Scaling of epidemiology models with multi-site compartments

Version 0.9

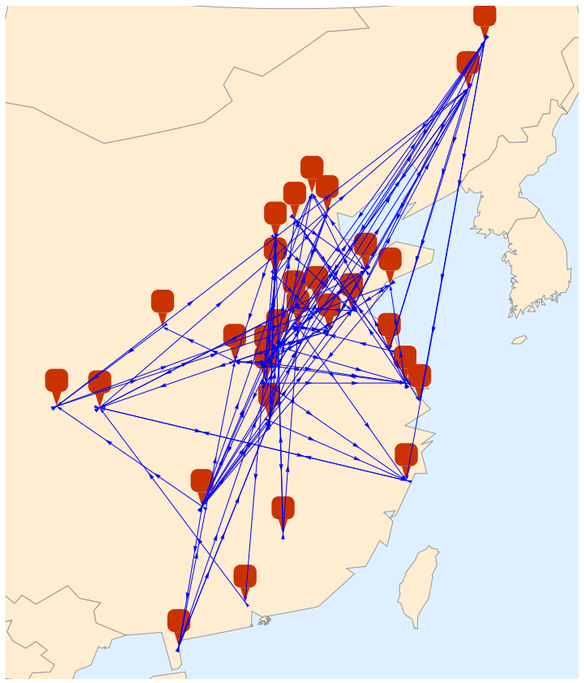
Anton Antonov
MathematicaForPrediction at WordPress
SystemModeling at GitHub
March 2020

Introduction

In this notebook we describe and exemplify an algorithm that allows the specification and execution geo-spatial-temporal simulations of infectious disease spreads. (Concrete implementations and examples are given.)

The assumptions of the typical compartmental epidemiological models do not apply for countries or cities that are non-uniformly populated. (For example, China, USA, Russia.) There is a need to derive epidemiological models that take into account the non-uniform distribution of populations and related traveling patterns within the area of interest.

Here is a visual aid (made with a random graph over the 30 largest cities of China):



In this notebook we show how to extend core, single-site epidemiological models into larger models for making spatial-temporal simulations. In the explanations and examples we use SEI2R, [AA2, AAp1], as a core epidemiological model, but other models can be adopted if they adhere to the model data structure of the package "EpidemiologyModels.m", [AAp1].

Definitions

Single site: A geographical location (city, neighbourhood, campus) for which the assumptions of the classical compartmental epidemiological models hold.

Single site epidemiological model: A compartmental epidemiological model for a single site. Such model has a system of Ordinary Differential Equations (ODE's) and site dependent initial conditions.

Multi-site area: An area comprised of multiple single sites with known traveling patterns between them. The area has a directed graph *G* with nodes that correspond to the sites and a positive matrix *tpm*(*G*) for the traveling patterns between the sites.

Problem definition: Given (i) a single site epidemiological model *M*, (ii) a graph *G* connecting multiple sites, and (iii) a traveling patterns matrix *tpm*(*G*) between the nodes of *G* derive an epidemiological model *S*(*M*, *tpm*(*G*)) that simulates more adequately viral decease propagation over *G*.

Multi-Site Epidemiological Model Extension Algorithm (MSEMEA): An algorithm that derives from a given single site epidemiological model and multi-site area an epidemiological model that can be used to simulate the geo-spatial-temporal epidemics and infectious disease spreads. (The description of MSEMEA is the main message of this notebook.)

Load packages

The epidemiological models framework used in this notebook is implemented with the packages [AAp1, AAp2, AA3]; the interactive plots functions are from the package [AAp4].

In[4]:=
```
Import["https://raw.githubusercontent.com/antononcube/SystemModeling/master/Projects/Coronavirus-propagation-dynamics/WL/EpidemiologyModelModifications.m"]
Import["https://raw.githubusercontent.com/antononcube/SystemModeling/master/Projects/Coronavirus-propagation-dynamics/WL/EpidemiologyModelingVisualizationFunctions.m"]
Import["https://raw.githubusercontent.com/antononcube/SystemModeling/master/WL/SystemDynamicsInteractiveInterfacesFunctions.m"]
```

Notebook structure

The section "General algorithm description" gives rationale and conceptual steps of MSEMEA.

The next two sections of the notebook follow the procedure outline using the SEI2R model as *M*, a simple graph with two nodes as *G*, and both constant and time-dependent matrices for *tpm*(*G*).

The next two sections provide examples with larger graphs: one is over a grid graph, the other is over a random graph.

The section "Money from lost productivity" shows how to track the money losses across the sites.

The last section "Future plans" outlines envisioned (immediate) extensions work presented in this notebook.

General algorithm description

Splitting and scaling

The traveling between large, densely populated cities is a very different process of the usual mingling in those cities. The usual large, dense city mingling is assumed and used in the usual epidemiological models. It seems it is a good idea to split the two processes and derive a common model.

Assume that all journeys finish within a day. We can model the people arriving (flying in) into a city as births, and people departing a city as deaths.

Let as take a simple model like SIR or SEIR and write the equation for every site we consider. This means for every site we have the same ODE's with site-dependent initial conditions.

Consider the traveling patterns matrix *K*, which is a contingency matrix derived from source-destination traveling records. The matrix entry of $K(i, j)$ tells us how many people traveled from site *i* to site *j*. We systematically change the ODE's of the sites in following way.

Assume that site *a* had only travelers coming from site *b* and going to site *b*. Assume that the Total Population (TP) sizes for sites *a* and *b* are N_a and N_b respectively. Assume that only people from the Susceptible Population (SP) traveled. If the adopted single-site model is SIR, [Wk1], we change the SP equation of site *a*

$$\mathrm{SP}_a'(t) = -\frac{\beta\, \mathrm{IP}_a(t)\, \mathrm{SP}_a(t)}{N_a} - \mathrm{SP}_a(t)\, \mu \tag{1}$$

to become

$$\mathrm{SP}_a{}'(t) = -\frac{\beta\, \mathrm{IP}_a(t)\, \mathrm{SP}_a(t)}{N_a} - \mathrm{SP}_a(t)\, \mu - \frac{K(a,b)\, \mathrm{SP}_a(t)}{N_a} + \frac{K(b,a)\, \mathrm{SP}_b(t)}{N_b}, \tag{2}$$

assuming that

$$\begin{aligned} &\frac{K(a,b)\, \mathrm{SP}_a(t)}{N_a} \le N_a\,, \\ &\frac{K(b,a)\, \mathrm{SP}_b(t)}{N_b} \le N_b. \end{aligned} \tag{3}$$

Remark: In the package [AAp3] the transformations above are done with the more general and robust formula:

$$\min\left(\frac{K(i,j)\, \mathrm{SP}_i(t)}{\mathrm{TP}_i(t)}, \mathrm{TP}_i(t)\right). \tag{4}$$

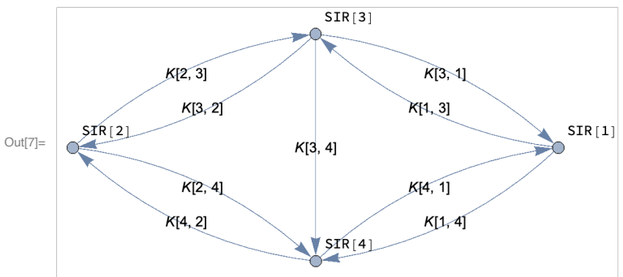
The transformed system ODE's of the sites are joined into one "big" system ODE's, appropriate initial conditions are set, and the "big" ODE system is solved. (The sections below show concrete examples.)

Steps of MSEMEA

Here is a visual aid:

In[7]:= CompoundExpression[...]

Out[7]=



1. Get a single-site epidemiological compartmental model data structure, *M*.
   - 1.1. The model data structure has stocks and rates dictionaries, equations, initial conditions, and prescribed rate values; see [AA2, AAp1].
2. Derive the site-to-site traveling patterns matrix *K* for the sites in the graph *G*.
3. For each of node *i* of *G* make a copy of the model *M* and denote with *M*[*i*].
   - 3.1. In general, the models *M*[*i*], $i \in G$ have different initial conditions.
   - 3.2. The models can also have different death rates, contact rates, etc.
4. Combine the models *M*[*i*], $i \in G$ into the scaled model *S*.
   - 4.1. Change the equations of *M*[*i*], $i \in G$ to reflect the traveling patterns matrix *K*.
   - 4.2. Join the systems of ODE's of *M*[*i*], $i \in G$ into one system of ODE's.
5. Set appropriate or desired initial conditions for each of the populations in *S*.
6. Solve the ODE's of *S*.
7. Visualize the results.

Precaution

Care should be taken when specifying the initial conditions of MSEMEA's system of ODE's (sites' populations) and the traveling patterns matrix. For example, the simulations can "blow up" if the traveling patterns matrix values are too large. As it was indicated above, the package [AAp3] puts some safe-guards, but in our experiments with random graphs and random traveling patterns matrices occasionally we still get "wild" results.

Analogy with Large scale air-pollution modeling

There is strong analogy between be MSEMEA and Eulerian models of Large Scale Air-Pollution Modeling (LSAPM), [AA3, ZZ1].

The mathematical models of LSAPM have a "chemistry part" and an "advection-diffusion part." It is hard to treat such mathematical model directly -- different kinds of splitting are used. If we consider 2D LSAPM then we can say that we cover the modeling area with steer tank reactors, then with the chemistry component we simulate the species chemical reactions in those steer tanks, and with the advection-diffusion component we change species concentrations in the steer tanks (according to some wind patterns.)

Similarly, with MSEMEA we separated the travel of population compartments from the "standard" epidemiological modeling interaction between the population compartments.

Similarly to the so called "rotational test" used in LSAPM to evaluate numerical schemes, we derive and study the results of "grid graph tests" for MSEMEA.

Single site epidemiological model

Here is the SEI2R model from the package [AAp1]:

In[8]:=
```
model1 = SEI2RModel[t, "InitialConditions" → True, "RateRules" → True, "TotalPopulationRepresentation" → "AlgebraicEquation"];
ModelGridTableForm[model1]
```

Out[9]=

⟨| Stocks →

#	Symbol	Description
1	TP[t]	Total Population
2	SP[t]	Susceptible Population
3	EP[t]	Exposed Population
4	INSP[t]	Infected Normally Symptomatic Population
5	ISSP[t]	Infected Severely Symptomatic Population
6	RP[t]	Recovered Population
7	MLP[t]	Money of lost productivity

, Rates →

#	Symbol	Description
1	μ[TP]	Population death rate
2	μ[INSP]	Infected Normally Symptomatic Population death rate
3	μ[ISSP]	Infected Severely Symptomatic Population death rate
4	sspf[SP]	Severely Symptomatic Population Fraction
5	β[INSP]	Contact rate for the normally symptomatic population
6	β[ISSP]	Contact rate for the severely symptomatic population
7	aip	Average infectious period
8	aincp	Average incubation period
9	lpcr[ISSP, INSP]	Lost productivity cost rate (per person per day)

,

Equations →

#	Equation
1	$SP'[t] == -\frac{INSP[t]\times SP[t]\times\beta[INSP]}{TP[t]} - \frac{ISSP[t]\times SP[t]\times\beta[ISSP]}{TP[t]} - SP[t]\times\mu[TP]$
2	$EP'[t] == \frac{INSP[t]\times SP[t]\times\beta[INSP]}{TP[t]} + \frac{ISSP[t]\times SP[t]\times\beta[ISSP]}{TP[t]} - EP[t]\left(\frac{1}{aincp} + \mu[TP]\right)$
3	$INSP'[t] == -\frac{INSP[t]}{aip} + \frac{EP[t]\,(1-sspf[SP])}{aincp} - INSP[t]\times\mu[INSP]$
4	$ISSP'[t] == -\frac{ISSP[t]}{aip} + \frac{EP[t]\times sspf[SP]}{aincp} - ISSP[t]\times\mu[ISSP]$
5	$RP'[t] == \frac{INSP[t]+ISSP[t]}{aip} - RP[t]\times\mu[TP]$
6	$MLP'[t] == lpcr[ISSP, INSP]\,(-RP[t] - SP[t] + TP[t])$
7	$TP[t] == Max[0, EP[t] + INSP[t] + ISSP[t] + RP[t] + SP[t]]$

, RateRules →

#	Symbol	Value
1	μ[TP]	$\frac{1}{45\,625}$
2	μ[ISSP]	$\frac{0.035}{aip}$
3	μ[INSP]	$\frac{0.01}{aip}$
4	β[ISSP]	6
5	β[INSP]	3
6	aip	28
7	aincp	6
8	sspf[SP]	0.2
9	lpcr[ISSP, INSP]	600

, InitialConditions →

#	Equation
1	SP[0] == 99 998
2	EP[0] == 0
3	ISSP[0] == 1
4	INSP[0] == 1
5	RP[0] == 0
6	MLP[0] == 0
7	TP[0] == 100 000

|⟩

Here we endow the SEI2R model with a (prominent) ID:

In[10]:= `ModelGridTableForm[AddModelIdentifier[model1, Style["myX", Red]]]`

Out[10]= ⟨| Stocks →

#	Symbol	Description
1	TP[myX][t]	Total Population
2	SP[myX][t]	Susceptible Population
3	EP[myX][t]	Exposed Population
4	INSP[myX][t]	Infected Normally Symptomatic Population
5	ISSP[myX][t]	Infected Severely Symptomatic Population
6	RP[myX][t]	Recovered Population
7	MLP[myX][t]	Money of lost productivity

, Rates →

#	Symbol	Description
1	μ[myX][TP]	Population death rate
2	μ[myX][INSP]	Infected Normally Symptomatic Population death rate
3	μ[myX][ISSP]	Infected Severely Symptomatic Population death rate
4	sspf[myX][SP]	Severely Symptomatic Population Fraction
5	β[myX][INSP]	Contact rate for the normally symptomatic population
6	β[myX][ISSP]	Contact rate for the severely symptomatic population
7	aip[myX]	Average infectious period
8	aincp[myX]	Average incubation period
9	lpcr[myX][ISSP, INSP]	Lost productivity cost rate (per person per day)

,

Equations →

#	Equation
1	$SP[myX]'[t] == -\frac{INSP[myX][t]\times SP[myX][t]\times\beta[myX][INSP]}{TP[myX][t]} - \frac{ISSP[myX][t]\times SP[myX][t]\times\beta[myX][ISSP]}{TP[myX][t]} - SP[myX][t]\times\mu[myX][TP]$
2	$EP[myX]'[t] == \frac{INSP[myX][t]\times SP[myX][t]\times\beta[myX][INSP]}{TP[myX][t]} + \frac{ISSP[myX][t]\times SP[myX][t]\times\beta[myX][ISSP]}{TP[myX][t]} - EP[myX][t]\left(\frac{1}{aincp[myX]} + \mu[myX][TP]\right)$
3	$INSP[myX]'[t] == -\frac{INSP[myX][t]}{aip[myX]} + \frac{EP[myX][t]\,(1-sspf[myX][SP])}{aincp[myX]} - INSP[myX][t]\times\mu[myX][INSP]$
4	$ISSP[myX]'[t] == -\frac{ISSP[myX][t]}{aip[myX]} + \frac{EP[myX][t]\times sspf[myX][SP]}{aincp[myX]} - ISSP[myX][t]\times\mu[myX][ISSP]$
5	$RP[myX]'[t] == \frac{INSP[myX][t]+ISSP[myX][t]}{aip[myX]} - RP[myX][t]\times\mu[myX][TP]$
6	$MLP[myX]'[t] == lpcr[myX][ISSP, INSP]\,(-RP[myX][t] - SP[myX][t] + TP[myX][t])$
7	$TP[myX][t] == Max[0, EP[myX][t] + INSP[myX][t] + ISSP[myX][t] + RP[myX][t] + SP[myX][t]]$

,

RateRules →

#	Symbol	Value
1	μ[myX][TP]	$\frac{1}{45\,625}$
2	μ[myX][ISSP]	$\frac{0.035}{aip[myX]}$
3	μ[myX][INSP]	$\frac{0.01}{aip[myX]}$
4	β[myX][ISSP]	6
5	β[myX][INSP]	3
6	aip[myX]	28
7	aincp[myX]	6
8	sspf[myX][SP]	0.2
9	lpcr[myX][ISSP, INSP]	600

, InitialConditions →

#	Equation
1	SP[myX][0] == 99 998
2	EP[myX][0] == 0
3	ISSP[myX][0] == 1
4	INSP[myX][0] == 1
5	RP[myX][0] == 0
6	MLP[myX][0] == 0
7	TP[myX][0] == 100 000

|⟩

Thus we demonstrated that we can do Step 3 of MSEMEA.

Below we use ID's that correspond to the nodes of graphs (and are integers.)

Scaling the single-site SIR model over a small complete graph

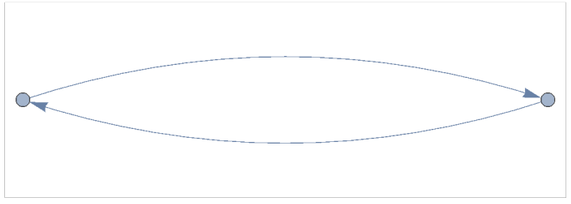
Constant travel matrices

Assume we have two sites and the following graph and matrix describe the traveling patterns between them.

Here is the graph:

In[11]:=
```
gr = CompleteGraph[2, DirectedEdges → True, GraphLayout → "SpringElectricalEmbedding"]
```

Out[11]=



And here is the traveling patterns matrix:

In[12]:=
```
SeedRandom[44];
matTravel = AdjacencyMatrix[gr] * RandomInteger[{100, 1000}, {VertexCount[gr], VertexCount[gr]}];
MatrixForm[matTravel]
```

Out[14]//MatrixForm=

$$\begin{pmatrix} 0 & 623 \\ 224 & 0 \end{pmatrix}$$

Note that there are much more travelers from 1 to 2 than from 2 to 1.

Here we obtain the core, single-site model (as shown in the section above):

In[15]:=
```
model1 = SEI2RModel[t, "InitialConditions" → True, "RateRules" → True, "TotalPopulationRepresentation" → "AlgebraicEquation"];
```

Make the multi-site compartments model with SEI2R and the two-node travel matrix using the function `ToSiteCompartmentsModel` of [AAp2]:

In[16]:=
```
modelBig =
  ToSiteCompartmentsModel[model1, matTravel, "MigratingPopulations" → {"Susceptible Population", "Exposed Population", "Infected Normally Symptomatic Population", "Recovered Population"}];
```

Show the unique stocks in the multi-site model:

In[17]:=
```
GetPopulationSymbols[modelBig, __ ~~ __]
```

Out[17]=
```
{TP[1], SP[1], EP[1], INSP[1], ISSP[1], RP[1], MLP[1], TP[2], SP[2], EP[2], INSP[2], ISSP[2], RP[2], MLP[2]}
```

From the symbolic form of the multi-model equations derive the specific equations with the adopted rate values:

In[18]:=
```
ModelGridTableForm[KeyTake[modelBig, {"Equations"}] //. modelBig["RateRules"]]
```

Out[18]=

#	Equation
1	$SP[1]'[t] == -\text{Min}\left[\frac{623\,SP[1][t]}{TP[1][t]}, TP[1][t]\right] + \text{Min}\left[\frac{224\,SP[2][t]}{TP[2][t]}, TP[2][t]\right] - \frac{SP[1][t]}{45\,625} - \frac{3\,INSP[1][t]\times SP[1][t]}{TP[1][t]} - \frac{6\,ISSP[1][t]\times SP[1][t]}{TP[1][t]}$
2	$EP[1]'[t] == -\text{Min}\left[\frac{623\,EP[1][t]}{TP[1][t]}, TP[1][t]\right] + \text{Min}\left[\frac{224\,EP[2][t]}{TP[2][t]}, TP[2][t]\right] - \frac{45\,631\,EP[1][t]}{273\,750} + \frac{3\,INSP[1][t]\times SP[1][t]}{TP[1][t]} + \frac{6\,ISSP[1][t]\times SP[1][t]}{TP[1][t]}$
3	$INSP[1]'[t] == -\text{Min}\left[\frac{623\,INSP[1][t]}{TP[1][t]}, TP[1][t]\right] + \text{Min}\left[\frac{224\,INSP[2][t]}{TP[2][t]}, TP[2][t]\right] + 0.133333\,EP[1][t] - 0.0360714\,INSP[1][t]$
4	$ISSP[1]'[t] == 0.0333333\,EP[1][t] - 0.0369643\,ISSP[1][t]$
5	$RP[1]'[t] == -\text{Min}\left[\frac{623\,RP[1][t]}{TP[1][t]}, TP[1][t]\right] + \text{Min}\left[\frac{224\,RP[2][t]}{TP[2][t]}, TP[2][t]\right] + \frac{1}{28}(INSP[1][t] + ISSP[1][t]) - \frac{RP[1][t]}{45\,625}$
6	$MLP[1]'[t] == 600\,(-RP[1][t] - SP[1][t] + TP[1][t])$
7	$TP[1][t] == \text{Max}[0, EP[1][t] + INSP[1][t] + ISSP[1][t] + RP[1][t] + SP[1][t]]$
8	$SP[2]'[t] == \text{Min}\left[\frac{623\,SP[1][t]}{TP[1][t]}, TP[1][t]\right] - \text{Min}\left[\frac{224\,SP[2][t]}{TP[2][t]}, TP[2][t]\right] - \frac{SP[2][t]}{45\,625} - \frac{3\,INSP[2][t]\times SP[2][t]}{TP[2][t]} - \frac{6\,ISSP[2][t]\times SP[2][t]}{TP[2][t]}$
9	$EP[2]'[t] == \text{Min}\left[\frac{623\,EP[1][t]}{TP[1][t]}, TP[1][t]\right] - \text{Min}\left[\frac{224\,EP[2][t]}{TP[2][t]}, TP[2][t]\right] - \frac{45\,631\,EP[2][t]}{273\,750} + \frac{3\,INSP[2][t]\times SP[2][t]}{TP[2][t]} + \frac{6\,ISSP[2][t]\times SP[2][t]}{TP[2][t]}$
10	$INSP[2]'[t] == \text{Min}\left[\frac{623\,INSP[1][t]}{TP[1][t]}, TP[1][t]\right] - \text{Min}\left[\frac{224\,INSP[2][t]}{TP[2][t]}, TP[2][t]\right] + 0.133333\,EP[2][t] - 0.0360714\,INSP[2][t]$
11	$ISSP[2]'[t] == 0.0333333\,EP[2][t] - 0.0369643\,ISSP[2][t]$
12	$RP[2]'[t] == \text{Min}\left[\frac{623\,RP[1][t]}{TP[1][t]}, TP[1][t]\right] - \text{Min}\left[\frac{224\,RP[2][t]}{TP[2][t]}, TP[2][t]\right] + \frac{1}{28}(INSP[2][t] + ISSP[2][t]) - \frac{RP[2][t]}{45\,625}$
13	$MLP[2]'[t] == 600\,(-RP[2][t] - SP[2][t] + TP[2][t])$
14	$TP[2][t] == \text{Max}[0, EP[2][t] + INSP[2][t] + ISSP[2][t] + RP[2][t] + SP[2][t]]$

Show the initial conditions:

In[19]:=
```
RandomSample[modelBig["InitialConditions"], UpTo[12]]
```

Out[19]=
$$\{ISSP[2][0] == 1,\ TP[1][0] == 100\,000,\ EP[2][0] == 0,\ EP[1][0] == 0,\ SP[1][0] == 99\,998,\ RP[2][0] == 0,\ RP[1][0] == 0,\ INSP[1][0] == 1,\ INSP[2][0] == 1,\ TP[2][0] == 100\,000,\ SP[2][0] == 99\,998,\ MLP[1][0] == 0\}$$

Show the total number of equations:

In[20]:=
```
Length[modelBig["Equations"]]
```

Out[20]=
```
14
```

Solve the system of ODE’s of the extended model:

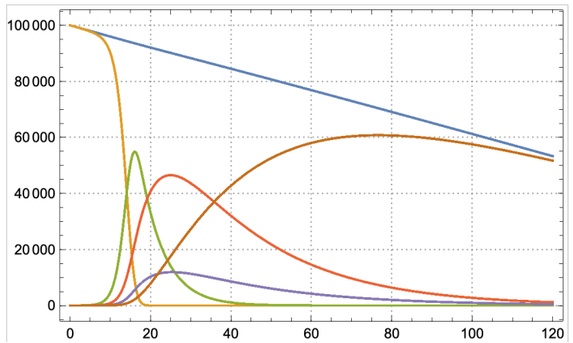
In[21]:=
```
maxTime = 120;
AbsoluteTiming[
  aSol = Association@First@
      NDSolve[
       Join[modelBig["Equations"] //. modelBig["RateRules"], modelBig["InitialConditions"]],
       GetStockSymbols[modelBig, __ ~~ "Population"],
       {t, 0, maxTime}
      ];
 ];
Length[aSol]
```

Out[23]=
```
12
```

Display the solutions for each site separately:

In[24]:=
```
ParametricSolutionsPlots[modelBig["Stocks"], #, None, maxTime, "Together" → True, PlotTheme → "Detailed", ImageSize → Medium] & /@ GroupBy[Normal@aSol, #[[1, 1]] &, Association]
```

Out[24]=



TP(1)(t), Total Population
SP(1)(t), Susceptible Population
EP(1)(t), Exposed Population
INSP(1)(t), Infected Normally Symptomatic Population
ISSP(1)(t), Infected Severely Symptomatic Population
RP(1)(t), Recovered Population

TP(2)(t), Total Population
SP(2)(t), Susceptible Population
EP(2)(t), Exposed Population
INSP(2)(t), Infected Normally Symptomatic Population
ISSP(2)(t), Infected Severely Symptomatic Population
RP(2)(t), Recovered Population

From the plots above we see that both sites start with total populations of 100 000 people. Because more travelers go from 1 to 2 we see that the exposed, infected, and recovered populations are larger at 2.

Time dependent travel matrices

Instead of using constant traveling patterns matrices we can use matrices with time functions as entries. It is instructive to repeat the computations above using this matrix:

In[25]:=
```
SeedRandom[232]
matTravel2 = matTravel * Table[Abs[Sin[RandomReal[{0.01, 0.1}] t]], VertexCount[gr], VertexCount[gr]];
MatrixForm[matTravel2]
```

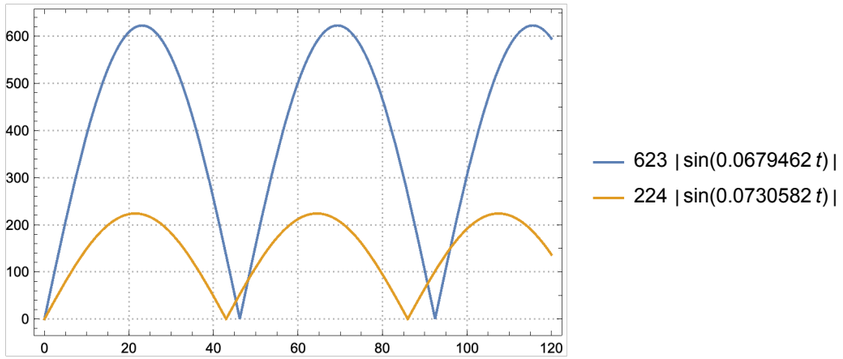
Out[27]//MatrixForm=

$$\begin{pmatrix} 0 & 623\,\text{Abs}[\text{Sin}[0.0679462\,t]] \\ 224\,\text{Abs}[\text{Sin}[0.0730582\,t]] & 0 \end{pmatrix}$$

Here are the corresponding number of traveling people functions:

In[28]:=
```
Plot[Evaluate[DeleteCases[Flatten@Normal@matTravel2, 0]], {t, 0, 120}, PlotTheme → "Detailed"]
```

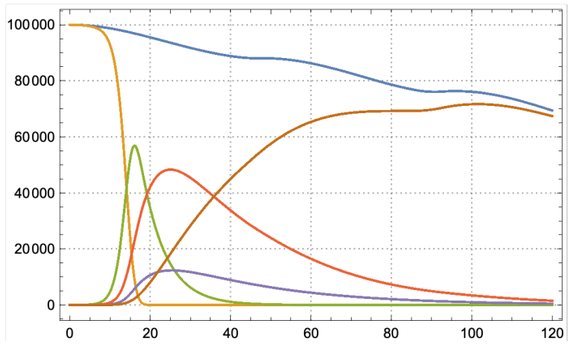
Out[28]=



Here we scale the SIR model, solve the obtained system of ODE's, and plot the solutions:

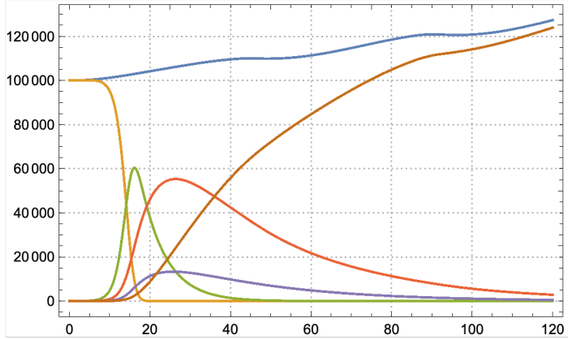
In[29]:=
```
modelBig = ToSiteCompartmentsModel[model1, matTravel2,
   "MigratingPopulations" → {"Susceptible Population", "Exposed Population", "Infected Normally Symptomatic Population", "Recovered Population"}];
aSol = Association@First@
    NDSolve[
     Join[modelBig["Equations"] //. modelBig["RateRules"], modelBig["InitialConditions"]],
     GetStockSymbols[modelBig, __ ~~ ""],
     {t, 0, maxTime}
    ];
ParametricSolutionsPlots[modelBig["Stocks"], #, None, 120, "Together" → True, PlotTheme → "Detailed", ImageSize → Medium][[1]] & /@
 GroupBy[Normal@KeySelect[aSol, ! MemberQ[{MLP}, Head[#]] &], #[[1, 1]] &, Association]
```

Out[31]=



TP(1)(t), Total Population
SP(1)(t), Susceptible Population
EP(1)(t), Exposed Population
INSP(1)(t), Infected Normally Symptomatic Population
ISSP(1)(t), Infected Severely Symptomatic Population
RP(1)(t), Recovered Population

2 →



TP(2)(t), Total Population
SP(2)(t), Susceptible Population
EP(2)(t), Exposed Population
INSP(2)(t), Infected Normally Symptomatic Population
ISSP(2)(t), Infected Severely Symptomatic Population
RP(2)(t), Recovered Population

Note that the oscillatory nature of the temporal functions in the travelling patterns matrix are reflected in the simulation results.

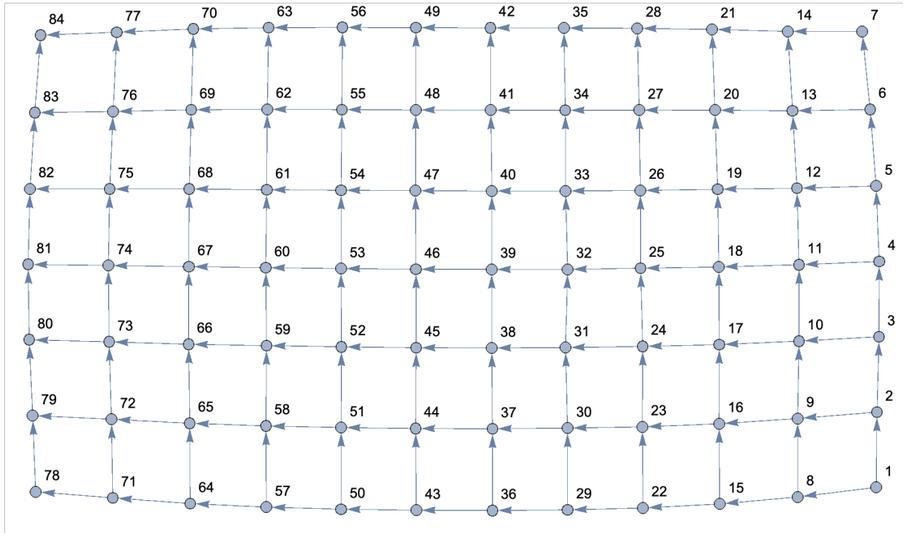
Constant traveling patterns over a grid graph

In this section we do the model extension and simulation over a regular grid graph with a constant traveling patterns matrix.

Here we create a grid graph with directed edges:

In[32]:=
```
{m, n} = {7, 12};
grGrid = GridGraph[{m, n}, DirectedEdges → True, GraphLayout → "SpringEmbedding", VertexLabels → Automatic, ImageSize → Large]
```

Out[33]=



Note that:

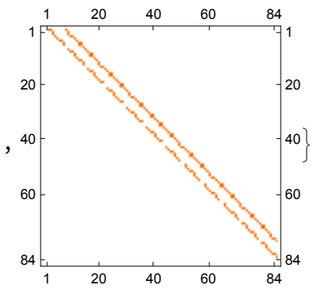
- There is one directed edge between any two edge-connected nodes
- All horizontal edges point in one direction
- All vertical edges point in one direction
- The edges are directed from nodes with smaller indexes to nodes with larger indexes.

Here we make a constant traveling matrix and summarize it:

In[34]:=
```
matGridTravel = AdjacencyMatrix[grGrid] * ConstantArray[1000, {VertexCount[grGrid], VertexCount[grGrid]}];
{ResourceFunction["RecordsSummary"][Flatten[matGridTravel], "All elements"][[1]],
 ResourceFunction["RecordsSummary"][Select[Flatten[matGridTravel], # > 0 &], "Non-zero elements"][[1]], MatrixPlot[matGridTravel]}
```

Out[35]=

1 All elements		1 Non−zero elements	
1st Qu	0	1st Qu	1000
3rd Qu	0	3rd Qu	1000
Median	0	Max	1000
Min	0	Mean	1000
Mean	21.1168	Median	1000
Max	1000	Min	1000



Here we scale the SEI2R model with the grid graph constant traveling matrix:

In[36]:=
```
model1 = SEI2RModel[t, "InitialConditions" → True, "RateRules" → True, "TotalPopulationRepresentation" → "AlgebraicEquation", "BirthsTerm" → True];
```

In[37]:=
```
modelGrid = ToSiteCompartmentsModel[model1, matGridTravel, "MigratingPopulations" → Automatic];
```

Change the initial conditions in the following way:

- Pick initial population size per site (same for all sites)
- Make a constant populations vector
- At all sites except the first one put the infected populations to zero; the first site has one severely symptomatic person
- Set the susceptible populations to be consistent with the total and infected populations.

In[38]:=
```
maxPopulation = 10^6;
lsRPopulations = ConstantArray[maxPopulation, Length[GetPopulationSymbols[modelGrid, "Total Population"]]];
modelGrid =
  SetInitialConditions[
   modelGrid,
   Join[
    Join[AssociationThread[Through[GetPopulationSymbols[modelGrid, "Total Population"][0]], lsRPopulations], <|TP[1][0] → maxPopulation|>],
    Join[Association@Map[#[0] → 0 &, GetPopulationSymbols[modelGrid, "Infected Severely Symptomatic Population"]], <|ISSP[1][0] → 1|>],
    Join[Association@Map[#[0] → 0 &, GetPopulationSymbols[modelGrid, "Infected Normally Symptomatic Population"]], <|INSP[1][0] → 0|>],
    Join[AssociationThread[Through[GetPopulationSymbols[modelGrid, "Susceptible Population"][0]], lsRPopulations], <|SP[1][0] → maxPopulation - 1|>]
   ]
  ];
```

Solve the system of ODE's of the scaled model:

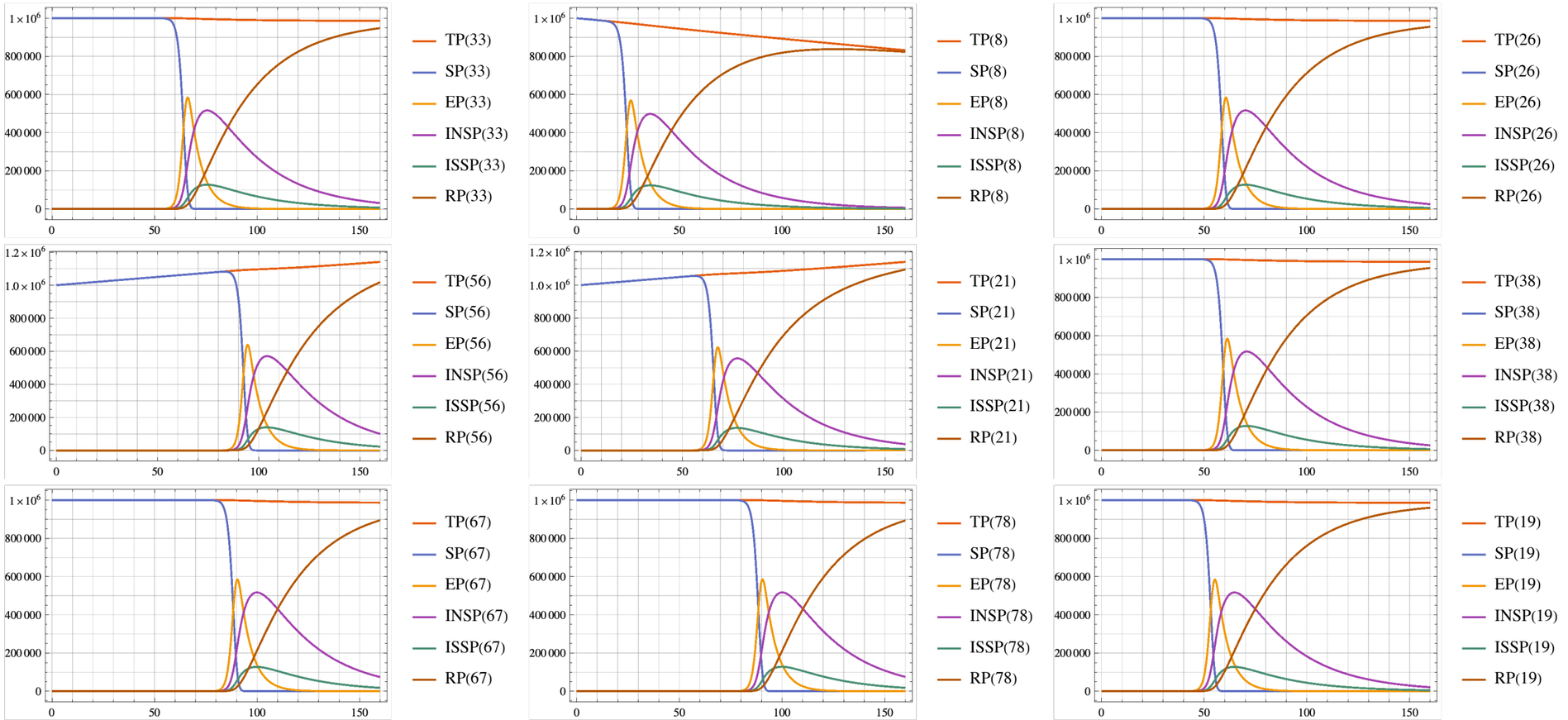
In[41]:=
```
maxTime = 160;
AbsoluteTiming[
 aSolGrid = Association@First@
     NDSolve[
      Join[modelGrid["Equations"] //. modelGrid["RateRules"], modelGrid["InitialConditions"]],
      GetStockSymbols[modelGrid, __ ~~ "Population"],
      {t, 0, maxTime}
     ];
]
```

Out[42]=
```
{1.64186, Null}
```

Randomly sample the graph sites and display the solutions separately for each site in the sample:

In[43]:=
```
Multicolumn[
 Table[
  Block[{aSol = KeySelect[aSolGrid, MatchQ[#, _Symbol[i]] &]},
   Plot[Evaluate[Map[#[t] &, Values[aSol]]], {t, 0, maxTime}, PlotRange → All, GridLines → All, PlotTheme → "Scientific", PlotLegends → Keys[aSol], ImageSize → 300]
  ], {i, RandomSample[Range[VertexCount[grGrid]], UpTo[9]]}],
 3]
```

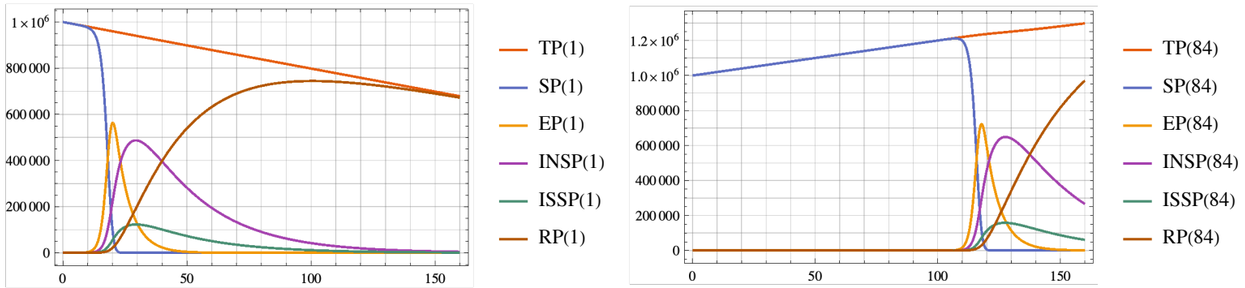
Out[43]=



Display solutions of the first and last site:

In[44]:=
```
Multicolumn[
 Table[
  Block[{aSol = KeySelect[aSolGrid, MatchQ[#, _Symbol[i]] &]},
   Plot[Evaluate[Map[#[t] &, Values[aSol]]], {t, 0, maxTime}, PlotRange → All, GridLines → All, PlotTheme → "Scientific", PlotLegends → Keys[aSol], ImageSize → 300]
  ], {i, {1, VertexCount[grGrid]}}],
 3]
```

Out[44]=



As expected from the graph structure, we can see in the first site plot that its total population is decreasing -- nobody is traveling to the first site. Similarly, we can see in the last site plot that its total population is increasing -- nobody leaves the last site.

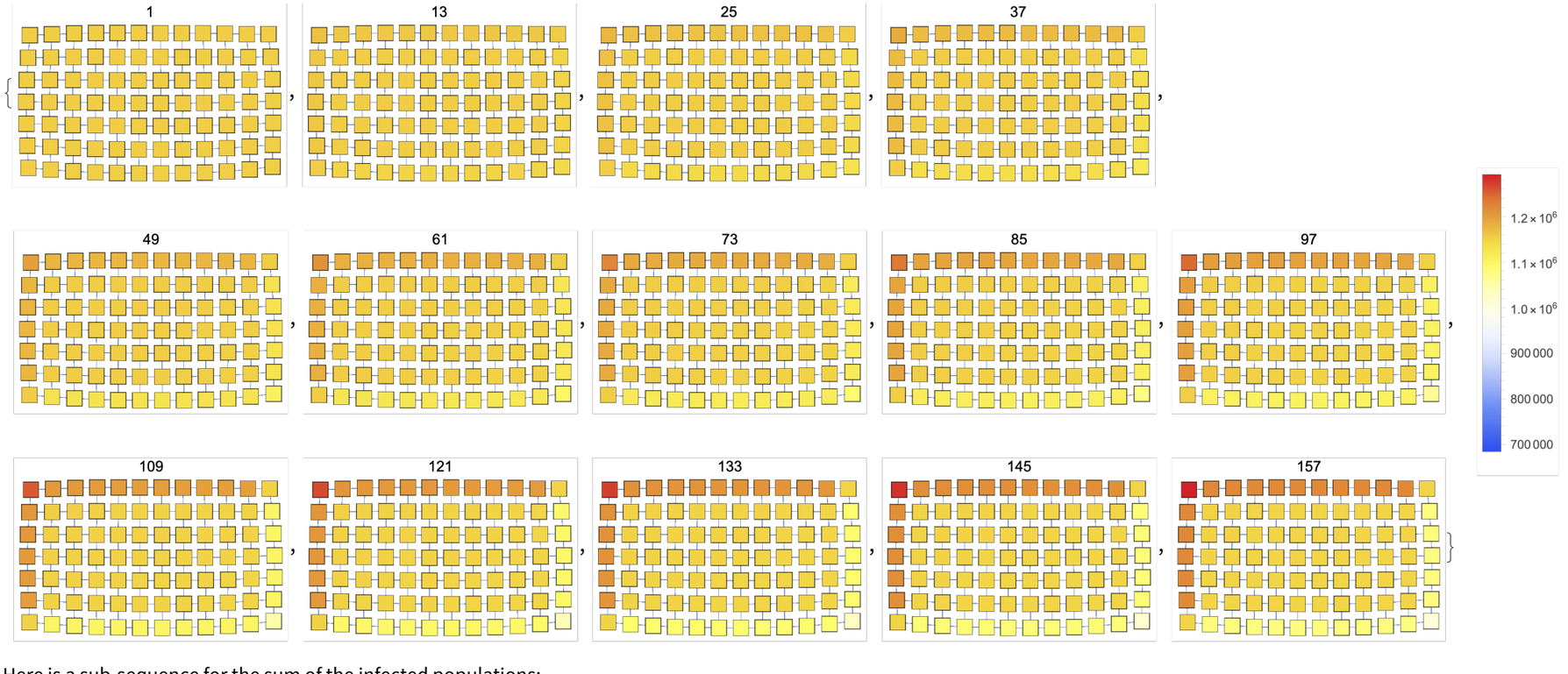
Graph evolution visualizations

We can visualize the spatial-temporal evolution of model's populations using sequences of graphs. The graphs in the sequences are copies of the multi-site graph each copy having its nodes colored according to the populations in the solutions steps.

Here is a sub-sequence for the total populations:

In[45]:=
```
EvaluateSolutionsOverGraph[grGrid, modelGrid, "Total Population", aSolGrid, {1, maxTime, 12},
 "NodeSizeFactor" → 5, "ColorScheme" → "TemperatureMap", "Legended" → True, VertexLabels → None, ImageSize → 200]
```

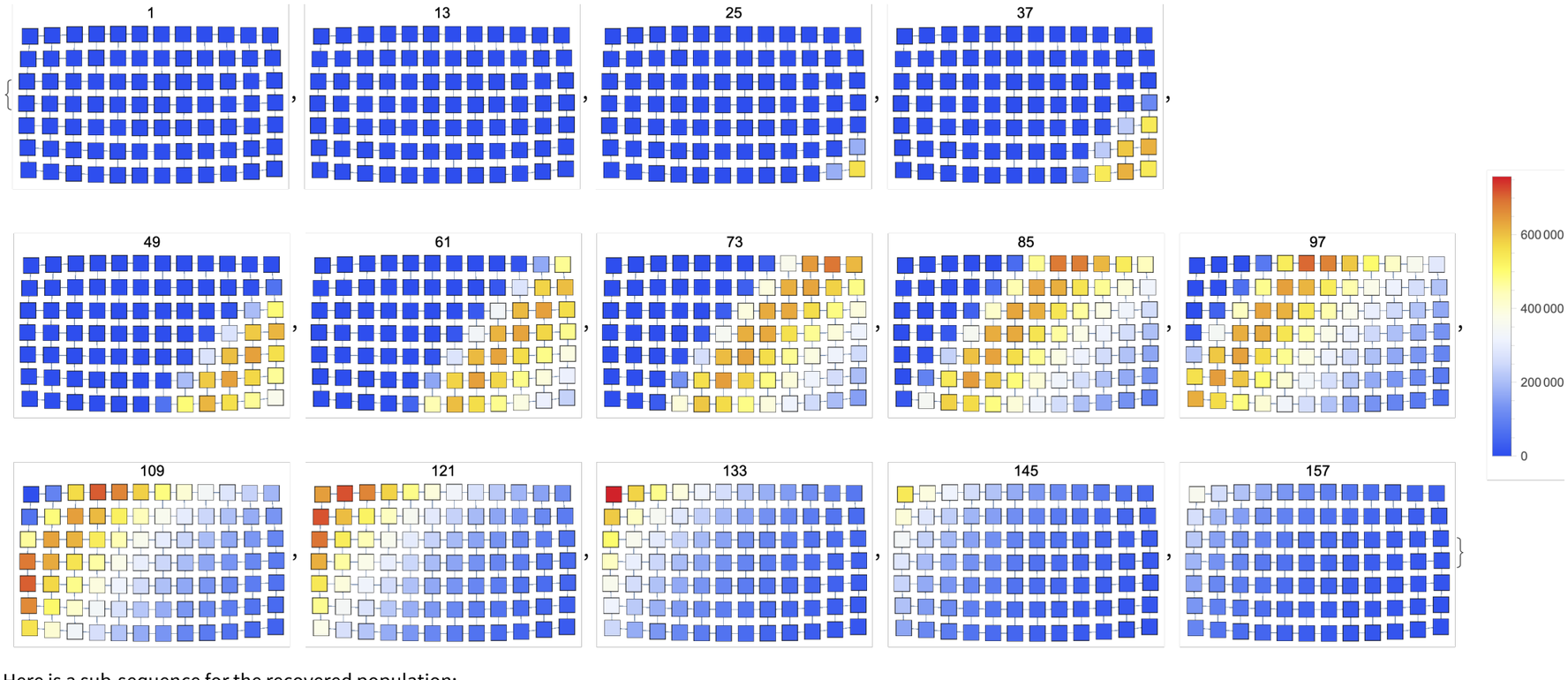
Out[45]=



Here is a sub-sequence for the sum of the infected populations:

In[46]:=
```
EvaluateSolutionsOverGraph[grGrid, modelGrid, {"Infected Normally Symptomatic Population", "Infected Severely Symptomatic Population"},
 aSolGrid, {1, maxTime, 12}, "NodeSizeFactor" → 5, "ColorScheme" → "TemperatureMap", "Legended" → True, VertexLabels → None, ImageSize → 200]
```

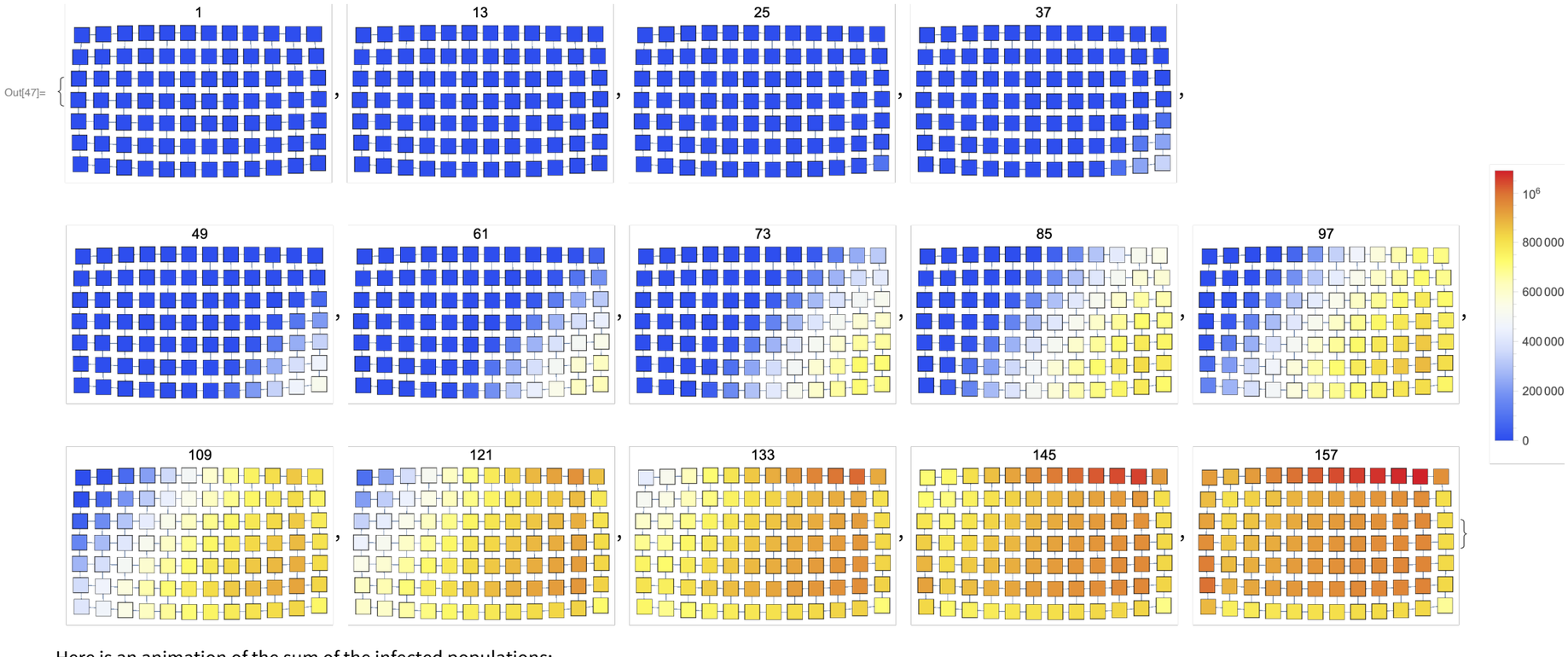
Out[46]=



Here is a sub-sequence for the recovered population:

In[47]:=
```
EvaluateSolutionsOverGraph[grGrid, modelGrid, "Recovered Population", aSolGrid, {1, maxTime, 12},
 "NodeSizeFactor" → 5, "ColorScheme" → "TemperatureMap", "Legended" → True, VertexLabels → None, ImageSize → 200]
```

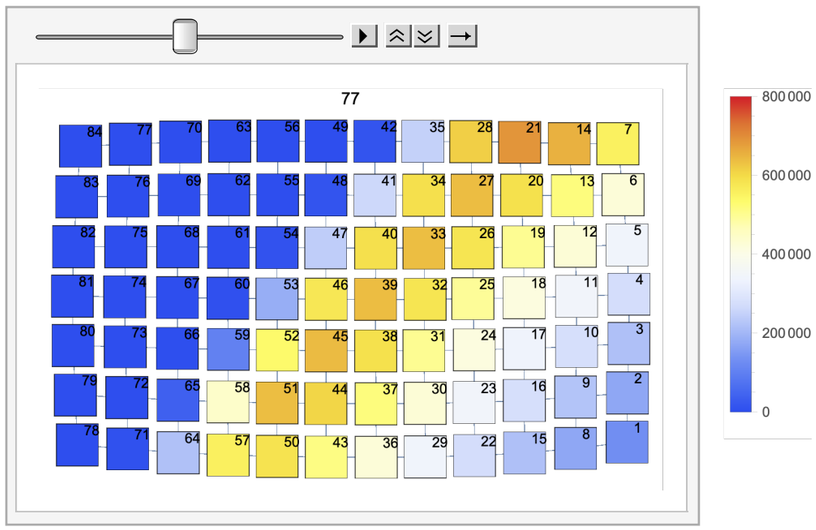
Out[47]=



Here is an animation of the sum of the infected populations:

In[48]:=
```
Block[{stocks = {"Infected Normally Symptomatic Population", "Infected Severely Symptomatic Population"}, colorScheme = "TemperatureMap", timeStep = 4},
 Legended[
  ListAnimate[EvaluateSolutionsOverGraph[grGrid, modelGrid, stocks, aSolGrid, {1, maxTime, timeStep}, "NodeSizeFactor" → 6, "ColorScheme" → colorScheme, ImageSize → 400]],
  BarLegend[{colorScheme, MinMax[EvaluateSolutionsOverGraphVertexes[grGrid, modelGrid, stocks, aSolGrid, {1, maxTime, timeStep}]]}]]
 ]
]
```

Out[48]=

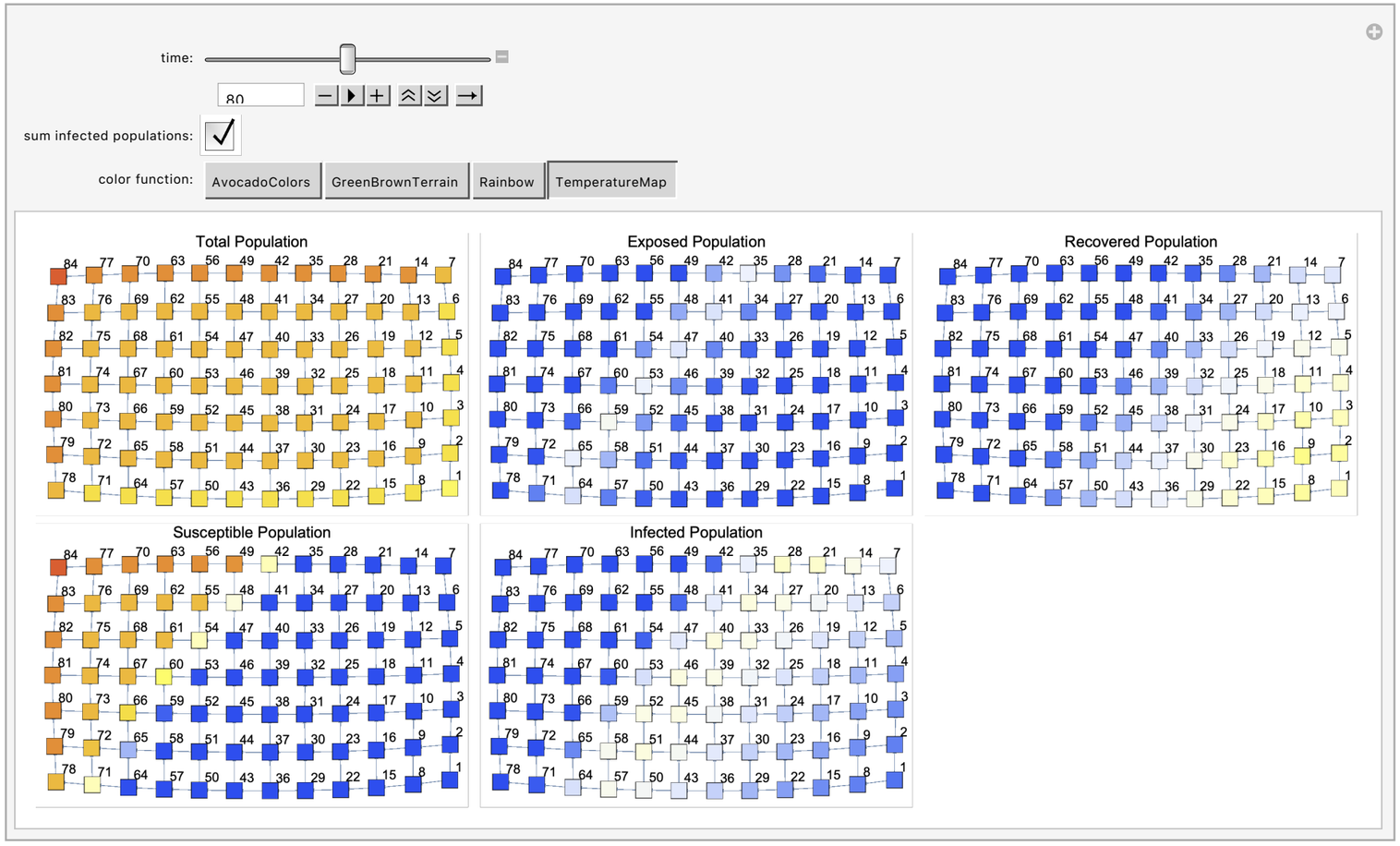


Interactive interface

With this interactive interface we see the evolution of all populations across the graph:

In[49]:=
```
Manipulate[
 Block[{aSol = aSolGrid},
  DynamicModule[{gr = grGrid, imageSize = 300, factor = 3.2, maxPopulation = aSolGrid[TP[VertexCount[grGrid]]][maxTime], vfTP, vfSP, vfEP, vfIP, vfINSP, vfISSP, vfRP, lsRes},
   vfTP[{xc_, yc_}, name_, {w_, h_}] :=
    {ColorData[cf, "ColorFunction"][Rescale[aSol[TP[name]][time], {0, maxPopulation}, {0, 1}]], Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfSP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[SP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfEP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[EP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfIP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[INSP[name]][time] + aSol[ISSP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfINSP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[INSP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfISSP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[ISSP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   vfRP[{xc_, yc_}, name_, {w_, h_}] := {ColorData[cf, "ColorFunction"][Rescale[aSol[RP[name]][time], {0, maxPopulation}, {0, 1}]],
     Rectangle[{xc - factor w, yc - factor h}, {xc + factor w, yc + factor h}]};
   lsRes = {
     GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfTP, PlotLabel → "Total Population"],
     GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfSP, PlotLabel → "Susceptible Population"],
     GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfEP, PlotLabel → "Exposed Population"],
     If[addInfectedPopulationsQ,
      GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfIP, PlotLabel → "Infected Population"],
      (*ELSE*)
      Sequence @@ {GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfINSP, PlotLabel → "Infected Normally Symptomatic Population"],
        GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfISSP, PlotLabel → "Infected Severely Symptomatic Population"]}
     ],
     GraphPlot[gr, ImageSize → imageSize, VertexShapeFunction → vfRP, PlotLabel → "Recovered Population"]
    };
   Multicolumn[lsRes, 3]
  ]],
 {{time, 80, "time:"}, 0, maxTime, 1, Appearance → {"Open"}},
 {{addInfectedPopulationsQ, True, "sum infected populations:"}, {True, False}},
 {{cf, "TemperatureMap", "color function:"}, {"AvocadoColors", "GreenBrownTerrain", "Rainbow", "TemperatureMap"}}]
```

Out[49]=



Observations

Obviously the simulations over the described grid graph, related constant traveling patterns matrix, and constant populations have the purpose to build confidence in conceptual design of MSEMEA and its implementation.

The following observations agree with our expectations for MSEMEA's results over the "grid graph test".

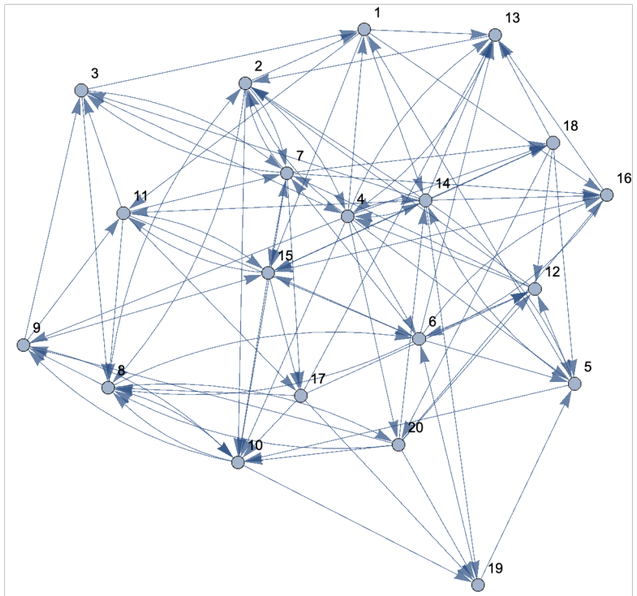
1. The populations plots at each site resemble the typical plots of SEI2R.
2. The total population at the first site linearly decreases.
3. The total population at the last site linearly increases.
4. The plots of the total populations clearly have gradually increasing gradients from the low index value nodes to the high index value nodes.
5. For the infected populations there is a clear wave that propagates diagonally from the low index value nodes to the high index value nodes.
   - 5.1. In the direction of the general "graph flow."
6. The front of the infected populations wave is much steeper (gives "higher contrast") than the tail.
   - 6.1. This should be expected from the single-site SEI2R plots.
7. For the recovered populations there is a clear "saturation wave" pattern that indicates that the recovered populations change from 0 to values close to the corresponding final total populations.

Time-dependent traveling patterns over a random graph

In this section we apply the model extension and simulation over a random graph with random time-dependent traveling patterns matrix.

In[50]:=
```
SeedRandom[84];
grRandom = RandomGraph[{20, 100}, DirectedEdges → True, GraphLayout → "SpringElectricalEmbedding", VertexLabels → "Name"]
```

Out[51]=



Remark: The computations in this section work with larger random graphs; we use a small graph for more legible presentation of the workflow and results. Also, the computations should clearly demonstrate the ability to do simulations with real life data.

Derive a traveling patterns matrix with entries that are random functions:

In[52]:=
```
Block[{gr = grRandom},
 matRandomTravel = AdjacencyMatrix[gr] * RandomInteger[{10, 100}, {VertexCount[gr], VertexCount[gr]}] * Table[Sin[RandomReal[{0.02, 0.1}] t], VertexCount[gr], VertexCount[gr]];
]
```

Here is a *fragment* of the matrix:

In[53]:=
```
Magnify[MatrixForm[matRandomTravel[[1 ;; 12, 1 ;; 12]]], 0.7]
```

Out[53]=

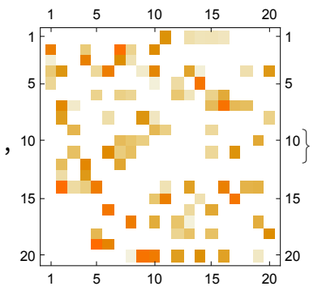
$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 64\sin[0.0694401\,t] & 0 \\
67\sin[0.0924486\,t] & 0 & 0 & 66\sin[0.0323956\,t] & 0 & 0 & 80\sin[0.0967228\,t] & 18\sin[0.0986403\,t] & 0 & 95\sin[0.0502723\,t] & 0 & 0 \\
13\sin[0.0332999\,t] & 0 & 0 & 66\sin[0.0907264\,t] & 0 & 0 & 71\sin[0.0663898\,t] & 24\sin[0.0514156\,t] & 0 & 0 & 0 & 0 \\
45\sin[0.0490302\,t] & 51\sin[0.0836639\,t] & 0 & 0 & 20\sin[0.0808982\,t] & 65\sin[0.0954059\,t] & 0 & 0 & 14\sin[0.0544798\,t] & 95\sin[0.0560608\,t] & 0 & 0 \\
26\sin[0.0646974\,t] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17\sin[0.0209341\,t] & 0 & 16\sin[0.0935063\,t] \\
0 & 0 & 0 & 0 & 40\sin[0.0444516\,t] & 0 & 34\sin[0.0550003\,t] & 83\sin[0.0205249\,t] & 0 & 0 & 0 & 31\sin[0.0750774\,t] \\
0 & 61\sin[0.0840154\,t] & 33\sin[0.0231161\,t] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 84\sin[0.0465466\,t] & 0 & 0 & 0 & 51\sin[0.0269879\,t] & 0 & 0 & 56\sin[0.0762898\,t] & 29\sin[0.036303\,t] & 0 & 0 \\
0 & 0 & 86\sin[0.0321295\,t] & 0 & 0 & 0 & 0 & 0 & 0 & 42\sin[0.0624568\,t] & 74\sin[0.0254778\,t] & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 59\sin[0.043806\,t] & 36\sin[0.0667602\,t] & 38\sin[0.0513656\,t] & 0 & 0 & 0 \\
0 & 0 & 34\sin[0.0737147\,t] & 0 & 0 & 89\sin[0.0557151\,t] & 47\sin[0.0429756\,t] & 43\sin[0.0591347\,t] & 0 & 0 & 0 & 0 \\
0 & 59\sin[0.0432117\,t] & 0 & 86\sin[0.0614769\,t] & 0 & 0 & 39\sin[0.0884544\,t] & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Summarize and plot the matrix at $t = 1$:

In[54]:=
```
Block[{matTravel = matRandomTravel /. t → 1},
 {ResourceFunction["RecordsSummary"][Flatten[matTravel], "All elements"][[1]],
  ResourceFunction["RecordsSummary"][Select[Flatten[matTravel], # > 0 &], "Non-zero elements"][[1]], MatrixPlot[matTravel]}
]
```

Out[54]=

1 All elements		1 Non–zero elements	
1st Qu	0	Min	0.319096
Median	0	1st Qu	1.63562
Min	0	Median	2.56621
3rd Qu	0.159548	Mean	3.19489
Mean	0.798722	3rd Qu	4.46208
Max	8.09391	Max	8.09391



Here we scale the SEI2R model with the random traveling matrix:

In[55]:=
```
model1 = SEI2RModel[t, "InitialConditions" → True, "RateRules" → True, "TotalPopulationRepresentation" → "AlgebraicEquation"];
```

In[56]:=
```
modelRandom = ToSiteCompartmentsModel[model1, matRandomTravel,
   "MigratingPopulations" → {"Susceptible Population", "Exposed Population", "Infected Normally Symptomatic Population", "Recovered Population"}];
```

Change the initial conditions in the following way:

- Pick maximum population size per site
- Derive random populations for the sites
- At all sites except the first one put the infected populations to zero; the first site has one severely symptomatic person
- Set the susceptible populations to be consistent with the total and infected populations.

In[57]:=
```
maxPopulation = 10^6;
lsRPopulations = RandomReal[{100 000, maxPopulation}, Length[GetPopulationSymbols[modelRandom, "Total Population"]]];
modelRandom =
  SetInitialConditions[
   modelRandom,
   Join[
    Join[AssociationThread[Through[GetPopulationSymbols[modelRandom, "Total Population"][0]], lsRPopulations], <|TP[1][0] → maxPopulation|>],
    Join[Association@Map[#[0] → 0 &, GetPopulationSymbols[modelRandom, "Infected Severely Symptomatic Population"]], <|ISSP[1][0] → 1|>],
    Join[Association@Map[#[0] → 0 &, GetPopulationSymbols[modelRandom, "Infected Normally Symptomatic Population"]], <|INSP[1][0] → 0|>],
    Join[AssociationThread[Through[GetPopulationSymbols[modelRandom, "Susceptible Population"][0]], lsRPopulations], <|SP[1][0] → maxPopulation - 1|>]
   ]
  ];
```

Here solve the obtained system of ODE's:

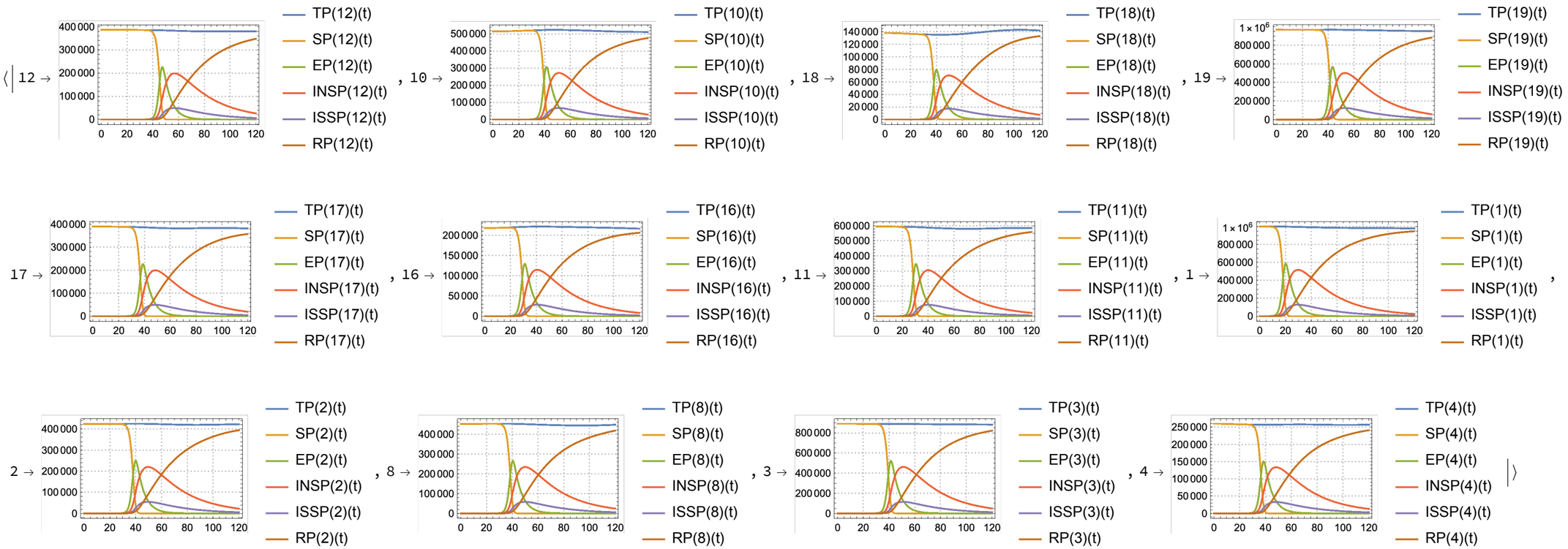
In[60]:=
```
maxTime = 120;
AbsoluteTiming[
 aSolRandom = Association@First@
      NDSolve[
       Join[modelRandom["Equations"] //. modelRandom["RateRules"], modelRandom["InitialConditions"]],
       GetStockSymbols[modelRandom, __ ~~ "Population"],
       {t, 0, maxTime}
      ];
]
```

Out[61]= {0.231681, Null}

Here we plot the solutions:

In[62]:=
```
ParametricSolutionsPlots[<||>, #, None, maxTime, "Together" → True, PlotTheme → "Detailed", ImageSize → Small][[1]] & /@ RandomSample[GroupBy[Normal@aSolRandom, #[[1, 1]] &, Association], UpTo[12]]
```

Out[62]=

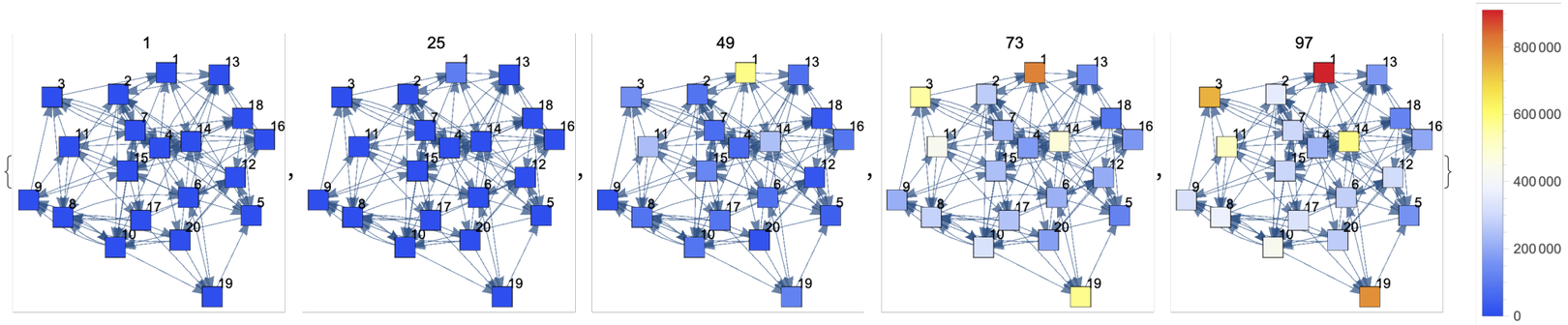


Graph evolution visualizations

As in the previous section we can visualize the spatial-temporal evolution of model's populations using sequences of graphs:

In[63]:=
```
Legended[EvaluateSolutionsOverGraph[grRandom, modelRandom, {"Recovered Population"},
  aSolRandom, {1, maxTime, 24}, "NodeSizeFactor" → 4, "ColorScheme" → "TemperatureMap", "Normalization" → "Global"],
 BarLegend[{"TemperatureMap", {0, Max[Values[EvaluateSolutionsOverGraphVertexes[grRandom, modelRandom, {"Recovered Population"}, aSolRandom, {1, maxTime, 24}]]]}}]]
```

Out[63]=



Money from lost productivity

The model SEI2R from [AAp1] has the stock “Money from Lost Productivity” shown as MLP(*t*) in the equations:

In[64]:=
```
ModelGridTableForm[KeyTake[SEI2RModel[t], "Equations"]]
```

Out[64]=

⟨| Equations →

#	Equation
1	$SP'[t] == -\frac{INSP[t] \times SP[t] \times \beta[INSP]}{TP[t]} - \frac{ISSP[t] \times SP[t] \times \beta[ISSP]}{TP[t]} - SP[t] \times \mu[TP]$
2	$EP'[t] == \frac{INSP[t] \times SP[t] \times \beta[INSP]}{TP[t]} + \frac{ISSP[t] \times SP[t] \times \beta[ISSP]}{TP[t]} - EP[t] \left(\frac{1}{aincp} + \mu[TP]\right)$
3	$INSP'[t] == -\frac{INSP[t]}{aip} + \frac{EP[t] (1-sspf[SP])}{aincp} - INSP[t] \times \mu[INSP]$
4	$ISSP'[t] == -\frac{ISSP[t]}{aip} + \frac{EP[t] \times sspf[SP]}{aincp} - ISSP[t] \times \mu[ISSP]$
5	$RP'[t] == \frac{INSP[t]+ISSP[t]}{aip} - RP[t] \times \mu[TP]$
6	$MLP'[t] == lpcr[ISSP, INSP] (-RP[t] - SP[t] + TP[t])$

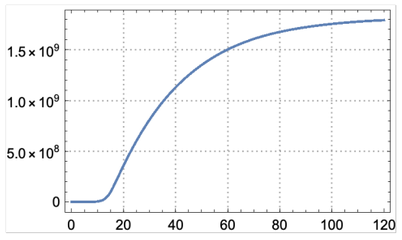
|⟩

Here are MLP plots from the two-node graph model:

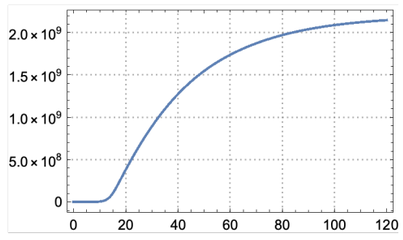
In[65]:=
```
ParametricSolutionsPlots[modelBig["Stocks"], #, None, 120, "Together" → True, PlotTheme → "Detailed", ImageSize → 250][[1]] & /@
 GroupBy[Normal@KeySelect[aSol, MemberQ[{MLP}, Head[#]] &], #[[1, 1]] &, Association]
```

Out[65]=

⟨| 1 →



MLP(1)(t), Money of lost productivity , 2 →

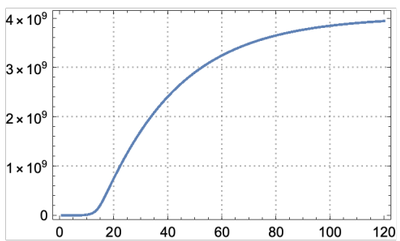


MLP(2)(t), Money of lost productivity |⟩

Here we plot the sum of the accumulated money losses:

In[66]:=
```
funcs = Values[KeySelect[aSol, MemberQ[{MLP}, Head[#]] &]];
ListLinePlot[{#, Total[Through[funcs[#]]]} & /@ Range[1, maxTime], PlotTheme → "Detailed", ImageSize → 250]
```

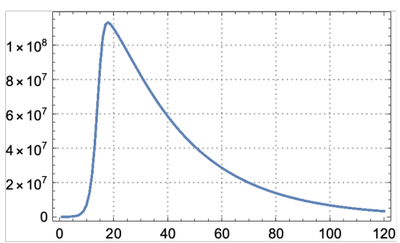
Out[67]=



Here is the corresponding “daily loss” (derivative):

In[68]:=
```
funcs = Map[D[#[t], t] &, Values[KeySelect[aSol, MemberQ[{MLP}, Head[#]] &]]];
ListLinePlot[{#, Total[funcs /. t → #]} & /@ Range[1, maxTime], PlotTheme → "Detailed", ImageSize → 250]
```

Out[69]=



Future plans

There are multiple ways to extend the presented algorithm, MSEMEA. Here are a few most immediate ones:

1. Investigate and describe the conditions under which MSEMEA performs well, and under which it “blows up”
2. Apply MSEMEA together with single site models that have large economics parts
3. Do real data simulations related to the spread of COVID-19.

References

Articles, books

[Wk1] Wikipedia entry, "Compartmental models in epidemiology".

[HH1] Herbert W. Hethcote (2000). "The Mathematics of Infectious Diseases". SIAM Review. 42 (4): 599–653. Bibcode:2000SIAMR..42..599H. doi:10.1137/s0036144500371907.

[AA1] Anton Antonov, "Coronavirus propagation modeling considerations", (2020), SystemModeling at GitHub.

[AA2] Anton Antonov, "Basic experiments workflow for simple epidemiological models", (2020), SystemModeling at GitHub.

[AA3] Anton Antonov, "Air pollution modeling with gridMathematica", (2006), Wolfram Technology Conference.

[ZZ1] Zahari Zlatev, Computer Treatment of Large Air Pollution Models. 1995. Kluwer.

Repositories, packages

[WRI1] Wolfram Research, Inc., "Epidemic Data for Novel Coronavirus COVID-19", WolframCloud.

[AAr1] Anton Antonov, Coronavirus propagation dynamics project, (2020), SystemModeling at GitHub.

[AAp1] Anton Antonov, "Epidemiology models Mathematica package", (2020), SystemsModeling at GitHub.

[AAp2] Anton Antonov, "Epidemiology models modifications Mathematica package", (2020), SystemsModeling at GitHub.

[AAp3] Anton Antonov, "Epidemiology modeling visualization functions Mathematica package", (2020), SystemsModeling at GitHub.

[AAp4] Anton Antonov, "System dynamics interactive interfaces functions Mathematica package", (2020), SystemsModeling at GitHub.