ME 2450 Semester Project

Objective

The goal of this team-based project is to develop a two-dimensional (2D) plant disease spread model and then use this model to design an optimal disease scouting strategy to enhance agricultural productivity. You will accomplish this using optimized search strategies from class or your own exploration. You will work with your assigned Lab 9 teams and **submit a combined report on Gradescope**.

The project will build on Lab 9's SLIRP model to include external sources from airborne spore dispersion creating a new SLIRPE model. Your new SLIRPE model will be an expanded system of 1st-order ordinary differential equations (ODEs) compared to the previous SLIRP model. Additionally, the new model will be solved over a grid of plants instead of just a single population.

References

- An introduction to function handles, *FunctionHandles.pdf**.
- Optional arguments in Matlab and Python, *OptionalArgs.pdf**.
- Lectures 13-16 and 19-21 in the course lecture notes.
- Chapters 13, 14, 15, 25.3.3, *Chapra, Numerical Methods for Engineers (7th Edition)*.

Model Description:

In labs 8 and 9 you created a program that first evolved a pathogen (disease) in a population and then add in the development of the population (e.g., the host plant). In both labs the possibility of disease introduced from an outside source was included but considered to be a constant. For many plant pathogens, disease is spread through the air. We will include this possibility so that we can break our population into individual plants who can be infected and have infections grow and then after a latent period spread to other plants. The disease cycle for a fungal plant pathogen that we are trying to recreate is depicted in Figure 1.

The basic set of equations builds on those from Lab 9 which included equations for P, P_B, S, L, I, and R and adds new equations for external source to a plant (E) and the size of the spreading population (e.g., fungal spores) F. The two new equations represent the conidiation (spore colony growth) and dispersion and deposition phases of the disease cycle. The new full set of ordinary differential equations is given by Equation (1) with the terms added since Lab 9 described below.

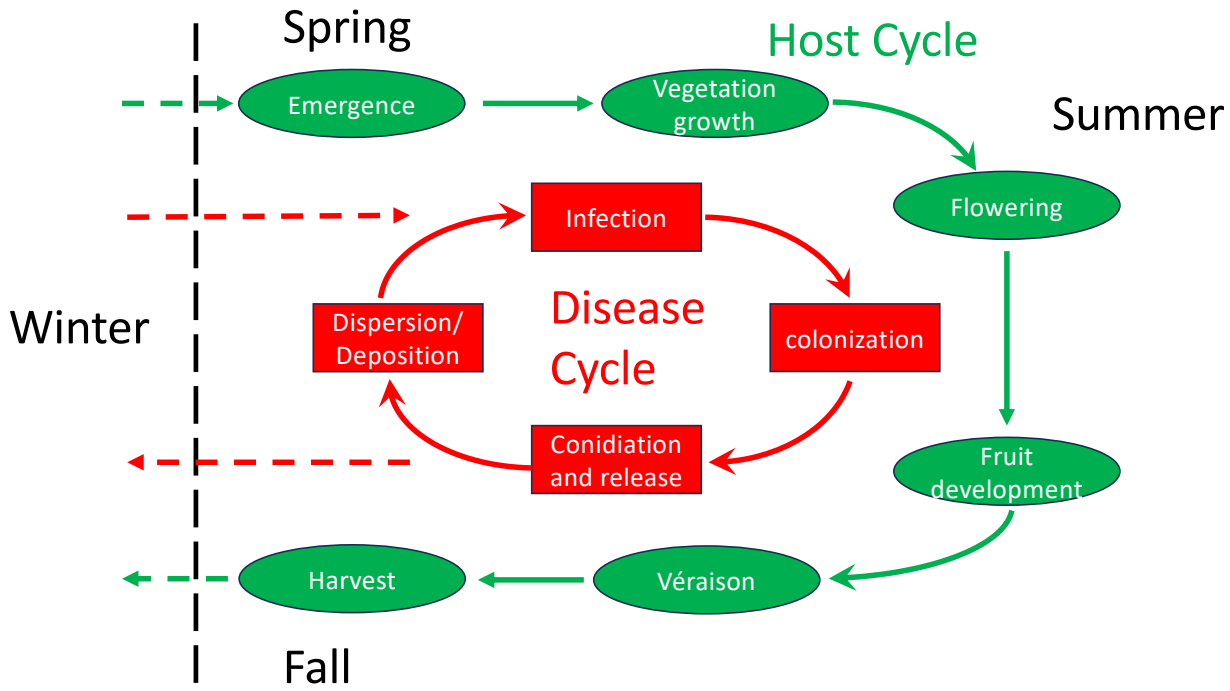

Figure 1. Basic plant and disease life cycles for an airborne plant pathogen and a fruiting plant. The disease cycle repeats throughout the growing season as the pathogen spreads among plants with spread subject to the availability of susceptible tissue through the plant host growth cycle. Our model (Eqn. 1) represents these components through the variables P and P_B for the host cycle and L (infection), I (colonization), F (conidiation and release), E (dispersion and deposition) for the disease cycle. The variable S and R track the amount of the host population available for spread.

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta SI + \frac{dP}{dt}\frac{1}{A_P} \\
\frac{dL}{dt} &= \beta SI - \mu_L^{-1}L + e \\
\frac{dI}{dt} &= \mu_L^{-1}L - \mu_I^{-1}I \\
\frac{dR}{dt} &= \mu_I^{-1}I \\
\frac{dP}{dt} &= \frac{dP_b}{dt} + \frac{dP_l}{dt} \\
\frac{dP_b}{dt} &= (0.1724P_b - 0.0000212P_b^2)T_E \\
\frac{dE}{dt} &= e \\
\frac{dF}{dt} &= \Gamma \times \exp(\alpha I A_P) - FR_f
\end{aligned}
\tag{1}
$$

Colony Growth:

Before we can estimate the distance over which the pathogen is transmitted to new host populations (e.g. to a new plant) we need to estimate the amount of material available for dispersion. We need a source strength which is represented in Equation (1) by *F*. For a passively released fungal plant pathogen (e.g., powdery mildew) this source strength is a function of the number of fungal spores (conidia) available to be transported by the air and the effectiveness of the wind in dislodging the spores so they can escape from their host plant. The rate of spore production can reasonably be taken to be a function of the size of the infected colonies (larger colonies can produce spores faster). We will assume that the colony size is proportional to the fraction of area that is infected (*I*) and use the functional form suggest by Calonnec et al. [i] (2008)

$$\text{Spore}_{\text{Prod}} = \Gamma \times \exp(\alpha I A_P) \tag{2}$$

where Γ and α are scaling coefficients chosen to fit laboratory data and A_P is the population size scale factor for an individual plant used in Lab 9. The units of Γ are spores per time and the units of α are effectively 1/m^2 and therefore A_P which is in cm^2 must be normalized appropriately to match with α (ie convert from cm^2 to m^2).

The rate that spores are released from the colony can be assumed to be largely a function of windspeed. Here we will use a form based on wind tunnel data developed by Willocquet et al.[ii] (1998)

$$\text{Spore}_{\text{escape}} = F\frac{\exp(\kappa M + \xi)}{\eta\left(1 + \exp(\kappa M + \xi)\right)} = FR_f \tag{3}$$

where κ, ξ, and η are all constants used to fit the model to the experimental data and $M = \sqrt{U^2 + V^2}$ is the windspeed magnitude where *U* and *V* are the horizontal wind vector components in the *X* and *Y* directions, respectively. The overall rate of change in spores (ie *F*) is then the difference between the production and escape terms:

$$\frac{dF}{dt} = \text{Spore}_{\text{Prod}} - \text{Spore}_{\text{escape}} = \Gamma \times \exp(\alpha I A_P) - FR_f \tag{4}$$

where R_f has been used in place of the ratio of exponentials in Equation (3) and the product FR_f is the effective release rate or source of inoculum (ie spores) from a specific host plant at a given time step.

Dispersion through the air and deposition to new hosts:

The release of fungal spores at the rate FR_f from an infected host plant that is done with its latent period (see Lab 9 for detailed descriptions of latent and infectious periods) starts the dispersion process through the air and the spread of the disease to new host plants in the simulation domain. Dispersion of spores (or any particles and pollutants) through the air is driven by turbulent wind motions. These motions are governed by Newton's 2nd law, conservation of momentum, and the conservation of mass equation. If we assume that winds are steady with no change in direction or magnitude for a given timestep and that the plant canopy has no direct impact on the wind (not an accurate assumption but a useful one) we can derive the Gaussian plume equation (Seinfeld and Pandis[iii], 2016) to describe dispersion and deposition of spores to new hosts:

$$e = \exp(-0.05X') \frac{F}{2\pi\sigma_y M} \exp\left(\frac{-Y'^2}{2\sigma_y^2}\right)(u_s d_l) \quad (5)$$

where $\sigma_y = \sqrt{\sigma_{y0}^2 + m_y^2 X'^2}$ is the width of the plume with downstream distance (how much it spread in Y') with $\sigma_{y0} = 1$ the initial width of the plume and $m_y = 1$ the rate of spread, F is our source strength from Equation (3), u_s=0.025 m/s is the settling velocity which accounts for removal of spores from gravity and impaction onto leaves, and $d_l = \sqrt{S}$ is a deposition length scale proportional to the amount of susceptible host tissue available for new infections to grow on. Equation (5) applies along the wind direction and therefore, X' and Y' are the wind aligned coordinates calculated from the U and V wind components.

The edition of Equations (4) and (5) closes the new system of equations. Like R in Labs 8 and 9, the equation for E in the Equation (1) system of ODEs is included for completeness and only tracks the amount of external disease being transmitted to a given host. It does not impact the other variables and could be skipped without impact to the overall system.

All other equations and parameters in Equation (1) are the same as the prior labs but noting that now e in our latent fraction L rate equation is now a variable calculated from Equation (5).

Numerical Methods

The set of ODEs given by Equation (1) can be integrated using the same techniques as those described in Labs 8 and 9, namely Euler's method or any Runge-Kutta scheme. The main complication is that we now have an array of host plants not a single population. This will require significant coding organizational skills.

Lab Assignment:

In this assignment you will complete a 2D disease spread code that solves the SLIRPE system of equations and then use this system to develop disease scouting routines. This will involve several steps outlined below.

1. 2D Disease Spread model

Your first step will be to create the disease spread model. The model will use a 2D grid of plants as illustrated in Figure (2). Each point is a 1m x 1m plant that grows and can be infected according to Equation (1). Variability is introduced into the system by giving each plant a random initial plant area (variable P from Equation (1)). The mean value of P denoted here as $\langle P \rangle$ is kept the same as in Lab 9, the expected value based on 30 days of growth since leaf emergence and variability is added by assigning values drawn from a normal (Gaussian) distribution with a mean value of $\langle P \rangle$ and a standard deviation of 20%$\langle P \rangle$. The initial infection point is also chosen as a random value. In this case, a random integer value corresponding to a given plant is drawn from a uniform distribution.

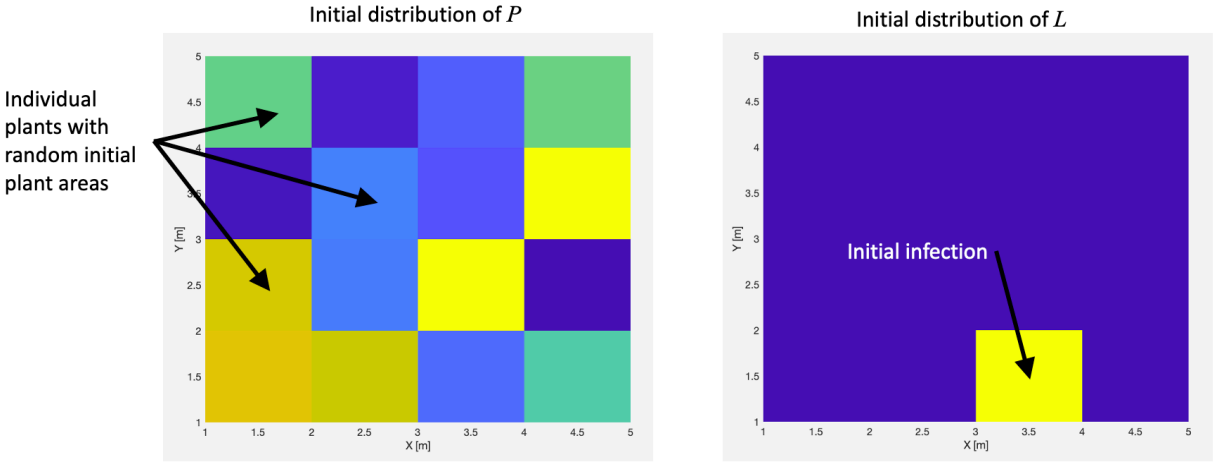

Figure 2. Example of the initial distributions of host plants (variable P in Equation (1)) in the left panel and the initial infection point in the right panel.

Once the initial plant and infection locations are proscribed, the simulation is ready to start. The basic outline (pseudo code) describing the simulation is as follows noting that we will use the specific term vine to mean a plant since this is the chosen variable name in distributed codes (and the growth parameters chosen match those of grape vines):

`Inputs:` Initial values of SLIRPE variables, Environmental forcing (U,V,T), number of vines in the X direction NpX, number of vines in the Y direction NpY, number of time steps $Nsteps$

`Outputs:` all SLIRPE variables

`For` $t = 1 \rightarrow Nsteps$

Make a list of vines that are actively spreading infection through the air (Infectious)

`For` each vine on the list call the function `GaussianPlumeDep()` to calculate the concentration of spores deposited on all other vines.

`Sum` the deposition contribution from each infectious vine at each receiver vine noting that all vines (even already infected and infectious ones) can receive new inoculum from external sources. This sum is the source (e in Equation (1)) of external disease for each vine.

`For` $cnt = 1 \rightarrow NpX \times NpY$

Check if each vine is latent and if it is the first time, calculate μ_L

`Call` `TimeInt()` to integrate SLIRPE for vine(cnt) to advance 1 time step.

Check if the vine is now infectious and if it is, switch its state

Take the SLIRPE outputs and calculate/plot statistics of our epidemic

A difference between this time integration and Labs 8 and 9 is that we are only calling `TimeInt()` for a single time step advancement. The dispersion calculation by `GaussianPlumeDep()` is by far the biggest computational cost in the simulation and so, no matter which time integration scheme we choose (Euler or Runge-Kutta) we only want to evaluate dispersion once or our computational cost will be too high for the simulation to be useful. This is common in the integration of real systems and can make it more difficult to use built in functions (e.g., Matlab's ODE45).

Note that `TimeInt()` will need an updated SLIRPE function that includes all components of

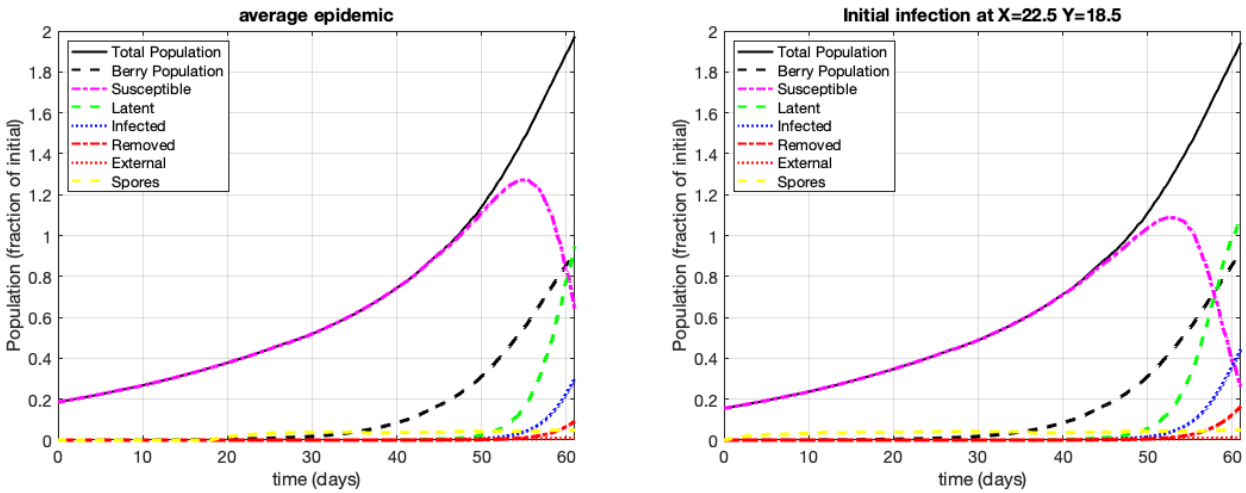

Figure 3. Example of the initial distributions of host plants (variable P in Equation (1)) in the left panel and the initial infection point in the right panel. Parameters used are those listed below in the Work Due section.

Equation (1). Example output of the average epidemic and the initially infected plant are plotted in Figure 3.

2. Scouting Routine

Our goal is to try to find the presence of disease as fast as possible to minimize the impact of any outbreak and take appropriate control measures. You will have significant flexibility in how you do this but you will need to do this efficiently. Scouting for disease (ie searching for infected plants) is time consuming and therefore costly for growers. Because of this the typical practice is to simple spray fungicides at regular intervals. This wastes resources and degrades the environment through unnecessary application of chemicals. Additionally, research has shown that overuse of specific chemistries leads to control failures as resistant strains of fungus are selected. These control failures can mean total crop loss for a grower.

Using your new simulation tool, you will come up with best ways to avoid this. You can use as many 'scouts' as you would like but you must track the total distance traveled by your scouts.

You may also use any search routine you like but your search must have the following constraints:

- The search must be completed during 1 hour of physical time (ie within 1 simulation time step).
- Scouts have a maximum speed of 0.5 m/s which will limit the number of points you can check in your search algorithm. For example, if my search is gridded and I check every other plant, I will need 4 seconds for each move of 2 m between plants and in 1 hour at maximum speed I can search 900 plants in my grid.
- An infection must be large enough to find and the size of infection you can find will depend on your scouting speed (equivalent to your scouting step size). We will use:

  $$DetectSize = \max(40 \times ScoutSpeed, 2)$$

  where detect size is in cm and is the diameter of the smallest infection a scout can identify, and scout speed is m/s (with a maximum of 0.5). The max function sets a minimum diameter of 2 cm to set a minimum detection limit. Based on this cost, I can go fast and check many points, but I won't be able to detect disease as early in an infection. Similarly, this sets a minimum effective speed beyond which going slower will not result in improved detection.
- You are using drones to scout which must charge, therefore each drone you use can only scout for 1 hour per day. Multiple may scout at the same time, but you have a maximum of 12 drones each day.
- **Scouting Cost:**
  - 1-hour of drone scouting = \$100 in cost
  - Each day after day 10 of the simulation that you do not detect the pathogen will cost \$1000

  Example: If I use 2 drones to scout each day (2 hours total) and I detect the pathogen on day 20 my total cost will be $2 \times \$100 \times 20 + (20 - 10) \times \$1{,}000 = \$14{,}000$
- **Your Goal is to minimize the scouting cost.**

3. Work Due

A. Your group's code with a working version of the epidemiological model in either Matlab or Python. You must complete the sections of the provided code labeled with comments as 'INSERT YOUR CODE HERE'. Note you can write your own code from scratch but it will still be required to work.

B. Plots of the average epidemic and the source plant disease evolution that demonstrate your code works (see Figure 3 for example). You will run your code with the following inputs for the full Environmental forcing time period (61 days):

- $NpX = 50$ the number of plants in the X direction
- $NpY = 50$ the number of plants in the Y direction
- $LpX = 1$ X-direction physical length of a plant in meters
- $LpY = 1$ Y-direction physical length of a plant in meters
- $A_P = 5000$ normalization factor for population ('final' plant surface area in cm^2)
- $P_i = 1.33 * 30 * (-0.35968 + 0.10789 * 15 - 0.00214 * 15 * 15) * 30$ initial size of the population in cm^2 (equal to the model after 30 days with constant temp of 15C)

- $\sigma_{P_i} = 20\% P_i$ the standard deviation of the initial population size
- $B_i = 1$ initial size of the berry population in cm^2 (berries grow starting the 1st day of the simulation)
- $\beta_{max} = 2$ max rate infection under ideal conditions (1/day)
- $\mu_{L,min} = 6$ minimum length of latent period (min number of days latent)
- $\mu_I = 10$ rate infection clears (number of days infectious)
- $\Gamma = 0.01$ spore production multiplier (data fit spores/day)
- $\alpha = 0.314$ spore production exponent multiplier (data fit in m^{-2})
- ξ = -2.0 release fraction offset (data fit unitless)
- η = 1 release fraction scale factor (data fit unitless)
- κ = 0.75 release fraction scale factor (data fit sec/m)
- $L_i(X_i, Y_i) = \frac{\pi 0.25^2}{4A_P}$ initial size of the population that is latent (0.25 cm diameter circle) at random initial point (X_i, Y_i)

C. A working scouting function of your choice (e.g., gridded, random, patterned, gradient). The function should take in the 2D infected population fraction field (vine variable *I*) and output a true or false for detection and the cost of the scouting event. Note you need to keep a running total of the overall cost of scouting AND delayed detection. The recommended location of this function is identified by a comment: SCOUTING FUNCTION GOES HERE. You may move the location if desired (e.g. out of the time loop using saved data) but should consider tradeoffs in doing this.

D. Optimization of your scouting routine to minimize your cost. You may do this in any way you see fit but you should optimize your chosen strategy.

E. An ensemble of 5 simulation runs with your optimized scouting routine using the parameters listed above. These 5 simulations will be your evidence of the effectiveness of your scouting strategy. You should report the maximum, minimum, and average cost for your ensemble.

F. A short-written discussion of your groups epidemic model, your scouting routine, and how you optimized your routine. Include your plots from part **B.** with proper captions and a brief description (~1/2 page max) of how the epidemic spreads in the field and in individual plants. Your discussion should also include your groups reasoning for choosing your scouting routine (~1/2 page max), a discussion and evidence (change in cost) of how you optimized your scouting routine, and a discussion of your final optimized scouting results (min, max, mean values from **D.**).

Scoring of the Assignment:

To receive credit for the lab, complete the program and submit your codes as Python or Matlab files along with a pdf containing the plots and any written discussion. **A single team report** should be submitted to Gradescope and you should identify each member of your team in your report and upon submission.

Your grade will be:

- 40% completion of the epidemic code and plotted evidence that it works
- 30% your scouting routine and scouting ensemble results
- 15% evidence of optimization of your scouting routine
- 15% your written discussion

Extra credit will be available for:

- 10% a top 5 finish in the class for lowest average cost of scouting
- 5% a 2D movie illustrating your epidemic model (example given in lecture)
- 5% the addition of your scouting routine to your epidemic movie

Submission:

Your submission to Gradescope should include at least two files (and likely more for your program). One file should be a PDF with that includes all required figures (2 minimum) with proper captions and the discussion from **F.**). Your code file (or files) should be included as a separate file(s) in its native format (e.g., .m, .py). To receive full credit, you must include both the code file (or multiple files if you save functions separately) and the PDF discussion file. To receive extra credit the appropriate files for that must also be included with your submission.

---

[i] Calonnec, A., Cartolaro, P., Naulin, J.-M., Bailey, D. and Langlais, M., 2008, 'A host-pathogen simulation model: powdery mildew of grapevine', Plant Pathology 57:493-508.

[ii] Willocquet, L., Berud, F., Raoux, L. and Clerjeau, M., 1998, 'Effects of wind, relative humidity, leaf movement and colony age on dispersal of conidia of Uncinula necator, causal agent of grape powdery mildew', Plant Pathology 47:234-242.

[iii] Seinfeld, J. H., and Pandis, S. N., 2016, 'Atmospheric chemistry and physics: from air pollution to climate change'. John Wiley & Sons.