SENSAI: README file

1 Software requirements

1. MATLAB 7.10.0 (R2010a)
2. Maple 14

NOTE: This version of SENSAI is no longer being maintained. It requires a MATLAB and Maple interface, which is no longer maintained as of MATLAB version 7.12 (R2011a). If you have a version of MATLAB at or above 7.12, you must use one of the other versions of SENSAI, which uses MuPAD. Unfortunately at this point, some of the examples do not work in the newer version, as MuPAD is not as effective as Maple just yet.

2 A Practical Guide to Using SENSAI

There are two basic ways to use SENSAI, directly from the GUI or indirectly using a Maple input file. If the model contains up to four equations with six parameters, the user may run SENSAI directly through the GUI (see section 2.1). Alternatively, the user may construct and save his or her model in a separate Maple worksheet and run it indirectly through the GUI (see section 2.2). A glossary of SENSAI terms appears in section 2.3.

2.1 Using SENSAI directly

1. Open MATLAB.
2. Within MATLAB, change the directory to the location in which the SENSAI software has been downloaded. We will call this `SENSAI_DIRECTORY`, e.g. C:/SENSAI/.
3. Open the SENSAI GUI by typing `sensai` in the MATLAB command window.
4. Select check box "Input from GUI?".
5. Select check box "Iterated nonlinear map?" if the model is in the form of a map.
6. Select check box "Compute solutions only?" to compute solutions but not sensitivities and elasticities.
7. Enter a job name to create a subfolder in which solutions will be stored.
8. Enter the model, using $[\cdot]$ to denote vector elements, i.e., Maple syntax. (An example of a simple SIR model is provided.)
9. Select "Create MATLAB files using Maple" which creates MATLAB files `gvec.m`, `dgvec_dxvec.m`, `dgvec_dparam.m`, `qoi.m` and `dcp_dparam.m` in `SENSAI_DIRECTORY`. Wait until a popup box appears that says "MATLAB files successfully created" before continuing.
10. Enter parameter values.
11. Select the desired plots in "Plotting Information" box.
12. Select "Execute MATLAB file created by Maple".
13. SENSAI creates a directory `SENSAI_DIRECTORY/JOB_NAME` in which it stores the MATLAB files `gvec.m`, `dgvec_dxvec.m`, `dgvec_dparam.m`, `qoi.m` and `dcp_dparam.m`, the input data, the plots and a binary file `output.mat` containing `xdim`, `kdim`, `tfinal`, `t`, `x`, `p`, `q`, `dxdp`, `dqdparam`, `elxp` and `elqp`.
14. Results from a new run can be saved into another folder in `SENSAI_DIRECTORY` by changing the Job Name in the second box in the upper right corner of the SENSAI GUI. (E.g. "run2").

2.2 Using SENSAI indirectly

1. Open MATLAB

2. Within MATLAB, change the directory to the location in which the SENSAI software has been downloaded. We will call this `SENSAI_DIRECTORY`, e.g. C:/SENSAI/.

3. Open the SENSAI GUI by typing `sensai` in the MATLAB command window.

4. Make sure the check box "Input from GUI?" is *unselected*.

5. Create Maple worksheet Model_Name.mw in what will be called `WORKING_DIRECTORY` by copying and modifying templates e.g.,

   Examples/ODE_examples/SIR/SIR.mw, or

   Examples/MAP_examples/Caswell08/Caswell08.mw.

   It is important that the right-hand side is a vector named g and the variables are in a vector named x, as SENSAI will look for these specific variable names.

6. Copy the path of `WORKING_DIRECTORY` into the box in the upper right hand within the GUI.

   (e.g. C:/SENSAI/Examples/ODE_examples/SIR/)

   Note: *The active directory within* MATLAB *must be the same one that contains the sensai.m program, i.e.,* `SENSAI_DIRECTORY`. *This is especially important for step 8.*

7. Execute Maple worksheet Model_Name.mw which creates MATLAB files `user_equations.m`, `user_inputs.m`, `user_plotdata.m`, `user_qoi.m` and `user_parameters.m` in `WORKING_DIRECTORY`.

8. In the SENSAI GUI, select "Create MATLAB files using Maple" which creates the files `gvec.m`, `dgvec dxvec.m`, `dgvec dparam.m`, `qoi.m`, and `dcp dparam.m` in `SENSAI_DIRECTORY`. Wait until a popup box appears that says "MATLAB files successfully created" before continuing.

9. Within MATLAB, control of the program is through the files `user_inputs.m` and `user_plotdata.m`, in `WORKING_DIRECTORY`.

   (a) Via `user_inputs.m`, you control parameter values, initial conditions, and the name of the subfolder of `WORKING_DIRECTORY` in which you wish to save your work (using "JOB").

   (b) Via `user_plotdata.m`, you control which solutions (x-values) to output and plot (using `ilist`), and which parameters to have their sensitivities tested (using `klist`).

10. Within MATLAB in the GUI, select "Execute MATLAB file created by Maple".

    All plots of the solutions, their sensitivities and elasticities are saved in the subfolder `WORKING_DIRECTORY/JOB`. All solutions including sensitivities and elasticities are stored in the binary file `WORKING_DIRECTORY/JOB/output.mat`.

11. Before performing another run using the SENSAI GUI, return to the SENSAI directory and enter the MATLAB commands
    >> `close all, clear all`
    to clear both plots and active memory before continuing. (Optional, but probably a good idea). Results from a new run can be saved into another folder in the `WORKING_DIRECTORY` by changing the name of "JOB" in `user_inputs.m`, e.g. JOB = 'run2'. (This can also be achieved by changing this line in the Maple file *but then the* Maple *file must be re-run and you will return to step 7.*)

2.3 Glossary

2.3.1 .m-files used in SENSAI

Name	Description
`sensai.m`	Main MATLAB file that controls the SENSAI GUI
`mm_interface.m`	Main MATLAB file that interfaces MATLAB with Maple
`gtype.m`	Main MATLAB function that directs the solving, plotting, and storing of data
`gvec.m`	Automatically generated MATLAB file that creates vector of right-hand side functions
`dgvec_dxvec.m`	Automatically generated MATLAB file that computes derivatives of right-hand side with respect to variables
`dgvec_dparam.m`	Automatically generated MATLAB file that computes derivatives of right-hand side with respect to parameters
`qoi.m`	Automatically generated MATLAB file that constructs quantity of interest and its sensitivity information
`r0_matrix.m`	Automatically generated MATLAB file that computes R_0 and its sensitivity information
`cp.m`	Automatically generated MATLAB file that constructs user-defined parameter and its sensitivity information

Table 1: Names and descriptions of .m-files used in SENSAI

2.3.2 Terms used in SENSAI

Name	Description
`xdim`	Dimension of variables $\mathbf{x}$
`kdim`	Dimension of parameters $\mathbf{p}$
`tfinal`	Final time step
`imap`	Logical that defines whether the system is a map or ode
`solution_only`	Logical that defines whether only the solutions are to be computed
`x0`	Vector of initial conditions
`p`	Vector of parameters
`ilist`	List of a subset of variables to be included in plots
`klist`	List of a subset of parameters to be included in plots
`iplot`	Structure that determines which figures to plot
`NextGen`	List of a subset of variables that describe infective classes
`R0_only`	Logical that defines whether R_0 is to be computed without its sensitivities
`DIR`	`WORKING_DIRECTORY`
`JOB`	Folder name where data from a run is stored

Table 2: Names and descriptions of terms used in SENSAI

2.3.3 Variables in *output.mat*

Name	Dimensions	Description
`xdim`	1	Number of variables
`kdim`	1	Number of parameters
`tfinal`	1	Final time step
`p`	kdim$\times$1	Parameters
`t`	$T \times 1$	Vector of times at which the solutions and sensitivities are determined
`x`	xdim $\times T$	Matrix of solutions at all times $t(\cdot)$
`q`	$T \times 1$	Quantity of interest
`r0`	1	The basic reproduction number R_0
`dxdp`	xdim $\times$ kdim $\times T$	Sensitivity of variables with respect to parameters
`dqdparam`	kdim$\times T$	Sensitivity of the quantity of interest with respect to parameters
`dR0dp`	kdim $\times$1	Sensitivity of R_0 with respect to parameters
`elxp`	xdim $\times$ kdim $\times T$	Elasticity of variables with respect to parameters
`elqp`	kdim $\times T$	Elasticity of quantity of interest with respect to parameters
`elR0p`	kdim $\times$1	Elasticity of R_0 with respect to parameters
`cparam`	1	User defined parameter value
`dxdc`	xdim $\times T$	Sensitivity of variables with respect to user defined parameter
`dqdc`	$T \times 1$	Sensitivity of the quantity of interest with respect to user defined parameter
`dR0dc`	1	Sensitivity of R_0 with respect to user defined parameter
`elxc`	xdim $\times T$	Elasticity of variables with respect to user defined parameter
`elqc`	$T \times 1$	Elasticity of quantity of interest with respect to user defined parameter
`elR0c`	1	Elasticity of R_0 with respect to user defined parameter

Table 3: Names, dimensions, and descriptions of variables in *output.mat*.

For notation purposes, T is the dimension of time steps MATLAB uses to solve the system, not to be confused with `tfinal`. The value T is *not* a variable in SENSAI.

3 Defining R_0 in SENSAI

For appropriate epidemiological models, SENSAI is capable of automatically calculating the basic reproduction ratio, R_0, as defined by the Next Generation method. However, since SENSAI is not limited to infection modeling, a specific syntax is required so that SENSAI recognizes if a model is compatible with the definition of R_0. The following guide instructs the user how to edit existing Maple templates so that SENSAI will produce R_0 and its sensitivities with respect to parameters and initial conditions.

1. Edit the Maple templates to define your model equations.

2. Indicate which equations describe the dynamics of infected classes. Store these indices in the variable $NextGen$. Note that $NextGen$ must be written in *matrix* syntax. Consider the following examples.

   (a) If the model includes three states, S, I, and R, in that order, $NextGen := matrix([2])$;.

   (b) If the model has more than one equation describing an infected class, list them in the order they appear. For example, if the model describes $S_1, I_1, R_1, S_2, I_2, R_2$ in that order, $NextGen := matrix([2, 5])$;.

   (c) *If you do not wish to calculate R_0 for the model, define $NextGen = matrix([0])$, or let the first entry of $NextGen$ be 0.*

3. If the model has four or more infected classes, you may want to consider computing R_0 without its sensitivities. R_0 will be a very lengthy expression for such models, and the analytic derivatives will require a lot of time to compute. If this is the case, define "R0_only" to be 1. If you wish to calculate the sensitivities anyway, define "R0_only" = 0.

   (While running the SENSAI GUI, you may encounter large delays in "Create MATLAB files using Maple" if R0_only = 0. If your patience has

run thin, you must terminate the program through the task manager. The emergency stop in MATLAB of CTRL+c in the command window will not work, as the computation of R_0 is done externally in a Maple procedure call.)

4. If the analytic expression for R_0 is already known, it may be faster (and more accurate if Maple can not solve R_0) to use this expression of R_0 as the quantity of interest (*qoi*) instead of re-deriving the expression during the "Create MATLAB files using Maple" phase.

3.1 Automatic construction of $\mathcal{F}, \mathcal{T}, \mathcal{V}$

1. For ODEs, the Next Generation definition of $R_0 = \rho(FV^{-1})$ where $F = \frac{\partial \mathcal{F}_i}{\partial x_j}(x^\star) \quad 1 \le i, j \le m$ describes new infections and $V = \frac{\partial \mathcal{V}_i}{\partial x_j}(x^\star) \quad 1 \le i, j \le m$ describes transfer of existing infections, $x^\star$ is the disease-free equilibrium, the infected classes are $1, \ldots, m$, and $\rho(\cdot)$ denotes the spectral radius operator.

2. For maps, the Next Generation definition of $R_0 = \rho(F(I-T)^{-1})$, where I is the $m \times m$ identity and F and $-T$ are defined the same as F and V for ODEs, respectively.

The following criteria are used by SENSAI to determine the placement of each term. If the terms of the model will not be placed in the biologically correct vectors, SENSAI fails to compute the Next Generation R_0.

1. If the term X in an equation describing an infective class involves a state variable from a noninfectious class, $X \in \mathcal{F}$, unless the occurrence of the noninfectious state variable is part of a sum of all state variables (that is, the term is scaled by the total population).

   Important Note for MuPAD: Within the current release, MuPAD is not as effective as Maple at analytical computations. There may be examples (like the Hantavirus model) where MuPAD can not account for the scaling by the total population. This problem is due to limitations of the `subs()` command and will hopefully be addressed in a future versions of MuPAD.

2. If the term X in an equation describing an infective class does not involve any state variables and is only a parameter, product of parameters, or quotient of parameters, $X \in \mathcal{F}$. However, if terms like these exist, the disease-free subspace will not be invariant, and the model will not have a valid Next Generation R_0.

3. Every other term X that does not satisfy the above will be placed in $\mathcal{V}$ for ODEs, or $\mathcal{T}$ for maps.

3.2 Potential problems with R_0

There are some examples in which the Next Generation construction of R_0 is not valid, or is not compatible with SENSAI. possible problems the user might encounter when trying to define R_0.

3.2.1 Potential problems with ODE models.

For R_0 to be valid, the model must satisfy the conditions of Theorem 2 of [8], specifically:

(A1) The fecundity matrix F is nonnegative.

The transition matrix V is nonsingular.

(A4) The disease-free subspace is invariant. That is, infection can enter a disease-free population through a nonzero component in a state that is identified as disease-free. This can occur in models with background infection rates, or in models where the infective classes are not identified properly.

(A5) The equilibrium is asymptotically stable in the absence of disease. That is, if $\mathcal{F} = 0$, all eigenvalues of the Jacobian of the full system evaluated at $x^\star$ have negative real part.

Notice that assumptions (A2) and (A3) of [8] for ODE models are not automatically checked by SENSAI. These assumptions must be verified by the user, but are usually true.

3.2.2 Potential problems with map models.

For R_0 to be valid, the model must satisfy the conditions of Theorem 2.1 in [1], specifically:

1. The fecundity matrix F is nonnegative.
2. The transition matrix T is nonnegative.
3. The transition matrix T is nonsingular.
4. The transition matrix T is asymptotically stable. That is, $\rho(T) < 1$.
5. The equilibrium is asymptotically stable in the absence of disease. That is, $\rho(C) < 1$ where C is the Jacobian of the right-hand side of the noninfectious states.

For map models, the assumption of a unique DFE is not checked by SENSAI, nor is the condition that $F + T$ is irreducible. These should also be checked by the user to ensure a valid R_0. It is difficult to check both of these conditions, but again, for most models, $F + T$ is irreducible based on the structure of T having a nonzero main diagonal and a sub-diagonal and the structure of F having a nonzero top row.

4 Template Examples

4.1 MAP Examples

4.1.1 Caswell 08

This model can be found in [4]. It involves two stages, juveniles (x_1) and adults (x_2) and the map

$$\mathbf{x}(t+1) = \begin{pmatrix} \sigma_1(1-\gamma) & f \\ \sigma_1\gamma & \sigma_2 \end{pmatrix} \mathbf{x}(t),$$

where the juvenile survival $\sigma_1(\mathbf{x}) = \tilde{\sigma}e^{-\mathbf{e}^{\mathrm{T}}\mathbf{x}}$, where $\mathbf{e}$ is a vector of ones, σ_2 is the adult survival, γ is the maturation probability, and f is the adult fertility. The parameter values given are $(f, \gamma, \tilde{\sigma}, \sigma_2) = (0.25, 1/15, 0.98, 0.95)$.

The main purpose of this model is to verify that SENSAI gives the same equilibrium and sensitivities to those mentioned in the paper. This model also serves as a template for MAP examples.

4.1.2 Hantavirus

This model is from [1]. It involves susceptible and infected male and female rodents. The map is

$$\left.\begin{aligned} S_m(t+1) &= \left[\frac{B}{2} + e^{-\beta_m I_m(t) - \beta_f I_f(t)} S_m(t)\right] D(N) \\ I_m(t+1) &= \left[(1 - e^{-\beta_m I_m(t) - \beta_f I_f(t)}) S_m(t) + I_m(t)\right] D(N) \\ S_f(t+1) &= \left[\frac{B}{2} + e^{-\beta_f I_m(t) - \beta_f I_f(t)} S_f(t)\right] D(N) \\ I_f(t+1) &= \left[(1 - e^{-\beta_f I_m(t) - \beta_f I_f(t)}) S_f(t) + I_f(t)\right] D(N) \end{aligned}\right\}$$

where the logistic growth is scaled by

$$D(N) = \frac{K}{K + (b/2)N},$$

where K is the carrying capacity and N is the total population. The birth function is the harmonic mean

$$B(N_m, N_f) = \frac{2bN_mN_f}{N},$$

where $N_m = S_m + I_m$ is the total number of males, $N_f = S_f + I_f$ is the total number of females, and $b > 0$ is the average litter size, k is the number of contacts that result in an infection, and β_m and β_f are the infection rate constants of males and females, respectively. Parameter values are not provided in [1], but some reasonable values are $K = 1000$, $\beta_f = 0.09$, $\beta_m = 0.9$, and $b = 6$.

The main purpose of this model is to verify that SENSAI gives the same value of R_0 as provided by [1]. This example exhibits two common practices in model formulation which SENSAI performs extremely well *in this version.* First, every term is scaled by the total population N. Second, the infection rate is given by a probability of an infection occurring. Even with these two difficulties, SENSAI is able to identify which terms belong to $\mathcal{F}$ and $\mathcal{V}$ correctly to produce a valid R_0. (Unfortunately, for the users with a later version, SENSAI is not able to handle the first issue due to limitations in MuPAD's `subs()` command.)

4.1.3 Six Stage Genetics

This model is based on [4], adding partially dominant selection based on viability and two alleles. The full model is explained in detail in [7].

4.2 ODE Examples

4.2.1 SIR

This model is a typical SIR model with logistic growth.

$$\left.\begin{aligned} \frac{dS}{dt} &= rN\left(1-\frac{N}{K}\right) - \beta SI - \delta S, \\ \frac{dI}{dt} &= \beta SI - \gamma I - \mu I - \delta I, \\ \frac{dR}{dt} &= \gamma I - \delta R, \end{aligned}\right\}$$

where $N = S + I + R$ is the total population at any time t, r is the per capita growth rate, K is the carrying capacity, β is the infection rate, δ is the natural death rate of the species, γ is the recovery rate, and μ is the disease specific death rate. Some reasonable parameter values are $r = 0.5$, $K = 1000$, $\beta = 0.1$, $\delta = 0.2$, $\gamma = 0.02$, and $\mu = 0.1$.

The main purpose of this model is to demonstrate the calculation of R_0 for a standard ODE infection model which can be easily verified by hand. This model also serves as a template for ODE examples.

4.2.2 SI (Indirect Transmission)

This model is an SI model that involves indirect transmission of the infection, and is

$$\left.\begin{aligned} \frac{dS}{dt} &= rN\left(1-\frac{N}{K}\right) \\ \frac{dI}{dt} &= \beta - \gamma I \end{aligned}\right\}$$

where $N(t) = S(t) + I(t)$ is the total population, r is the per capita growth rate, K is the carrying capacity, β is the background transmission probability, and γ is the recovery rate.

A detailed explanation of this model can be found in [6]. Parameter values may be chosen as $r = 0.5$, $\beta = 0.8$, $\gamma = 0.02$, and $K = 1000$.

The main purpose of this model is to show that models with a background (indirect) transmission of the disease through the environment or some alternative source do not have a valid R_0.

4.2.3 Plague

This model is given by [3]. The first three classes are the SIR classes of rats, N is the average number of fleas living on a rat, and F is the number of free infectious fleas that are searching for a new host. The system of ordinary differential equations is

$$\left.\begin{aligned}
\dot{S}_R &= r_R S_R \left(1 - \frac{T_R}{K_R}\right) + r_R R_R (1-p) - d_R S_R - \beta_R \frac{S_R}{T_R} F(1 - e^{-aT_R}) \\
\dot{I}_R &= \beta_R \frac{S_R}{T_R} F(1 - e^{-aT_R}) - (d_R + m_R) I_R \\
\dot{R}_R &= r_R R_R \left(p - \frac{T_R}{K_R}\right) + m_R g_R I_R - d_R R_R \\
\dot{N} &= r_F N \left(1 - \frac{N}{K_F}\right) + \frac{d_F}{T_R} F(1 - e^{-aT_R}) \\
\dot{F} &= (d_R + m_R(1 - g_R)) I_R N - d_F F
\end{aligned}\right\}$$

where $T_R = S_R + I_R + R_R$ is the total size of the rat population, r_R is the net rat reproduction rate, K_R is the rat carrying capacity, p is the proportion of offspring that inherit the disease, d_R is the natural rat death rate, β_R is the transmission rate from rats to fleas, m_R is the rate that rats leave the infected class, g_R is the fraction of rates that become resistant, a is the searching efficiency of the fleas, r_F is the net flea reproductive rate, d_F is the natural flea death rate, and K_F is the flea carrying capacity.

4.2.4 Dengue

This model is given by [5]. It is an SEIR model that describes the dynamics of dengue fever, an infection carried by a vector, mosquitos. The equations

are

$$\left.\begin{aligned}
\frac{dS_H}{dt} &= \Pi_H - \lambda_H S_H - \mu_H S_H \\
\frac{dE_H}{dt} &= \lambda_H S_H - (\sigma_H + \mu_H) E_H \\
\frac{dI_H}{dt} &= \sigma_H E_H - (\tau_H + \mu_H + \delta_H) I_H \\
\frac{dR_H}{dt} &= \tau_H I_H - \mu_H R_H \\
\frac{dS_V}{dt} &= \Pi_V - \lambda_V S_V - \mu_V S_V \\
\frac{dE_V}{dt} &= \lambda_V S_V - (\sigma_V + \mu_V) E_V \\
\frac{dI_V}{dt} &= \sigma_V E_V - (\mu_V + \delta_V) I_V
\end{aligned}\right\}$$

where $\lambda_H = \frac{C_{HV}}{N_H}(\eta_V E_V + I_V)$ is the human infection rate, $\lambda_V = \frac{C_{HV}}{N_H}(\eta_H E_H + I_H)$ is the vector infection rate, and $N_H = S_H + E_H + I_H + R_H$ is the total human population. The other parameter values and interpretations can be found in Table 4.

Parameter	Numerical Value	Interpretation
μ_H	0.0195	$1/\mu_H$ is average human lifespan
σ_H	0.5300	Rate of transfer of exposed to infected humans
Π_H	10	Human recruitment rate
δ_H	0.9900	Disease specific human death rate
η_H	0.9900	Infectiousness factor of exposed to infected humans
τ_H	0.2000	Human recovery rate
μ_V	0.0140	$1/\mu_V$ is average vector lifespan
σ_V	0.2000	Rate of transfer of exposed to infected vectors
Π_V	30	Vector birth rate
δ_V	0.0057	Disease specific vector death rate
η_V	0.9800	Transmissibility factor of exposed to infectious vectors
C_{HV}	0.038	Infection rate of mosquitoes

Table 4: Parameter Values for Dengue Model.

An interesting initial condition for the model is $\mathbf{x_0} = (\frac{\Pi_H}{\mu_H}, 0, 0, 0, \frac{\Pi_V}{\mu_V}, 0, 200)$, which will show that even though $R_0 < 1$, infection may still persist in the population.

4.2.5 Typhoid

This model is given by [2]. It is a 9-stage SIR type model with the variables as described in Table 5 and the model equations are

$$\left.\begin{aligned}
\dot{x_1} &= -(\rho_{12}+\rho_{13})x_1 y + \rho_{41}x_4 + \rho_{51}x_5 + \rho_{61}x_6 + \rho_{81}x_8 + \rho_{91}x_9 - \mu x_1 + \mu \\
\dot{x_2} &= \rho_{12}x_1 y - (\rho_{23}+\rho_{24}+\rho_{25}+\mu)x_2 + \rho_{32}x_3 \\
\dot{x_3} &= \rho_{13}x_1 y - (\rho_{32}+\rho_{34}+\rho_{35}+\mu)x_3 + \rho_{23}x_2 \\
\dot{x_4} &= \rho_{24}x_2 + \rho_{34}x_3 + \rho_{54}x_5 - (\rho_{41}+\rho_{45}+\rho_{46}+\rho_{48}+\mu)x_4 \\
\dot{x_5} &= \rho_{25}x_2 + \rho_{35}x_3 + \rho_{45}x_4 - (\rho_{51}+\rho_{54}+\rho_{58}+\mu)x_5 \\
\dot{x_6} &= \rho_{46}x_4 - (\rho_{61}+\rho_{67}+\rho_{68}+\mu)x_6 \\
\dot{x_7} &= \rho_{67}x_6 - \mu x_7 \\
\dot{x_8} &= \rho_{48}x_4 + \rho_{58}x_5 + \rho_{68}x_6 - (\rho_{81}+\rho_{89}+\mu)x_8 \\
\dot{x_9} &= \rho_{89}x_8 - (\rho_{91}+\mu)x_9.
\end{aligned}\right\}$$

Parameter values are given in Table 5.

Name	Description
x_1	susceptibles
x_2	incubating noninfectious
x_3	incubating infectious
x_4	sick infectious
x_5	sick noninfectious
x_6	temporary carrier
x_7	permanent carrier
x_8	short resistance
x_9	long resistance

Table 5: Variables for Typhoid Model.

This example demonstates SENSAI's ability to implement a large system and compute R_0 effectively. In this model, states x_2 through x_7 are considered infective, so the next generation matrix is a 6×6 matrix with analytical (not numerical) components. Parameter values can be found in Table 6.

Parameter	Numerical Value	Interpretation
ρ_{12}	8.43381×10^{-3}	Infection rate
ρ_{13}	8.51900×10^{-5}	Infection rate
ρ_{23}	2.85720×10^{-3}	Transfer rate from state of incubation
ρ_{24}	6.78585×10^{-2}	Transfer rate from state of incubation
ρ_{25}	7.14300×10^{-4}	Transfer rate from state of incubation
ρ_{32}	7.14300×10^{-4}	Transfer rate from state of incubation
ρ_{34}	6.42870×10^{-2}	Transfer rate from state of incubation
ρ_{35}	6.42870×10^{-3}	Transfer rate from state of incubation
ρ_{41}	3.46000×10^{-3}	Transfer rate from state of sickness
ρ_{45}	3.46000×10^{-3}	Transfer rate from state of sickness
ρ_{46}	3.46000×10^{-3}	Transfer rate from state of sickness
ρ_{48}	2.40124×10^{-2}	Transfer rate from state of sickness
ρ_{51}	3.46000×10^{-3}	Transfer rate from state of sickness
ρ_{54}	6.92000×10^{-3}	Transfer rate from state of sickness
ρ_{58}	2.40124×10^{-2}	Transfer rate from state of sickness
ρ_{61}	1.11100×10^{-3}	Transfer rate from temporary carrier
ρ_{67}	3.33300×10^{-3}	Transfer rate from temporary carrier
ρ_{68}	6.66600×10^{-3}	Transfer rate from temporary carrier
ρ_{81}	2.74000×10^{-4}	Transfer rate from short resistance
ρ_{89}	2.46600×10^{-3}	Transfer rate from short resistance
ρ_{91}	2.74000×10^{-4}	Transfer rate from long resistance
μ	5.48000×10^{-5}	Overall birth and death rate

Table 6: Parameter Values for Typhoid Model according to Bailey, all units are $days^{-1}$. Note that each ρ_{ij} is a transfer rate from state i to state j.

References

[1] L. Allen and P. van den Driessche. The basic reproduction number in some discrete-time epidemic models. *Journal of Difference Equations and Applications*, 14:1127–1147, 2008.

[2] N.T.J. Bailey and J. Duppenthaler. Sensitivity analysis in the modelling of infectious disease dynamics. *Journal of Mathematical Biology*, 10:113–131, 1980. 10.1007/BF00275837.

[3] M. Buzby, D. Neckels, M.F. Antolin, and D. Estep. Analysis of the sensitivity properties of a model of vector-borne bubonic plague. *Journal of The Royal Society Interface*, 5(26):1099–1107, 2008.

[4] H. Caswell. Perturbation analysis of nonlinear matrix population models. *Demographic Research*, 18(3):59–116, 2008.

[5] S.M. Garba, A.B. Gumel, and M.R. Abu Bakar. Backward bifurcations in dengue transmission dynamics. *Mathematical Biosciences*, 215(1):11 – 25, 2008.

[6] M. Mikucki. Sensitivity Analysis of the Basic Reproduction Number and other Quantities for Infectious Disease Models. Master's thesis, Colorado State University, 2012.

[7] S.J. Tavener, M. Mikucki, S.G. Field, and M.F. Antolin. Transient sensitivity analysis for nonlinear population models. *Methods in Ecology and Evolution*, 2(5):560–575, 2011.

[8] P. van den Driessche and J. Watmough. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical Biosciences*, 180(1-2):29 – 48, 2002.