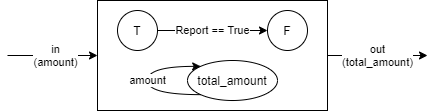
**Atomic Models**

***Accumulator =* <X, S, Y, δint, δext, λ, ta> (Recovered/Deceased)**

The accumulator atomic model will take in a float value as an input and add it to the total\_value associated with it. For example:

input = new\_deceased = 5,

total\_deceased = 10,

output = total\_deceased + new\_deceased = 15

X = {amount}

Y = {total\_amount}

S = {total\_amount, report ∈ (true, false)}

δint () = if (report == true) {report = false}

δext(S, e, X) = {

total\_amount = total\_amount + X.amount

report = true

}

λ(report == true) = {Y. total\_ amount = total\_ amount }

ta(report == true) = {0}

ta(report == false) = {inf}

***Infective* = <X, S, Y, δint, δext, λ, ta>**

The infective atomic model receives inputs from two sources, the in source takes in information from the exposed model. The infective\_in in port takes in new\_infective individuals from the other coupled model. The model outputs the new\_deceased individuals, the new\_recovered individuals as well as the infective\_neighbor value which is used as the input for the other coupled model. The model uses the calculations listed below to create the new\_deceased and new\_recovered numbers.

Model parameters: mortality, infective\_period, initial\_infective, dt

X = {infective, in\_infective}

Y = {total\_infective, deceased, recovered, out\_infective}

S = {total\_infective, new\_deceased, new\_recovered, infective\_neighbor, report ∈ (true, false)}

δint (report == true) = {report = false}

δext(S, e, X) = {

new\_deceased = ((total\_infective/infective\_period)\*(mortality/100))\*dt

new\_recovered = ((total\_infective/infective\_period)\*(1-(mortality/100)))\*dt

1. total\_infective = total\_infective + X.infective – new\_deceased – new\_recovered + new\_infective\_neighbor

infective\_neighbor = total\_infective \* 0.005

1. total\_infective = total\_infective – infective\_neighbor

report = true

}

λ(report == true) = {

Y. total\_infective = total\_infective

Y.deceased = new\_deceased

Y.recovered = new\_ recovered

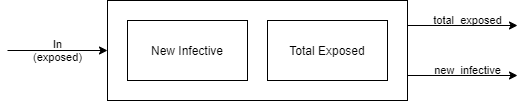
Y. out\_infective = infective\_neighbor

}

ta(report == true) = {0}

ta(report == false) = {inf}

***Exposed* = < X, S, Y, δint, δext, λ, ta>**

The exposed atomic model receives a single input from the susceptible model. The input holds a float value that represents the newly exposed individuals. The model outputs the total number of exposed individuals as well as the exposed individuals that became infective.

Model parameters: incubation\_period, dt

X = {exposed}

Y = {total\_exposed, new\_infective}

S = {total\_exposed, new\_infective, report ∈ (true, false)}

δint (report == true) = {report = false}

δext(S, e, X) = {

new\_infective = (total\_exposed/incubation\_period)\*dt

total\_exposed = total\_exposed + X.exposed - new\_infective

report = true

}

λ(report == true) ={

Y.total\_exposed = total\_exposed

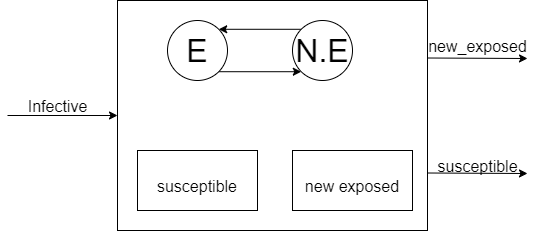
Y.new\_infective = new\_infective

}

ta(report == true) = {0}

ta(report == false) = {inf}

***Susceptible* = < X, S, Y, δint, δext, λ, ta>**

The susceptible atomic model receives input only when the SDL\_STATE is non empty, in other words once the infective model outputs new\_susceptible individuals they will be inputted into the model. The model calculates the number of new exposed to be outputted as well as the remaining susceptible individuals.

Model parameters: total\_population, transmission\_rate, initial\_infective, dt

X = {infective}

Y = {susceptible, new\_exposed}

S = {SDL\_STATE ∈ {EMPTY, NOEMPTY}, susceptible, new\_exposed, report ∈ (true, false)}

δint (report == true) = {report = false}

δext(S, e, X) = {

report = true

if(SDL\_STATE == NOEMPTY)

new\_exposed = min (susceptible, ((transmission\_rate\*susceptible\*X.infective)/total\_population)\*dt)

susceptible = susceptible – new\_exposed

If (susceptible >0)

SDL\_STATE = NOEMPTY

else

SDL\_STATE = EMPTY

Else

new\_exposed = 0

}

λ(report == true)= {

Y.susceptible = susceptible

Y.new\_exposed = new\_exposed

}

ta(report == true) ={dt}

ta(report == false) = {inf}

**Coupled Models**

***VANIER* = <X, Y, D, EIC, EOC, IC>**

X = {inp1}

Y = {out1}

D = {susceptible, exposed, infective, recovered, deceased}

EIC = {inp1, “infective”}

EOC = {“infective”, out\_infective}

IC = {(susceptible, “exposed”), (exposed, “infective”), (infective, “deceased”), (infective, “recovered”), (infective, “susceptible”), (infective, “infective”)

***ROCKCLIFFE* = <X, Y, D, EIC, EOC, IC>**

X = {inp2}

Y = {out2}

D = {susceptible, exposed, infective, recovered, deceased}

EIC = {inp1, “infective”}

EOC = {“infective”, out\_infective}

IC = {(susceptible, “exposed”), (exposed, “infective”), (infective, “deceased”), (infective, “recovered”), (infective, “susceptible”), (infective, “infective”)

***TOP MODEL: SEIRD* = <X, Y, D, EIC, EOC, IC>**

X = { }

Y = { }

D = { VANIER, ROCKCLIFFE }

EIC = { }

EOC = { }

IC = {(outp1, “ROCKCLIFFE”), (outp2, “VANIER”) }

**Testing Strategies:**

The model will be tested using the black box testing method. Testing will be conducted by passing defined input data through the model and checking whether the outputs are as we expect.

**Testing**

**Data longevity:**

The first test I conducted was a basic longevity test, to see if I was losing too much data due to inaccuracies in rounding. This was done by taking my input data at the beginning of the model and comparing it with the outputs at the end of the model. We can see below the input data looked as follows:

|  |  |  |
| --- | --- | --- |
| Rockcliffe | Vanier | Total |
| TIME = 0 | | |
| total population = 1922  initial infective = 10  sum = 1932 | total population = 16509  initial infective = 100  sum = 16609 | total population = 18431  initial infective = 110  sum = **18541** |
| TIME = 100 | | |
| total infective = 22.26  total recovered = 5889.05  total deceased = 309.95 | total infective = 22.28  total recovered = 10390.90  total deceased = 1906.02 | total infective = 44.54  total recovered = 16279.95  total deceased = 2215.97  sum = **18540.46**  **diff = -.56** |

With this test we can see we lose approximately .56 entities throughout the model due to in accurate rounding. But we can also see that out entities being passed from one coupled model to the other are not being lost though the model.

**Deceased + Recovered testing:**

To test how the deceased and recovered models are works in the scope of the whole model we will look at how the accumulator adds them to the new states. Lets look at TIME 0.1 and 0.2 for Vanier below:

**0.1:**

[susceptible\_defs::report: {16509}, susceptible\_defs::new\_exposed: {0}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {100}, infective\_defs::total\_infective: {100}, **infective\_defs::deceased: {0}**, **infective\_defs::recovered: {0}**, infective\_defs::out\_infective: {0}] generated by model infective

**0.2:**

[susceptible\_defs::report: {16489.1}, susceptible\_defs::new\_exposed: {19.8796}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {98.7893}, infective\_defs::total\_infective: {98.7893}, **infective\_defs::deceased: {0.110714}**, **infective\_defs::recovered: {0.603571}**, infective\_defs::out\_infective: {0.496429}] generated by model infective

Now assuming our model is properly passing the data through the accumulator and calculating our new recovered and deceased counts we should get the following:

Recovered = 0 + 0.603571 = 0.603571

Deceased = 0 + 0.110714 = 0.110714

We can see out accumulator outputs has to correct outputs below:

[accumulator\_defs::report: {0.603571}] generated by model recovered

[accumulator\_defs::report: {0.110714}] generated by model deceased

**Infective:**

Next we are going to look at the Infective model. Lets start by looking at how this model calculates new recovered and new deceased. Lets look at TIME 0.2 again below:

[susceptible\_defs::report: {1920.01}, susceptible\_defs::new\_exposed: {1.98965}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {9.87893}, infective\_defs::total\_infective: {9.87893}, infective\_defs::deceased: {0.00357143}, infective\_defs::recovered: {0.0678571}, infective\_defs::out\_infective: {0.0496429}] generated by model infective

So if our model correctly calculated the new deceased and recovered numbers we should see the following in the next time iteration:

**Deceased:** [(9.87893/14)\*(5/100)]\*(0.1) = 0.003528189

**Recovered:** [(9.87893/14)\*(1- (5/100))]\*(0.1) = 0.067035596

Below is the next time iteration showing these calculations are in fact correct:

[susceptible\_defs::report: {1918.05}, susceptible\_defs::new\_exposed: {1.96352}] generated by model susceptible

[exposed\_defs::report: {1.98965}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {10.2533}, infective\_defs::total\_infective: {10.2533}, infective\_defs::deceased: {**0.00352819**}, infective\_defs::recovered: {**0.0670356**}, infective\_defs::out\_infective: {0.051524}]

Next in infective is the calculation of the new infective counts. This calculations takes more inputs than the previous, lets look at how ***Rockcliffe*** 0.3 is calculated by looking at 0.2 below:

**Rockcliffe 2.0**:

[susceptible\_defs::report: {1920.01}, susceptible\_defs::new\_exposed: {1.98965}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {**9.87893**}, infective\_defs::total\_infective: {9.87893}, infective\_defs::deceased: {0.00357143}, infective\_defs::recovered: {0.0678571}, infective\_defs::out\_infective: {0.0496429}] generated by model infective

**Vanier 2.0**:

[susceptible\_defs::report: {16489.1}, susceptible\_defs::new\_exposed: {19.8796}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {98.7893}, infective\_defs::total\_infective: {98.7893}, infective\_defs::deceased: {0.110714}, infective\_defs::recovered: {0.603571}, infective\_defs::out\_infective: {**0.496429**}] generated by model infective

So, we are looking for the *Rockcliffe* 0.3 new\_infective numbers. To find this we need to first calculate the number of new deceased and new recovered from 0.3, lets look below:

**Rockcliffe 3.0:**

[susceptible\_defs::report: {1918.05}, susceptible\_defs::new\_exposed: {1.96352}] generated by model susceptible

[exposed\_defs::report: {1.98965}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {**10.2533**}, infective\_defs::total\_infective: {10.2533}, infective\_defs::deceased: {**0.00352819**}, infective\_defs::recovered: {**0.0670356**}, infective\_defs::out\_infective: {**0.051524**}] generated by model infective

So, lets calculate this. In Rockcliffe 2.0 we need to get the Infective count 9.87893, then we need to remove the new deceased, recovered and out\_infective from 3.0:

9.87893 – 0.00352819 – 0.0670356 – 0.051524 = 9.75684221

With this, we need to add the out\_infective neighbors coming from 2.0 vanier:

9.75684221 + 0.496429 = **10.25327**

We can see by reviewing the Rockcliffe 3.0 above, this calculation was correct.

**Exposed:**

The next model to investigate is the exposed model. This model calculates the number of new infective individuals moving from exposed to infective. For this we will look at the number of exposed individuals in time 0.3 and then the number of new\_infective individuals in 0.4. If the math adds up in this we know out model is working. Below we have *Rockcliffe* 0.3:

[susceptible\_defs::report: {1918.05}, susceptible\_defs::new\_exposed: {1.96352}] generated by model susceptible

[exposed\_defs::report: {1.98965}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {10.2533}, infective\_defs::total\_infective: {10.2533}, infective\_defs::deceased: {0.00352819}, infective\_defs::recovered: {0.0670356}, infective\_defs::out\_infective: {0.051524}] generated by model infective

So, if the model is working correctly, we will take the number of current exposed (1.98965) and pass it through the model. The output should be:

(1.98965/5)\*0.1 = **0.039793**

Our output for *Rockcliffe* 0.4 below shows this is correct:

[susceptible\_defs::report: {1916.01}, susceptible\_defs::new\_exposed: {2.03584}] generated by model susceptible

[exposed\_defs::report: {3.91338}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {**0.039793**}] generated by model exposed

[infective\_defs::report: {10.6173}, infective\_defs::total\_infective: {10.6173}, infective\_defs::deceased: {0.00366188}, infective\_defs::recovered: {0.0695758}, infective\_defs::out\_infective: {0.0533535}] generated by model infective

**Susceptible:**

Out last model to check is the susceptible model. Lets look at how the susceptible model calculates the new exposed population from TIME 0.2 to 0.3. *Rockcliffe* 0.2 is shown below

[susceptible\_defs::report: {1920.01}, susceptible\_defs::new\_exposed: {1.98965}] generated by model susceptible

[exposed\_defs::report: {0}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {9.87893}, infective\_defs::total\_infective: {9.87893}, infective\_defs::deceased: {0.00357143}, infective\_defs::recovered: {0.0678571}, infective\_defs::out\_infective: {0.0496429}] generated by model infective

Out calculation should show the following:

((2\*1920.01\*9.87893)/1932)\*0.1 = **1.963524264**

Below, we see *Rockcliffe* 0.3 showing the correct output.

[susceptible\_defs::report: {1918.05}, susceptible\_defs::new\_exposed: {**1.96352**}] generated by model susceptible

[exposed\_defs::report: {1.98965}, exposed\_defs::total\_exposed: {}, exposed\_defs::new\_infective: {0}] generated by model exposed

[infective\_defs::report: {10.2533}, infective\_defs::total\_infective: {10.2533}, infective\_defs::deceased: {0.00352819}, infective\_defs::recovered: {0.0670356}, infective\_defs::out\_infective: {0.051524}] generated by model infective