**Homework #5**

For all questions, please submit your R code, C++ code, and a doc file with a copy of the screen output and plots (where applicable) to [smtowers@asu.edu](mailto:smtowers@asu.edu). Due Thus April 18th at 6pm.

The code in your R and C++ files should exhibit all of the good coding practices mentioned in <http://sherrytowers.com/2012/12/14/good-programming-practices-in-any-language/>

All plots should exhibit all of the good practices in creating figures, mentioned in

<http://sherrytowers.com/2013/01/04/good-practices-in-producing-plots/>

Please submit your files in a tar’d and gzip’d file with name <first name>\_<initial of last name>.tar.gz

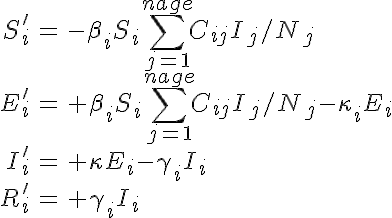
This homework assignment involves a study of a simulated epidemic within a population. Data are number of newly identified cases in children and adults as being newly *infectious* each day (not newly infected… this particular disease has a latent period after initial infection during which individuals are not infectious, nor do they have symptoms. Symptoms begin the day the individual becomes infectious).

The public health authorities had first noticed that the disease was spreading in the population on day 1, which is when the data collection began (which is not necessarily the same day that the disease was first introduced into the population). The data look like this, and are contained in the file <http://www.sherrytowers.com/SEIRage_simulated_epidemic_data.txt>

:



In this homework assignment, you will fit the parameters of an age-structured SEIR model, with nage=2 age groups, to these data:



C\_ij is the number of contacts per day made by a person in age group i with a person in age group j. The parameters for which you will fit are: the reproduction number R0, time of introduction t0 (assume that one infectious case in the child class is introduced at this time), and the pre-immune fractions in both age classes at t0.

In going through the steps involved in the fitting process, you will likely find it useful to refer back to the SIR model example presented in Module XI in class. With some relatively straightforward modifications, most of the R and C++ code related to that example can be re-purposed to be applicable here.

Items of information that are already known about this population and this disease:

1. The fraction of children in the population is 0.25
2. The total population size is 2,500,000
3. The daily contact patterns in the population have been estimated from previous survey studies, where participants were asked to keep track of their daily contact patterns. The contact matrix describing the number of contacts per day between the groups (where K is kids, and A is adults) is C\_KtoK=20, C\_KtoA= 6, C\_AtoK=3, C\_AtoA=12. Assume that these are exactly known quantities.
4. The transmission rate, beta, is the same in both age groups.
5. The progression of the disease has been well studied in previous epidemics, and it is known that the latent period is on average 10 days, and the infectious period is on average14 days. Assume that these are exactly known quantities and assume they are the same for both age groups.
6. The disease is always quite serious, requiring medical attention, so the number of cases identified can be assumed to be the true number in the population.

A) Calculate the next-generation matrix of the system of ODE’s, take the largest real eigenvalue, and solve for beta in terms of the reproduction number R0.

B) Write R code to output the data in a format suitable for inclusion in a C++ class that contains the data.

Use this to help write the C++ code for a class, SimulatedDataSEIRage, that contains the data.

C) Write R code to use the methods in the deSolve library to numerically solve the system of ODE’s for an age-structured SEIR model. Write the code such that it is generalized to any number of age classes.

D) Write the C++ code for a class that has a method to employ the Runge Kutta method to numerically solve an age-structured SEIR model with an arbitrary number of age classes (ie; make the code generalized). Also make it so that beta, kappa, and gamma may be age-specific rather than the same for all age groups.

E) Write a C++ program, SEIR\_test.cpp that employs this method to calculate the number of newly infectious children and number of newly infectious adults in increments of 0.1 days. Assume

1. population size N=2,500,000 with 25% kids,
2. beta=0.10 in both age groups,
3. kappa=1/10 in both age groups
4. gamma = 1/14 in both age groups
5. at t0=0 we have I0\_kids=10, and I0\_adults=10, E0\_kids=0, E0\_adults=0
6. the pre-immune fractions for the kids and adults are 0.7 and 0.8, respectively.

Run the program, and output the estimated epidemic curve information to a file.

Read this file into an R script, SEIR\_test.R that compares the calculation of the C++ program to the same calculation in R with the methods of the deSolve library by making a plot that overlays the results, and also checks the fractional difference between the C++ and R calculations. Do they appear to be in good agreement? How about if you change the time step to 0.5 days? (note that the larger you can make the time step in C++, the faster the fitting process will be).

If you have coded things up correctly in both C++ and R, your plot will look like this:



**DO NOT CONTINUE ON TO THE NEXT PARTS OF THIS HOMEWORK UNTIL YOU HAVE ENSURED THAT YOUR C++ AND R CALCULATIONS RETURN THE SAME EPIDEMIC CURVE!**

F) Write an R script SEIRage\_initial\_fit.R that will randomly uniformly sample values of the reproduction number R0, the time of introduction, and the pre-immune fractions in the child and adult classes, and solve the age structured SEIR model for each parameter hypothesis. Choose ranges of the parameter sweeps that you feel are reasonable.

Calculate the sum of the Pearson chi-squared statistics comparing the model prediction for the number of newly infectious individuals per day in the child and adult classes to the simulated data set. If the total Pearson chi-squared statistic is the best yet calculated so far of all your parameter hypotheses, plot the model predictions overlaid on the data. Every 10th iteration, plot the best-fit model, and also plot the Pearson chi-squared statistic vs the parameter hypotheses for each parameter. In short, you R script should do essentially what the <http://www.sherrytowers.com/SIR_initial_study.R> script does for the SIR model presented in Module XI in class.

Run this script for a modest number of iterations (say, 5000 or more… whatever it takes to ensure that the Pearson chi-squared vs parameter hypothesis plots are populated enough that you can reasonable detect approximately what the best-fit parameter values are), and ensure that the minimum Pearson chi-squared is well within the range of the parameter hypotheses ranges. If not, adjust the parameter ranges to ensure that this is the case, and repeat the iteration process. Keep adjusting the ranges and iteration process until you think you have a reasonable ranges for the parameter sweeps.

Do the data appear to be significantly over-dispersed? If so, correct the Pearson chi-squared statistic for the over-dispersion. Estimate the best-fit parameters and one std dev uncertainties.

Note that when calculating the epidemic curves, it is OK to obtain the numerical solution with larger time steps than you would use in the C++ code: the methods in the deSolve library internally use a dynamical time step calculation. When you call lsoda() with a vector of times at which you want the model estimates, it does not use those times in its internal calculations.

G) Write a C++ program SEIRage\_initial\_fit.cpp what will randomly uniformly sample values of the reproduction number R0, the time of introduction, and the pre-immune fractions in the child and adult classes, using the ranges determined by the R script above.

Calculate the sum of the Pearson chi-squared statistics comparing the number of newly infectious individuals per day in the child and adult classes to the simulated data set. On your local computer, run this program for a modest number of iterations (say, 50000… it will run much faster than your R script even though it has to use a smaller time step). Output the results to a file.

H) Write an R script to read in the file output in step G and plot the Pearson chi-squared statistic (corrected for over-dispersion, if necessary) vs the parameter hypotheses. Do the ranges used to sample the parameters still look reasonable now that your plots are better populated? If not, adjust the ranges, and repeat steps G and H. Estimate the best-fit parameter values and one std dev uncertainties.

I) To obtain well populated plots of the Pearson chi-squared statistic vs the parameter hypotheses, upload your C++ files to Saguaro, and compile them there with the Intel C++ compiler. Submit 100 jobs in batch, with a time limit of 15 minutes each job. Concatenate the results, and copy the file back to your local computer, and use the R script in step H to plot the results and estimate the best-fit parameter values and one std dev uncertainties.