Simulating the Spread of the Common Cold

Richard Corban Harwood
Department of Mathematics
George Fox University
Newberg OR 97132 USA
rharwood@georgefox.edu

Abstract: This modeling scenario guides students to simulate and investigate the spread of the common cold in a residence hall. An example floor plan is given, but the reader is encouraged to use a more relevant example. The two-week project presented here uses a hands-on activity followed by solution of a first order differential equation by the separable method (method of separation of variables) [4] and further graphical analysis using a spreadsheet. In groups, students run repeated simulations, collect data, derive a differential equation model, solve that equation, estimate parameter values by hand and through regression, visually evaluate the consistency of the model with their data, and present their results to the class. The simulation portion can be done in a 50 minute class period and has been successfully implemented on the first day of class. Required supplies include small beans or candy pieces to shake onto the printed floor plan.

Keywords: common cold, data, model, nonlinear, parameter estimation
Tags: hands-on, simulation, residence hall, disease, first order, differential equation, separable method

STATEMENT

Set-Up Instructions

A certain number \( (x_0) \) of residents on a floor of a residence hall (with a total of \( N \) residents) have the common cold and have left traces of the virus at various places. In this simulation, beans scattered onto a printed floor plan represent residents susceptible to the cold milling around this floor. Those that come in contact with the virus become infected and leave more traces of the virus. How does the number of residents infected vary with time?
Simulating the Spread of the Common Cold

1. Get into groups of four and pick up two printed floor plans (see example in Figure 1), four paper clips, and \( N = 30 \) beans (optional: also pick up two plastic cups, two overhead slide pieces, and two dry erase markers). Choose two group members to be “Recorders” and two to be “Simulators,” one for each printed floor plan.

2. For each floor plan, the Simulator should fold it in a way that lays the rooms flat with raised paper sidewalls. For the example in Figure 1, fold along the dotted lines in an alternating fashion (in-out-in) to create a ‘w’ shape. Then pinch the two floors together and fold in between them to make the second and third floors lay down horizontally. The Recorder should then use two paper clips to fasten the middle ridge at both ends.

3. Each Recorder sets up five tables, each with four rows, to record the results of five simulations. Label the four rows Time (round number, starting with zero), Susceptible (number dumped), Infected (number removed from the board and set aside), and Change in Infected (number infected in this round and used to get the number Infected in the next round).

In-Class Simulation

Each Simulator-Recorder pair will do the following to run and record data from five simulations.

1. The Recorder chooses a starting infected count, \( x_0 \), randomly chosen between 1 and \( N - 1 \) and fills in the table for this simulation with 0 (Time), \( N - x_0 \) (Susceptible), \( x_0 \) (Infected). To aid model development, start the first simulation with \( x_0 = 1 \).

2. The Simulator sets \( x_0 \) beans to the side as infected and marks with an ‘X’ that many rooms on the second floor chosen at random locations (excluding bathrooms/hallways/stairwells).

3. The Recorder walls off the two ends of the second floor with their hands, while the Simulator shakes and dumps the susceptible beans from their hand/cup onto the center of the floor. If any land outside of the floor, shake and dump them again.

4. The Simulator leaves in place any beans that are more than halfway inside an infected room, and returns all others back to your hand/cup. For each bean left, find the closest uninfected room to infect and mark it with an ‘X’. Afterwards, add these beans to the infected pile.

5. The Recorder counts those the Simulator left behind, tallies this number (\( \Delta x \)) under Change in Infected, and then updates the table for the next round by subtracting \( \Delta x \) from the number susceptible and adding \( \Delta x \) to the number infected.

6. Repeat steps 3-5 until there are no more susceptible beans left. When complete, erase all marks from the floorplan (or overhead slide strip).

Model Development

As a group, complete the following. This is most easily done on a single computer when the group is together. Otherwise, work on a shareable spreadsheet (such as Google Sheets).
1. Type all ten simulation tables into a spreadsheet, each table expanded as show below. The Change in Infected is a rough estimate for the derivative (rate of change in the infected population), and can be improved by taking the average of the current and previous values to compute the Infected Derivative. Note, this means the leftmost value (at $t = 0$) is $\frac{0 + \Delta x}{2}$. Once the group model is chosen, its solution is computed in the Model: Infected row and square errors are computed below it and summed up.

<table>
<thead>
<tr>
<th>Simulation #</th>
<th>Parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (rounds)</td>
<td>0 1 2 ...</td>
</tr>
<tr>
<td>Data: Susceptible</td>
<td>$N - x_0$</td>
</tr>
<tr>
<td>Infected</td>
<td>$x_0$</td>
</tr>
<tr>
<td>Change in Infected</td>
<td>$\Delta x$</td>
</tr>
<tr>
<td>Infected Derivative</td>
<td>$\frac{0 + \Delta x}{2}$</td>
</tr>
<tr>
<td>Model: Infected</td>
<td></td>
</tr>
<tr>
<td>Square Errors</td>
<td></td>
</tr>
</tbody>
</table>

2. Choosing a simulation with the lowest starting value $x_0$, create a scatter plot with independent variable Time and dependent variables Susceptible, Infected, Infected Derivative.

3. From this scatter plot, individually come up with a possible differential equation model to match the shape of the slope $\frac{dx}{dt}$ (Infected Derivative) using $x$ (Infected), $N - x$ (Susceptible), and $t$ (Time). The instructor may have some suggestions to choose from or leave it open-ended.

4. Get together as a group and compare models. Graph/sketch the slope functions from each and work together to choose/create the model with the most potential. Make sure there is at least one parameter to provide room to fit it to the data.

5. Summarize why your group’s model fits the shape of the data and how the parameters adjust it. Briefly share this with the class and record any feedback. Update your model accordingly.

**Analysis**

Your analysis begins with fitting your model equation using a trend line and then a hands-on optimization with the main parameter(s) using either a local search or global search method.

**Parameter Estimation**

1. Determine the model component $X$ to needed to write your model differential equation as $Y = a_1X + a_0$ (linear) or $Y = a_nX^n + ... + a_1X + a_0$ (polynomial). For example, $x' = a \sin bx$ is linear with $X = \sin bx$ where $b$ can be estimated from the period of the oscillation in the data.

2. Choosing an example simulation with the lowest starting value $x_0$, insert a row labeled Model component: $X$ below the Infected row and compute the model component, $X$.

3. Create a scatter plot with independent variable Model component: $X$ and dependent variable Infected Derivative. Add a trendline and edit it to use the correct form and display the equation and R-squared value on the chart. Also, set the intercept if you have no constant to solve for. This is your initial estimate for the parameter(s). Note, the R-squared value gives the percentage
of the spread in the data explained by a linear relationship and is one way of measuring model fit.

4. Solve your model for the general solution.

5. For each simulation, enter the general solution into the spreadsheet row Model: Infected using your initial estimate for the parameter(s) from this example simulation. Make a scatter plot with independent variable Time and dependent variables Infected, Model: Infected

6. Complete either Local Optimization or Global Optimization section to optimize model parameter estimates for each simulation. Note how the graphs of the model and data are brought closer together with each improved estimate.

7. Summarize the values for the fitted parameter(s) for all ten simulations by the overall mean, and interval [min, max] for each parameter that was estimated this way. If another method was used, such as estimating the period of oscillation make a note of this.

Local Optimization

1. For each simulation, compute \((\text{Infected} - \text{Model:Infected})^2\) in the spreadsheet row Square Errors and sum up the row to compute the sum of square errors: \(\text{SSE} = \sum (\text{Data} - \text{Model})^2\).

2. Choose the initial interval radius, \(\Delta p\), for midpoint \(M = p\) and compute the end points \(L = p - \Delta p\) and \(R = p + \Delta p\). As a rule of thumb, start with \(\Delta p = 10^k\) where \(k\) the power of ten when \(p\) is written in scientific notation.

3. Compute the SSE at end points \(L\) and \(R\).
   - If \(\text{SSE}(L) < \text{SSE}(R)\), then reset \(R = M\) and \(M = \frac{L + M}{2}\) to choose the left sub-interval.
   - Else, reset \(L = M\) and \(M = \frac{M + R}{2}\) to choose the right sub-interval.

4. Repeat steps 2-3 until you have obtained an \(M\) value with three significant digits unchanged from the previous \(M\) value. Set \(p = M\).

Global Optimization

1. For each simulation table, copy the pair of rows, Model: Infected and Square Errors, and paste this pair ten times below it.

2. For a chosen parameter with initial estimate \(p\), choose an initial radius \(\Delta p = \frac{1}{2}p\) of your search interval. If there are any other parameters, use their initial estimate value.

3. Divide the interval \([p - \Delta p, p + \Delta p]\) into ten sub-intervals and assign each of the eleven points to a different model pair.

4. For each model pair, compute \((\text{Infected} - \text{Model:Infected})^2\) in the spreadsheet row Square Errors and sum up the row to compute the sum of square errors: \(\text{SSE} = \sum (\text{Data} - \text{Model})^2\). Choose the parameter value \(p^*\) with minimum SSE.
   - If \(p^*\) is the same as \(p\), reset \(\Delta p = \frac{1}{10}\Delta p\) to zoom-in, and repeat steps 3-4 (unless this makes \(\Delta p < 1E - 14\), in which case go to step 6).

5. If at least three significant digits of \(p^*\) agree with \(p\), then update \(p = p^*\) for the final estimate.
Simulating the Spread of the Common Cold

Otherwise, repeat steps 3-4 with $\Delta p = |p - p^*|$ and $p = p^*$.

6. Repeat this process for each additional parameter.

In-Class Presentation

1. Explain the main steps in deriving your model differential equation
2. Explain the main steps needed in solving your model differential equation
3. From the ten simulations, show example graphs of your best and worst fitting models to the data.
4. Explain in what ways your model described the data well and in what ways did it not.
5. How could your model be improved? How could the simulation of the spread of a cold be improved?

COMMENTS

This project was inspired by an example in [3] which investigates the spread of an infection on an evenly-spaced, two-dimensional grid. If the reader would like to better control variability in infected rooms, such a grid might be preferable. The relevancy of choosing a residence hall (which housed a few of my students), however, provided substantial motivation for this project in my own classes. The following comments go over helpful guidelines for implementation, suggested ways to customize this project, suggested means of assessment, and a transcription of one group’s wiki report. An example Excel spreadsheet “1-37-Simulating Common Cold,” is available in the Supplemental Docs section of this Modeling Scenario to demonstrate sample instructions for a class-shared document and one group’s data analysis.

Implementation

The only differential equations knowledge needed for this project is the separable method [4], but familiarity with the slope field and differential equations with parameters is helpful. I suggest the following two-week (three days a week) schedule for implementing this project and guiding its completion.

Day 1 In-Class Simulation (Topic: Intro to differential equations)
Day 2 Fill group spreadsheet (Topic: Separable method)
Day 3 Finish Model Development (Topic: Slope field)
Day 4 Finish parameter estimation (Topic: Phase line analysis)
Day 5 Finish Analysis (Topic: Integrating factor method)
Day 6 In-Class Presentation (Topic: Project discussion)
I use this project as a discovery technique to introduce the first two weeks of differential equations. As you can see in the suggested topics above, the In-Class Simulation motivates the need for the separable method and develops visual analysis for dealing with slope fields and phase lines. Looking at the graph of the Infected populations across all simulations, students can estimate qualitative
Simulating the Spread of the Common Cold

features of the governing equation such as the existence (and stability) of equilibrium points and the constancy of slopes at fixed population values to determine if the equation is autonomous. Completion of this project is a great jump-off point for numerical methods and more advanced methods and models.

Before the In-Class Simulation activity, it is helpful to have a class discussion about what the simulation represents (spread of a cold throughout a residence hall) and what students think should happen. After groups have gathered their supplies, it is helpful to demonstrate how to fold the paper, decide which of the dropped beans become infected, and how to mark the spread of the infection. This demonstration can be projected with a document camera or onto a printed overhead slide.

A shareable spreadsheet like through Google Sheets is a helpful way for students to collaborate on their own schedule and present together in class. To reduce grading, I suggest a single sheet shared with the whole class with independent tabs for each group. Having access to each other’s work is a risk, but this is an opportunity for students to practice respectful sharing of resources. A wiki can also be used to consolidate group work for class presentations. A private wiki, accessed through a university course management system like Moodle (http://www.moodle.com) or separately like PBWorks (http://www.pbworks.com), is a web environment, similar to Wikipedia, where students can create and link together multiple web pages in a more controlled environment.

The Analysis portion begins with a graphical fitting of the group’s model and then a hands-on optimization of the main parameter(s) using either a local or global optimization method. It may be helpful to demonstrate examples of writing a model in a form that spreadsheets can do least squares regression: linear $Y = mX + b$, polynomial $Y = a_nX^n + ... + a_1X + a_0$, logarithmic $\ln Y = mX + b$, and a few others. See the example in the student instructions along with the following. Model $x' = a(t-b)(t-c)$ can be rewritten in the polynomial form $x' = at^2 - a(b+c)t + abc$ with $X = t$ and each parameter can be estimated through the regression constants $a = a_2, -a(b+c) = a_1, abc = a_0$. Note, this kind of regression optimizes the parameter(s) for the model differential equation instead of the solution of that equation, so there will be room to improve upon the estimate(s).

Local Optimization is a basic adaptation of the bisection method for finding roots [1] to finding a minimum. At every step, students compare the sum of square errors (SSE) on the left and right and subdivide their search interval sequentially to better estimate the parameter value which minimizes the SSE. Global Optimization is a basic global search method which evaluates the SSE at a regularly-spaced set of values for a given parameter. This set is then centered at the lowest SSE value and compressed for better precision.

After the presentations, I suggest a class discussion of the various models derived and the range of values used for parameter(s) to help wrap up the project. One of the main things I bring up is the difference between our simple simulation for the spread of the common cold and a more realistic SIR model for the spread of the disease. Note that the logistic model (model four below) can be written as $I' = aIS$ since $S + I = N$. This model assumes no recovered population ($R = 0$), which can be justified as the recovery time of the common cold is longer than the time it takes for the infection
Simulating the Spread of the Common Cold
to spread throughout this high-density residence hall. For a week-long cold and a simulation which
ends in about seven rounds, this seems realistic. It is helpful to point out that a true SIR model
would be a system of differential equations instead of just one and such systems are analyzed later
on in the course.

Supplies
The following supplies are listed for groups of four.

- Residence hall floor plan. Draw fold lines along outside walls to create a ‘u’ or ‘w’ shape (see
  example lines in Figure 1): print two per group.
- Bags of small beans (or candy pieces): N pieces per group, where N is the total number of
  rooms on the chosen floor, excluding bathrooms, stairwells, hallways, and elevators. The second
  floor in Figure 1 with N = 30 rooms is specified in the directions (N = 31 for the third floor).
- Paper clips: four per group.

Optional Supplies
The optional supplies may make it easier for the students to complete the simulation, however,
the prep time to set it up is not worth it for first time this is implemented. Instead of the cups,
students can use their hands to shake and dump the beans, and they can use pencils with erasers
instead of using dry erase markers on strips of overhead slides (that the reader would have to
pre-cut to fit the floor plan).
- Plastic cups: two per group
- Overhead slides cut into strips the width of the hall in floor plan: two strips per group
- Dry erase markers: two per group
- Dry erasers (shared): about one for every three groups

Customizations
The following suggests are offered to tailor the project as the reader sees fit.

One-Day Abridged Version
This project can be abridged to fit one 50-minute class period by doing the following. For this
version, I suggest that students be familiar with the separable method beforehand and bring 1-2
laptops per group (or meet in a computer lab).

1. Have students only do one simulation per pair.
2. Have groups record the data directly into a shared spreadsheet projected for the class to see.
3. Graph the data in front of the class. Lead a class discussion in choosing slope functions which
   agree with the shape of the derivative data. Choose a model together. I suggest leading them
towards the logistics model (model four below).
Simulating the Spread of the Common Cold

4. Graph the derivative data against the model component and add a trendline to estimate the parameter.

5. Solve the differential equation model, plug its solution into the spreadsheet using this estimate, and graph the data vs the modeled infected population.

6. Compute the SSE for this parameter and demonstrate the Local Optimization to decrease the SSE in finding a better parameter estimate. Point out how the two graphs (Data, Model) converge through this optimization process.

Floor Plan Adaptations

An example floor plan is given in Figure 1, but you are welcome to choose one from your own campus. I suggest one which is fairly straight to simplify folding.

One curiosity of infecting only the dorm rooms of the residence hall is the creation of a “safe-zone” in the hallways, stairwells, and bathrooms. This is somewhat counter-intuitive as those locations have the most traffic and would likely be the locations where a virus is picked up. One simple modification of the simulation is to consider these common areas as infected from the start (but not counting for the susceptible or infected totals). Any bean landing in them would infect the nearest susceptible room. This would speed up the infection rate. Another possible modification would be to add them to the number \( N \) of susceptible rooms and infect them when the rooms on both sides are infected. This modification is less clear for when to infect the hallways, but it can be defined as when rooms across from each other are infected.

Guided Modeling

I tend to guide groups through the final selection of their model using their data comparison graph by pointing out patterns in the data and shapes of their models. A less open-ended approach would be to offer them the following potential models (based upon initial ideas from students). Each of these slope functions can be argued to share a shape with the infected derivative from the change in infected data, but the first two do not match the dependence. Model four is the traditional logistic growth model usually attributed to capacity-limited population growth. The models are all separable, the first three are also linear, and they are ordered with increasing difficulty of their solution. Note, the solution to the fourth model requires partial fraction decomposition, while the fifth is tricky to get into an explicit form.

1. \( \frac{dx}{dt} = a \sin(bt) \)
2. \( \frac{dx}{dt} = a(t - b)(t - c) \)
3. \( \frac{dx}{dt} = a(N - x) \)
4. \( \frac{dx}{dt} = ax(N - x) \)
5. \( \frac{dx}{dt} = a \sin(bx) \)
Guided Analysis

For the initial parameter estimation in the first and fifth models, the angular frequency $b$ should be estimated using the period of oscillation measured from the Infected Derivative. This can be done either as Period = 4 (interval from zero to maximum) or Period = 2 (Interval from zero to zero). All other parameters can be estimated using the linear regression through the trend line on the graph. See the example spreadsheet for a demonstration of this regression on both a single simulation and the aggregate of all simulations together using the fourth model equation (the most popular).

The reader is welcome to encourage students to either all use Local Optimization or all use Global Optimization for consistency in grading and to reduce the demonstration of each. See the example Excel spreadsheet (1-37-T-Excel-CommonColdSpread-TeacherVersion found in the Supplemental Docs to this Modeling Scenario) for a demonstration of Local Optimization (Option A) on ten simulations and a demonstration of Global Optimization (Option B) on one. The analysis completed through either of these options is tedious, but it gives students a hands-on perspective of (crude) optimization techniques. Such quantitative data analysis is helpful in making the model more implementable. Winkel, among others, argues in [5] for having more data analysis in differential equations, especially parameter estimation. For other ways to estimate parameters in spreadsheets, I suggest [2] where Miller includes another sample Excel spreadsheet. For more rigorous optimization, students with sufficient programming experience can implement a traditional optimization method such as Newton’s method on the derivative of the error function $SSE(p) = \sum (Y_i - Y(t_i; p))^2$ [1].

Assessment

When I assign this project, I share with my students a scoring guide or rubric. The following scoring guide for grading the group projects is suggested and the reader is welcome to customize it accordingly.

Scoring Guide: 20 points


(5) Presentation contribution: Presentation is organized, engaging, and time conscious. Individual contribution evident.

Example Student Solution

We started the model derivation by brainstorming potential models in class and comparing the graphs, shown in Figure , of the data for susceptible population ($s = 30 - x$), infected population ($x$), and change in infected population (which estimated the derivative $\frac{dx}{dt}$). We ultimately chose
Figure 2. Graphical comparison of simulation data for susceptible population, infected population, and change in infected population used for model development.

Figure 3. Data from ten simulation runs, each at a different initial number of infected.

the equation
\[ \frac{dx}{dt} = ax(30 - x) \]
because it made the most sense intuitively in that, the slope (change in infected) starts shallow when there is a small number of infected, and ends shallow when there is a small number of susceptible. The slope is steepest in the middle when there is a large number of both infected and susceptible students.

To solve this differential equation, we separated the \( x \) from the \( t \). On the left side we had a denominator with \( x(30 - x) \), so we had to use partial fractions to make the left side integrable.
Eventually we came to a logarithmic growth equation for $x$, but with an arbitrary constant, $C$. To resolve the constant, we had to solve for an initial condition, $x(0) = x_0$, to get the general solution

$$x(t) = \frac{30}{1 + \left(\frac{30-x_0}{x_0}\right)e^{-30at}}.$$

Our average (growth rate) parameter value was $a = 0.0276816$ over the interval of values $[0.01695, 0.04025]$. We ended up with a poor $R^2 = 0.46$ value from the model regression (only accounting for 46% of the spread of the data), but the least-squares parameter value of $a = 0.023$ is within the range of optimized values found for all simulations. The best-fit graph in Figure with error 6.56 and worst-fit graph in Figure with error 16.04 show that our model can fit the experiment very well. Additionally, most of the plotted data in Figure had the same shape, showing a consistent slope field. We only had one noticeable outlier where the graph of the data from one simulation crossed the graph of another simulation, since no two solutions to the same differential equation (same model) should cross. In general our model worked well and matched most of our data. The majority of our discrepancies came from human error, especially because the folded paper “Brandt Hall” in Figure 1 got bent part-way through, making the walls farther away and allowing the bean residents to magically jump out of the windows to escape the infection. We infected rooms before the common areas and tended to focus on the middle section of the floor when dumping our beans so the infection spread from there.

This makes our model a good approximation for predicting future outbreaks. This model could be used to predict the rate at which the entire group will become infected based on the initial number of infected. This model could be improved by executing more trials to find a better average for the growth rate as well as a more consistent testing mechanism. Also, the model could have perhaps

**Figure 4.** Demonstration of best model fit for infected population in first simulation with SSE = 6.56 (sum of square errors).
been based on the area contaminated rather than number of people infected. Further improvement would require live test subjects such as mice to better model the rate of spread of the disease.

REFERENCES


