SENSAI: README file

1 Software requirements

SENSAI uses MATLAB and the symbolic MATLAB software package called MuPad.

2 A Practical Guide to Using SENSAI GUI

1. Open MATLAB
2. Within MATLAB, change the directory to the location of sensai.m. This is SENSAI directory, e.g., C:/SENSAI/.
3. Open the SENSAI GUI by typing `sensai` in the MATLAB command window.

2.1 Input from the GUI

Assuming "Input from GUI?" selected. (This is limited to four equations and six variables.)

1. Select check box "Iterated nonlinear map?" if the model is in the form of a map.
2. Select check box "Compute solutions only?" to compute solutions but not sensitivities and elasticities.
3. Enter data using [.] to denote vector elements, i.e., MuPAD syntax.

4. Select "Create MATLAB files using MuPAD" which creates `gvec.m`, `dgvec_dxvec.m`, `dgvec_dparam.m`, `qoi.m` and `dcp_dparam.m`. Wait until a popup box appears that says "MATLAB files successfully created" before continuing.

5. Select plots required in "Plotting Information" box.

6. Select "Execute MATLAB file created by MuPAD".

7. 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, dxdp, q, dqdparam, elxp` and `elqp`.

8. Results from a new run can be saved into another folder in the SENSAI directory by changing the Job Name in the second box in the upper right corner of the SENSAI GUI. (E.g. "run2").

9. Back to step 7(b) if the model equations or quantity of interest is new. Back to step 7(d) if the only changes are in plotting data, parameter values, or initial conditions.

2.2 Input from a MuPAD file

Assuming "Input from GUI?" is *not* selected,

1. Create MuPAD worksheet user_input.mu using templates e.g.,
   Examples/ODE_examples/SIR/SIR.mu, or
   Examples/MAP_examples/Caswell08/Caswell08.mu

2. Execute MuPAD worksheet user_input.mu which creates MATLAB files `user_equations.m`, `user_input.m`, `user_plotdata.m`, `user_QoI.m`, `user_parameters.m`, `user_bifndata.m`, and `user_FIMdata.m`

3. Find the folder with the MuPAD file that contains the program and input field for the model (e.g. C:/SENSAI/Examples/ODE_examples/SIR/, C:/SENSAI/Examples/MAP_examples/Caswell08/, etc.).

   - This is the WORKING directory.

- Copy the path of the WORKING directory into the box in the upper right hand within the GUI (e.g. C:/SENSAI/Examples/ODE_examples/SIR/).

4. In the SENSAI GUI, select "Create MATLAB files using MuPAD" which creates the files `gvec.m`, `dgvec dxvec.m`, `dgvec dparam.m`, `qoi.m`, and `dcp dparam.m` within the SENSAI directory. Note: *The active directory within* MATLAB *must be the same one that contains the sensai.m program, i.e. the* SENSAI *directory.* Wait until a popup box appears that says "MATLAB files successfully created" before continuing.

5. Within MATLAB, control of the program is through the files `user_inputs.m` and `user_plotdata.m`, in the WORKING directory with the MuPAD file containing the program.

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

6. Within MATLAB in the GUI, select "Execute MATLAB file created by MuPAD".

2.3 Outputs

1. All of the plots of the solutions, sensitivities, and elasticities specified in the run of the model, and a file with all of the outputs from the model (output.mat) will be saved in the WORKING directory in a folder named by the variable string "JOB."

   - To get the solutions, sensitivity values, and elasticities into data files that can be plotted, either work within MATLAB on the data in output.mat, or ...
   - Use the exported information in the text files that can be imported into other programs for plotting (e.g. R). The (large number of) files each contain the solutions, sensitivities, and elasticities for the run specified above.

2. Before carrying out another run using the SENSAI GUI, within MATLAB, return to the SENSAI directory and enter the commands to clear both plots and active memory before moving on (optional, but probably a good idea):
   `>> close all, clear all`

   (a) 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 done by changing this line in the MuPAD file. *(But this is overkill, since you must go back to step 8(b) after this point.)*
        `>> JOB_NAME:= "run2"; # Sets the folder name for the output.`

   (b) Modify values in `user_inputs.m` and `user_plotdata.m` in the WORKING directory to explore other values.

   (c) Back to step 8(*f*).

3 Basic reproduction number

Please see R0_readme.pdf

4 Bifurcation

Please see BIFN_readme.pdf

5 Active subspaces, Fisher information and parameter estimation

Please see FIM_readme.pdf

6 Template Examples

6.1 MAP Examples

Please see R0_readme.pdf

6.2 ODE Examples

6.2.1 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 that it is a standard ODE infection model with a simple R_0 that can be easily verified by hand. This model also serves as a great template for ODE examples.

6.2.2 SI (Indirect Transmission)

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

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

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.

The explaination of this model in detail can be found in *Sensitivity Analysis of the Basic Reproduction Number and other Quantities for Infectious Disease Models*, Masters Thesis by Mikucki, 2012. 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.

6.2.3 Plague

This model is given by Buzby et. al. in *Analysis of the sensitivity properties of a model of vector-borne bubonic plague*, 2008. 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.

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

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 inherity 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.

6.2.4 Dengue

This model is given by Garba in *Backward bifurcations in dengue transmission dynamics*, 2008. It is an SEIR model that descirbes the dynamics of dengue fever, an infection carried by a vector, mosquiteos. 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 parameter values provided are $\mu_H = 0.0195$, $\sigma_H = 0.5300$, $\Pi_H = 10$, $\delta_H = 0.9900$, $\eta_H = 0.9900$, $\tau_H = 0.2000$, $\mu_V = 0.0140$, $\sigma_V = 0.2000$, $\Pi_V = 30$, $\delta_V = 0.0057$, $\eta_V = 0.9800$, and $C_{HV} = 0.038$.

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 thought $R_0 < 1$, infection may still persist in the population.

6.2.5 Typhoid

This model is given by Bailey and Duppenthaler in *Sensitivity Analysis in the Modelling of Infectious Disease Dynamics*, 1980. It is a 9-stage SIR type model where 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, and x_9 = long resistance.

$$\left.\begin{aligned}
\dot{x}_1 &= -(\rho_{12}+\rho_{13})x_1 y + \rho_{41}x_4 + \rho_{51}x_5 + \rho_{61}x_6 + \rho_{81}x_8 + \rho_{91}x_9 - \mu x_1 + \mu \\
\dot{x}_2 &= \rho_{12}x_1 y - (\rho_{23}+\rho_{24}+\rho_{25}+\mu)x_2 + \rho_{32}x_3 \\
\dot{x}_3 &= \rho_{13}x_1 y - (\rho_{32}+\rho_{34}+\rho_{35}+\mu)x_3 + \rho_{23}x_2 \\
\dot{x}_4 &= \rho_{24}x_2 + \rho_{34}x_3 + \rho_{54}x_5 - (\rho_{41}+\rho_{45}+\rho_{46}+\rho_{48}+\mu)x_4 \\
\dot{x}_5 &= \rho_{25}x_2 + \rho_{35}x_3 + \rho_{45}x_4 - (\rho_{51}+\rho_{54}+\rho_{58}+\mu)x_5 \\
\dot{x}_6 &= \rho_{46}x_4 - (\rho_{61}+\rho_{67}+\rho_{68}+\mu)x_6 \\
\dot{x}_7 &= \rho_{67}x_6 - \mu x_7 \\
\dot{x}_8 &= \rho_{48}x_4 + \rho_{58}x_5 + \rho_{68}x_6 - (\rho_{81}+\rho_{89}+\mu)x_8 \\
\dot{x}_9 &= \rho_{89}x_8 - (\rho_{91}+\mu)x_9
\end{aligned}\right\}$$

Parameter values are provided by Bailey are as follows: $\rho_{12} = 8.43381 \times 10^{-3}$, $\rho_{13} = 8.51900 \times 10^{-5}$, $\rho_{23} = 2.85720 \times 10^{-3}$, $\rho_{24} = 6.78585 \times 10^{-2}$, $\rho_{25} = 7.14300 \times 10^{-4}$, $\rho_{32} = 7.14300 \times 10^{-4}$, $\rho_{34} = 6.42870 \times 10^{-2}$, $\rho_{35} = 6.42870 \times 10^{-3}$, $\rho_{41} = 3.46000 \times 10^{-3}$, $\rho_{45} = 3.46000 \times 10^{-3}$, $\rho_{46} = 3.46000 \times 10^{-3}$, $\rho_{48} = 2.40124 \times 10^{-2}$, $\rho_{51} = 3.46000 \times 10^{-3}$, $\rho_{54} = 6.92000 \times 10^{-3}$, $\rho_{58} = 2.40124 \times 10^{-2}$, $\rho_{61} = 1.11100 \times 10^{-3}$, $\rho_{67} = 3.33300 \times 10^{-3}$, $\rho_{68} = 6.66600 \times 10^{-3}$, $\rho_{81} = 2.74000 \times 10^{-4}$, $\rho_{89} = 2.46600 \times 10^{-3}$, $\rho_{91} = 2.74000 \times 10^{-4}$, and $\mu = 5.48000 \times 10^{-5}$.

This example demonstates SENSAI's ability to implement a large system and compute R_0 effectively. In this model, equations 2-7 are considered infective, so the next generation matrix is a 6×6 matrix with analytical (not numerical) components.