Student Research Projects and Opportunities at a Two-Year College

Chris McCarthy
Borough of Manhattan Community College
City University of New York

SIMIODE EXPO
February 12, 2022
Virtual
Borough of Manhattan Community College

Part of the City University of New York (CUNY)
Borough of Manhattan Community College

Established: 1964

More than 27,000 students in over 45 associate degree programs

More than 10,000 students in adult and continuing education programs

Students come from over 145 countries.

Full-time Faculty: 540+ (75+ in the math Dept.)
Borough of Manhattan Community College

Established: 1964

More than 27,000 students in over 45 associate degree programs

More than 10,000 students in adult and continuing education programs

Students come from over 145 countries.

Full-time Faculty: 540+ (75+ in the math Dept.)

Main Campus Location:
199 Chambers Street, New York, NY 10007

Located in lower Manhattan on the West Side, On the Hudson River Just north of the the World Trade Center.
Borough of Manhattan Community College

Fitterman Hall, part of the BMCC main campus was damaged on 9-11 by debris from the falling towers.

It was eventually rebuilt, as was the World Trade Center.
Borough of Manhattan Community College

The new Fitterman Hall

Borough of Manhattan Community College

On-Campus Undergraduate Research Programs

• BMCC Foundation Fund for Undergraduate Research
• Collegiate Science and Technology Entry Program (CSTEP)
• CUNY Research Scholars Program (CRSP)
• Louis Stokes Allied Minority Participation (LSAMP)
• Minority Science Engineering Improvement Program -Retention and Improvements in STEM Education (MSEIP-RISE) Grant
• Science and Technology Entry Program (STEP for High School Students)
• BMCC Honors Program
BMCC One of 15 Colleges Nationwide to Win $1 Million NSF Grant for STEM Education
"If there were no dark matter, life wouldn't exist," says BMCC Professor of Science and astrophysicist Quinn Minor. He just received a National Science Foundation (NSF) award of $235,407 to study cold, or slow-moving dark matter, and explains its role in our existence.

Early stars "spit out heavier elements like silicon and iron through supernovas," Minor says, "and they spewed them out so fast, if the extra gravitational pull of dark matter hadn't been around to keep it all from escaping into intergalactic space, our earth would never have been formed."
Two afternoon poster sessions showcased 78 multi-disciplinary research projects.
The national labs and institutes are a great place for students to get a summer research experience.

Typical email from a national lab regarding student research opportunities.
RESOURCES ABOUT UNDERGRADUATE RESEARCH

Recent CURM-related articles about doing undergraduate research:

"Information for Faculty New to Undergraduate Research" by Cayla McBee and Violeta Vasilevska, INVOLVE 7:3 (2014), pp. 395-401.


"Undergraduate Research: How Do We Begin?" by Brad Bailey, Mark Budden, Michael Dorff, and Urmig Ghosh-Dastidar, published in the MAA Focus, Jan. 2009, pp. 14-16.


A great resource for ideas and projects involving differential equation models is:
What is Undergraduate Research in Math at a 2 year college????

Original research & results in deep, technical mathematics typically requires a lot of training. That is what PhD programs are for.
What is Undergraduate Research in Math at a 2 year college???

Original research & results in deep, technical mathematics typically requires a lot of training. That is what PhD programs are for.

However, undergraduates at 2 year colleges can have the "research experience".

Open ended problems with no single “correct” answer.

Read a research article or learn about the professor's ongoing research. Then reproduce\explain\help write-up the results.

Write computer code -- Data collection (experiments) – Analysis.

Modeling various phenomena. Creating, tweaking, and\or applying a model.

Exposure to grant writing, conference presentations, networking.

Discuss with professor his/her research.
What are the problems with Undergraduate Research in Math at a 2 year college????

• How to choose RA's (research assistants)? GPA? Enthusiasm? Knowing the student from previous classes?
• Research vs classwork vs job vs friends & family!
• Your research is important to YOU, but maybe not so important to your undergrad RA's.
• Students not knowing enough math. What takes you a couple of minutes to figure, might take your students ½ the semester.

• BE REALISTIC!!! Your RA's are just beginners. So...
  A good experience is more important than getting good results.
What are the benefits of Undergraduate Research in Math at a 2 year college – for students???

- Students (almost always) enjoy it.
- The students learn how to do research\open ended problems. Not just book problems.
- It helps students find out what they really want to do.
- The experience "sticks" with the students. They will remember doing research with you long after they forget all the math they learned.
- Students gain confidence, pride, a chance to show off & often get paid for it.
- The Research Experience looks great on their CV.
What are the benefits of Undergraduate Research in Math at a 2 year college – for professors???

• The professor (almost always) enjoys it. It looks good on the CV.

• Sometimes students will do useful work for the professor.

• Having students is motivating. I always feel proud of my students ☺

• I remember ALL the students I mentor.
Student research projects I've supervised

They almost all involve **modeling with differential equation.** Why?

• The students who take Diff Eq's at a 2 year college tend to be outstanding & serious & and have more mathematical maturity.

• Most of my Diff Eq students are interested in engineering or science. They realize the need to understand or be familiar with modeling.

• Students can use their physical intuition to understand what should happen mathematically. They might not understand the math, but they can understand what we are trying to model.

• Most of the students aren't ready to do research in "pure" math. They haven't had analysis, abstract algebra, topology, etc.
ODE Model of Adsorption Based Water Filters

Senayit Menasche and Abdulai Jalloh

ABSTRACT

We present a simple mathematical model which can predict the response of adsorption based column filters. In our lab we have applied this model to column filters which we have constructed out of spent tea leaves. The filters are able to remove heavy metals from water at the rates predicted by our model.

INTRODUCTION

Our lab has been conducting research into the biocommodation of environmental pollutants. One project involves constructing filters out of organic waste materials [1, 2]. When heavy metal contaminated water comes into contact with the tea leaves, the heavy metal ions have an affinity for "functional groups" (i.e., binding sites) expressed on the surface of the leaves and bind to them. As a result, it is possible to construct filters out of spent tea leaves which can remove heavy metals, such as copper, zinc, and cobalt from water [3, 4]. In this paper we develop and use a simple model to predict the behavior of such filters.

BACKGROUND INFORMATION

Heavy metal water pollution has become a challenging issue for many regions across the globe (Figures 1 and 2.) The presence of heavy metals in water can cause serious health effects for example, reduced growth and development, cancer, organ damage, nervous system damage, and even death. For this reason, the removal of heavy metals is a critical environmental issue. It is important for researchers to find economical and effective methods for heavy metal removal.

FILTERING MODEL (CONCEPTUAL)

A filter modeled as a one dimension strip with S binding sites (Figure 3). Particles bind to a site with probability p and don’t bind with probability q = 1 - p. As a pollutant unit is carried by the water through the filter it has the potential to interact with, on average, S binding sites. For each binding site there is probability q that the pollutant unit will stick to that binding site, and probability q = 1 - p that it won’t stick.

FILTERING MODEL (USABLE)

As a pollutant unit is carried by the water through the filter it has the potential to interact with, on average, S binding sites. For each binding site there is probability q that the pollutant unit will stick to that binding site, and probability q = 1 - p that it won’t stick.

REFERENCES

Marvin Villalba’s Honors Project becomes part of my web page

ONLINE https://mccarthymat501.commons.gc.cuny.edu/newtonian-cooling/
Students copy and modify the R script. They run it on online (RexTester.com) or on their computer.
Open ended modeling question for students

Modify Newton's model to account for the varying room temperature.

This increase in temperature is due to sun moving across the window.
Funding Acknowledgements:
NYS OER Scale Up Initiative & CUNY

CUNY (City University of New York) was awarded $4,000,000 from New York State to establish, sustain, and enhance new and ongoing OER initiatives throughout CUNY (FY 2018). The expected result will be large-scale course conversions throughout the university.

BMCC Librarian Professor Jean Amaral
OER Warrior Extraordinaire
2018 - 2019 CRSP
Sources for Virtual Experiments
NetLogo Simulation & Programming Environment
Chemotaxis Sim
McCarthy & Watts (2019)
Predators (red) find their prey (yellow) via chemoattractants (blue).

Virtual Experiments
Chris McCarthy
cmccarthy@bmcc.cuny.edu
Active Matter: Chemotaxis

Gianni Watts, Adama Sene, Jorwyn Medina, Muhammad Hannan
Mentor: Professor Chris McCarthy (Mathematics) cmcCarthy@bmcc.cuny.edu
City University of New York, Borough Of Manhattan Community College

ACTIVE MATTER

Active matter research focuses on the paradigm of emergence. Simple rules can lead to complex behavior: schools of fish, swarms of insects, self-assembly of macromolecules. Organisms organizing themselves without top-down commands, e.g. the flocking of birds [1, 2].

CHEMOTAXIS

Chemotaxis is when an organism’s motion is effected by a chemical gradient. If the organism moves in the direction of gradient, the chemical is called a chemoattractant [3]. Example: neutrophils (white blood cells) hunting down bacteria (pathogens).

METHODS

We wrote and ran the simulations using NetLogo’s “Behavior Space” feature. The data from the simulations were saved as csv files (Excel spreadsheet) and then imported into the statistical package R for analysis by a custom R script we wrote.

RESULTS (for Diffusion Rates)

In Figure 1 we see that the shape of the Diffusion Rate vs Average Extinction Time graph seems to decrease asymptotically to about 250 time steps. However, a more detailed simulation, Figure 2, shows that increasing the diffusion rate beyond 0.3 results in taking longer for the predators to capture the prey. Figure 2 required 10,000 simulations (500 simulations at 20 different diffusion rates).

RESULTS (for Decay Rates and Prey Numbers)

In Figure 3 we see that the shape of the Decay Rate vs Average Extinction Time graph seems to increase in a sort of Y shape, with a minimum of 225 time steps to extinction when the decay rate is 0.4. If the decay rate is close to zero, the chemotactic effect is zero, and the predator is misled by chemoattractant remnants. If the decay rate is close to 1, the chemotactic effect is too quick to be of use to the predator.

ACKNOWLEDGMENTS

BMCC Provost Erwin J. Wong, BMCC Director of Research Dr. Helene Barth. Funding: BMCC Provost Erwin J. Wong, City University of New York Research Scholars Program (CRSP).

REFERENCES


FUTURE RESEARCH

In the future, we hope to accomplish
1. Understanding the uptake in time to extinction when the diffusion and decay rates approach 0 or 1. See Figures 1, 2, and 3.
2. Creating mathematical models that allow us to predict the behavior of the simulations.
3. Designing more life-like simulations. For example, where both species reproduce and die, where species are more biologically accurate.

2018 – 2019

CRSP

Students Presented at various conferences including the 2019 Joint Mathematics Meetings in Baltimore
Active Matter: Predator Prey Interactions

Jorwyn Medina, Muhammad Hannan, Adama Sene
Mentor: Professor Chris McCarthy
BMCC Mathematics

Background
Active matter is composed of large numbers of active "agents", each of which consumes energy (e.g., by eating). The consumption of energy allows these systems to be out of thermal equilibrium (and their members to stay "alive"). An example of energy consumption is when a predator eats its prey.

The predator-prey relationship is the base of the food chain. When there are large amounts of prey, the amount of predators can increase. This in turn causes the amount of prey to decrease, which then causes the amount of predators to decrease, which then causes the amount of prey to increase. A mathematical model of this predator-prey relationship is called the Lotka Volterra model [1,2].

Lotka Volterra

The Lotka Volterra predator-prey equations are a pair of nonlinear first order differential equations that describe the interaction over time of a prey species (s for sheep) and a predator species (w for wolves):

\[ \frac{ds}{dt} = aK - bs \]
\[ \frac{dw}{dt} = cw - ds \]

Where a, b, c, d, e, f, h are positive constants.

The graph shown above shows the wolf, sheep and grass splined functions being compared to the solution of the non linear model Lotka Volterra. K is the carrying capacity.

1. Stability

Our Lotka Volterra system of ODE's has an equilibrium point where \( g(x)=0 \) and \( h(x)=0 \). An interesting question is whether that equilibrium point is stable meaning if we perturb the system from equilibrium, will it return to equilibrium? Since the real parts of all eigenvalues of the Jacobian matrix are negative, the answer is YES! This was figured out by taking inside a matrix J the partial derivatives of all variables. Then, we calculate det(J)=0.

Netlogo and Matlab

1. Using Netlogo we simulated Lotka Volterra predator prey system. The Wolf-Sheep simulation we used was created by U. Wilensky [4].

2. We export the wolf, sheep and grass population data to a spreadsheet. We then import the data into MATLAB. Below, we plotted the regular data taken from Netlogo simulation. In order to have better approximation for the differentiation, we use the caspas function to smother the data. However, plot the smoothed data require it first into a function using the fractal function.

Research

Our research includes coding, creating and running agent based simulations, modeling them with differential equations and developing tools (in matlab) to fit the models to the data from these simulations.

Some M-Code Snippets

If import data from excel using importdata from the home tab and then import it into the command window. To access its data, do the following:

- Assign important files and the code:
  ```matlab
  ...% Importing important files
  %
  ...%
  
  - Smoothing: create spline object and trim it into a function.
  ```

- In order to plot it, we have to turn the spline object into a function using the final function:

- We estimate parameters using linear regression. We then use Ranche-Kutta to numerically solve the Lotka Volterra system with the parameters a,b,c,d,e,f and h found above. The code looks like this:

Future Research

1. Further improve the algorithm to estimate the parameters.
2. Understand the changes in our parameters.

References


Students Presented at various conferences including the 2019 Joint Mathematics Meetings in Baltimore
Differential Equations Model And Resource Creators Workshop

View from Mt. Hood, Oregon

SIMIODE DEMARC Workshop
George Fox University
Oregon, July 2019
DEMARC Goal

To Develop Diff Eq Modeling Projects

(That are good for students)

I developed a modeling project involving Euler's Method and drag (air resistance)

The drag on a ball
Heuristic argument: drag force proportional to $v^2$

\[ F = ma = \frac{d}{dt} mv \approx \frac{\Delta mv}{\Delta t} \]

\[ \rho = \text{density of air.} \quad v = \text{velocity of projectile} \]

\[ A = \text{cross sectional area.} \quad \Delta x = v\Delta t \]

Mass of air collided with in $\Delta t = \rho \frac{A\Delta x}{\text{volume}}$

\[ \Delta v_{air} \propto v_{\text{projectile}} = v \]

\[ F_{\text{drag}} \propto \frac{\rho A \Delta x \Delta v_{air}}{\Delta t} \approx \rho A v v = \rho A v^2 \]

\[ F_{\text{drag}} \text{ is in opposite direction of } v. \]

Drag Equation

\[ F_{\text{drag}} = \frac{1}{2} C_D \rho A v^2 \]

$C_D = \text{drag coefficient}$
Euler recursive relation including drag

\[
\begin{pmatrix}
    x \\
    y \\
    v_x \\
    v_y \\
    t
\end{pmatrix}
_{n+1}
= 
\begin{pmatrix}
    0 \\
    2 \\
    12 \cos \theta \\
    12 \sin \theta \\
    0
\end{pmatrix}
_{0}

\begin{pmatrix}
    x \\
    y \\
    v_x \\
    v_y \\
    t
\end{pmatrix}
_{n}
+ 
\begin{pmatrix}
    -\frac{c}{m} \sqrt{v_x^2 + v_y^2} \\
    v_x \\
    -g - \frac{c}{m} \sqrt{v_x^2 + v_y^2} \\
    v_y \\
    1
\end{pmatrix}
_{n}
\Delta t

Initial conditions
Position 2 meters up
Speed 12 m/s
Launch angle \( \theta \) varies
My honors student Kujtim Bardhyll worked with me to test the drag model on a real pendulum.
From Kutjim’s presentation

- I used **Tracker Video Analysis and Modeling Tool** from Open Source Physics to plot the points of the tennis ball.
- This app tracks objects in motion. It helped me see the oscillation points of the pendulum.
- These points are helpful because they use real time tracked data points against the calculations made in python.
- It creates a graph of the points showing the user where they are on the x and y axis.
11.5 in fishing line with weights / tracked
The pendulum data from the Tracker software was imported into Python where it was combined with our ODE model, which was solved using Euler's method.

From Kutjim’s presentation
No added weight

With added weight

\[ T \approx 2\pi \sqrt{\frac{L}{g}} \]

From Kutjim’s presentation
Using Machine Learning to Recognizing Graphs and Functions

Ziqi Polimeros, Borelle Fabrice Tene
Mentor: Professor Chris McCarthy, Borough of Manhattan Community College

Using machine learning we can create neural nets which can accurately distinguish computer and hand drawn images of mathematical functions.

It takes about 5 minutes depending of what kind of computer you are using, to train the neural net to recognize 7 function classes if we use about 150 images. Once trained, the neural net will almost instantly correctly categorize the input image of a function (if it is of one of the 7 types).

Figure 12: Hand drawn image of a line with negative slope
Figure 13: Best fitting (regressed) line (black) is superimposed on the hand drawn line from Figure 12 (now colored red).
Figure 14: Hand drawn image of a parabola concave down
Figure 15: Best fitting (regressed) parabola (black) is superimposed on the hand drawn parabola from Figure 14 (now colored red).

Python function that produced the superposed images

Figures 6 - 15 are of the hand drawn function (left) which we input to our Python program. On the right, is the output of our Python program: the name of the function type, together with the best fitting curve of that type (in black), found by regression, and superimposed over the original hand drawn image (in red).

Figure 17 shows the set up for training our neural net. We used a mixture of computer and hand drawn images of functions, see Figure 17 (b). The more training data, especially training data that is similar to the images to be categorized, the better the accuracy in categorization.

On our computer, the training images are organized in a certain way. Each training image needs to be in its appropriate category folder. We had 7 folders, see Figure 17 (a). Figure 17 (b) shows what is inside one of those folders. We put a minimum of 20 different training images in each folder.

Python function that superposed images

The model summary tells us about the layers in our convolution neural net. For example, the first two layers are convolution layers. Convolution layers look for features like edges and lines in the image which will help to identify the image. Then there are other layers which serve to pool or combine data to reduce the complexity or size of the model.

Then there are layers which work to connect the features by flattening the previous layer, e.g. in the Flatten layer we have $62 \times 62 \times 64 = 246016$. The final layer has size 7 because of the seven function types.

Using machine learning we can create neural nets which can accurately distinguish computer and hand drawn images of graphs of mathematical functions.

References

15. [https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-4df22d3f]
17. [https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-4df22d3f]
Using Machine Learning to Recognizing Graphs and Functions

Zui Polimeros, Borrelle Fabrice Tene
Mentor: Professor Chris McCarthy, Borough of Manhattan Community College

Abstract

Machine learning has been applied successfully to many fields and is increasingly used in computational science and engineering. One area of this project is to build and train neural networks that can distinguish images of mathematical graphs. The resulting language used is Python, on its own, and with the help of machine learning packages such as TensorFlow (for Google and Keras). So far, the neural net is capable of distinguishing certain aspects of each graph, such as vertices and edges. The next step is to recognize the data and use it to train our network.

Results: recognizing computer drawn graphs

We used a combination of computer rendered and hand-drawn images of functions. Ideally, the machine learning techniques shown in the pattern recognition network can be combined with the graphing capabilities. With these features, the neural network can identify specific areas in the image and use that information to create a representation of the function. For example, you can input hand-drawn images of a function, the application will recognize if it is concave up or down, and then superimpose the hand-drawn image on the result of applying the function. The advantage of this approach is that it is flexible and can be applied to a wide variety of images.
In textbook treatments of solving the cable equation, symmetry or some knowledge of where the low point of the cable will be, is used to solve for the horizontal tension $T_0$.

The following method solves a more general case.

\begin{align*}
  w(x) &= u d g(f_1(x) - f_0(x)) \\
  u &= \text{density (kg/m}^2) \\
  d &= \text{thickness (m)} \\
  g &= \text{acceleration of gravity 9.8 (m/s}^2) \\
  \frac{d^2 y}{dx^2} &= \frac{w(x)}{T_0} \\
\end{align*}

1. Integrate $w(x)$ twice
2. $y(x) = \int \int \frac{w(x)}{T_0} \ dx \ dx + c_1 x + c_0$
3. Using BC (tower attachment heights) solve for $c_1$ and $c_0$ in terms of $T_0$
4. For each tension $T_0$, (numerically) find minimum of $y(x), x \in [0, s]$.

Call this function $myT(T_0)$. Note. $myT(T_0)$ is monotonically increasing.

5. Find $T_0$ so cable low point $myT(T_0)$ is at desired height. (Newton’s Method)
6. Find $x$ coord of low point of cable. Newton or any lazy algorithm as $y(x)$ is concave up.
7. Calculate where suspender cables are attached.
Bridges with unusual geometries

In textbook treatments of solving the cable equation, symmetry or some knowledge of where the low point of the cable will be, is used to solve for the horizontal tension $T_0$.

The following method solves a more general case.

$$w(x) = udg(f_1(x) - f_0(x))$$
$$u = \text{density} \ (kg/m^2)$$
$$d = \text{thickness} \ (m)$$
$$g = \text{acceleration of gravity} \ 9.8 \ (m/s^2)$$

$$\frac{d^2y}{dx^2} = \frac{w(x)}{T_0}$$

1. Integrate $w(x)$ twice
2. $y(x) = \int \int \frac{w(x)}{T_0} \ dx \ dx + c_1 x + c_0$
3. Using BC (tower attachment heights) solve for $c_1$ and $c_0$ in terms of $T_0$
4. For each tension $T_0$, (numerically) find minimum of $y(x), x \in [0, s]$.

Call this function $myT(T_0)$. Note. $myT(T_0)$ is monotonically increasing.

5. Find $T_0$ so cable low point $myT(T_0)$ is at desired height. (Newton’s Method)
6. Find $x$ coord of low point of cable. Newton or any lazy algorithm as $y(x)$ is concave up.
7. Calculate where suspender cables are attached.
Bridges with unusual geometries

In textbook treatments of solving the cable equation, symmetry or some knowledge of where the low point of the cable will be, is used to solve for the horizontal tension $T_0$.

The following method solves a more general case.

$$w(x) = udg(f_1(x) - f_0(x))$$

$u =$ density $(kg/m^2)$

$d =$ thickness $(m)$

$g =$ acceleration of gravity $9.8 \ (m/s^2)$

$$\frac{d^2y}{dx^2} = \frac{w(x)}{T_0}$$

(1) integrate $w(x)$ twice

(2) $y(x) = \int \int \frac{w(x)}{T_0} \ dx \ dx + c_1x + c_0$

(3) Using BC (tower attachment heights) solve for $c_1$ and $c_0$ in terms of $T_0$

(4) For each tension $T_0$, (numerically) find minimum of $y(x), x \in [0, s]$.

Call this function $myT(T_0)$. Note. $myT(T_0)$ is monotonically increasing.

(5) Find $T_0$ so cable low point $myT(T_0)$ is at desired height. (Newton’s Method)

(6) Find $x$ coord of low point of cable. Newton or any lazy algorithm as $y(x)$ is concave up.

(7) Calculate where suspender cables are attached.
Bridges with unusual geometries
CAD and Building It

Spring 2021

Michael L.
(CRSP mentee 2020 – 2021)

FreeCad Model Builder
Extraordinaire
Michael L. (CRSP mentee 2020 – 2021)
FreeCad Model Builder Extraordinaire
Growth and diffusion of bacteria in a thin pipe. The bacteria randomly diffuse and replicate. At each location along the length of the pipe the carrying capacity is $K$. At the ends of the pipe are antibiotics which kill any bacteria that reach the ends.
Growth and diffusion of bacteria in a thin pipe. The bacteria randomly diffuse and replicate. At each location along the length of the pipe the carrying capacity is $K$. At the ends of the pipe are antibiotics which kill any bacteria that reach the ends.

We combine the diffusion PDE

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

with the logistic growth model (the reaction term)

$$\frac{dc}{dt} = r_0 c \left(1 - \frac{c}{K}\right)$$

To get our Reaction Diffusion Equation:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right)$$

**Boundary Conditions**

$$c(0, t) = c(L, t) = 0$$

$c(x, t)$ = concentration of bacteria

$x$ = position in tube

$t$ = time

$D$ = diffusivity constant

$K$ = concentration carrying capacity

$r_0$ = instantaneous relative growth rate at low concentrations
Reaction Diffusion Model

Numerical Solution Euler’s Method

\[ c(x, t + \Delta t) \approx c(x, t) + \frac{\partial c}{\partial t}(x, t) \Delta t \]

\[ \frac{\partial^2 c}{\partial x^2} \approx \frac{\partial c}{\partial x}(x + \Delta x, t) - \frac{\partial c}{\partial x}(x, t) \frac{\Delta x}{\Delta x} \]

We approximate \( \frac{\partial^2 c}{\partial x^2} \) using Finite Differences
Reaction Diffusion Model

Numerical Solution Euler’s Method

\[
c(x, t + \Delta t) \approx c(x, t) + \frac{\partial c}{\partial t}(x, t) \Delta t
\]

\[
c(x, t+\Delta t) \approx c(x, t)+ \left( D \left( \frac{c(x+\Delta x, t) - 2c(x, t) + c(x-\Delta x, t)}{(\Delta x)^2} \right) + r_0 c(x, t) \left( 1 - \frac{c(x, t)}{K} \right) \right) \Delta t
\]

\[
c(x_i, t_{j+1}) = c(x_i, t_{j}) + \left( D \left( \frac{c(x_{i+1}, t_{j}) - 2c(x_i, t_{j}) + c(x_{i-1}, t_{j})}{(\Delta x)^2} \right) + r_0 c(x_i, t_{j}) \left( 1 - \frac{c(x_i, t_{j})}{K} \right) \right) \Delta t
\]

\[
x_{i+1} = x_i + \Delta x
\]
\[
x_{i-1} = x_i - \Delta x
\]
\[
t_{j+1} = t_j + \Delta t
\]
Reaction Diffusion Model

Numerical Solution Euler’s Method

\[ x_{i+1} = x_i + \Delta x \]
\[ x_{i-1} = x_i - \Delta x \]
\[ t_{j+1} = t_j + \Delta t \]

\[ c(x_i, t_{j+1}) = c(x_i, t_j) + \left( D \frac{c(x_{i+1}, t_j) - 2c(x_i, t_j) + c(x_{i-1}, t_j)}{(\Delta x)^2} + r_0 c(x_i, t_j) \left( 1 - \frac{c(x_i, t_j)}{K} \right) \right) \Delta t \]

This finite difference numerical method works provided the Courant number:

\[ D \frac{\Delta t}{(\Delta x)^2} \leq 0.5 \]

For details see convergence of numerical solutions of the diffusion equation. E.g., Section 3.2 of “Numerical Methods of PDE’s” by Seongjai Kim.
Reaction Diffusion Equation

Solution

Tube of length $L = 1$.

$K = 10$ = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

\[
\frac{\partial c}{\partial t} = \underbrace{D \frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + r_0 c \left(1 - \frac{c}{K}\right) \quad \underbrace{\text{reaction term}}_{\text{convergent Courant number} = 0.49791 < 0.5}
\]
Reaction Diffusion Equation

Solution

Tube of length $L = 1$.

$K = 10 = \text{carrying capacity}$

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

\[
\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right)
\]

diffusion term

reaction term

"convergent" Courant number $= 0.49791 < 0.5$
Reaction Diffusion Equation

\[
\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left( 1 - \frac{c}{K} \right)
\]

Diffusion term \quad \text{reaction term}

Tube of length \(L = 1\).

K = 10 = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

“NOT convergent” Courant number = 0.50037 > 0.5

\(K = 10 \quad D = 10.15 \quad D \Delta t / (\Delta x)^2 = 0.50037\)
Reaction Diffusion Equation
Solution

\[ \frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left( 1 - \frac{c}{K} \right) \]

- diffusion term
- reaction term

Tube of length \( L = 1 \).
- \( K = 10 \) = carrying capacity
- Blue = initial concentration of bacteria
- Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

"NOT convergent" Courant number = 0.50037 > 0.5
Larger D = faster diffusion. Bacteria diffuse to deadly pipe ends, from safety of pipe’s interior, more quickly.

“convergent” Courant number = 0.49928 < 0.5

\[ \frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right) \]

(diffusion term) (reaction term)

**Reaction Diffusion Equation**

**Solution**

Tube of length L = 1.

K = 10 = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method
Larger $D$ = faster diffusion. Bacteria diffuse to deadly pipe ends, from safety of pipe's interior, more quickly. "convergent" Courant number = 0.49928 < 0.5

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right)$$

diffusion term

reaction term

Reaction Diffusion Equation
Solution

Tube of length $L = 1$.

$K = 10$ = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method
Larger $D$ = faster diffusion. Bacteria diffuse to deadly pipe ends, from safety of pipe’s interior, more quickly.

"NOT convergent" Courant number = 0.50245 > 0.5

\[
\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right)
\]

**diffusion term**  \(\frac{D \Delta t}{(\Delta x)^2} = 0.50245\)

**reaction term**

**Reaction Diffusion Equation**

Solution

Tube of length $L = 1$.

$K = 10$ = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method
Larger $D$ = faster diffusion. Bacteria diffuse to deadly pipe ends, from safety of pipe’s interior, more quickly.

“NOT convergent” Courant number = 0.50245 $> 0.5$

\[
\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + r_0 c \left(1 - \frac{c}{K}\right)
\]

**Diffusion term** \hspace{1cm} **Reaction term**

**Reaction Diffusion Equation Solution**

Tube of length $L = 1$.

$K = 10$ = carrying capacity

Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method
Reaction Diffusion Partial Differential Equations are used to model a variety of phenomena in biology, chemistry, and physics. We use a reaction diffusion equation to model bacteria in a thin tube that has antibiotics at both ends. The bacteria diffuse and replicate in the tube. When they reach the ends of the tube they die due to the antibiotics.

The diffusion PDE:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

is a consequence of Fick’s law which states that particles diffuse from higher to lower concentration.

We combine the diffusion equation with the differential equation for the logistic growth model for bacteria (the reaction term):

$$\frac{dc}{dt} = \frac{r_0}{K} c(K - c);$$

to get our reaction diffusion equation.

$$\frac{dc}{dt} = \frac{\partial^2 c}{\partial x^2} + \frac{r_0}{K} c(K - c)$$

Boundary Conditions:

$$u(0, t) = u(L, t) = 0$$

is the logistic growth model for bacteria. Bacteria are killed by antibiotics at the ends of tube $x = 0$ and $x = L$.

$c = \text{concentration}$

$x = \text{position}$

$t = \text{time}$

$D = \text{diffusivity constant}$

$K = \text{concentration carrying capacity}$

$r_0 = \text{instantaneous relative growth rate at low concentrations}$

We numerically solve this reaction diffusion equation and use it to analyze the diffusion and concentration of bacteria in a tube with antibiotics at both ends.

**Background**

Diffusion is the movement of particles from a region of higher concentration to lower concentration. Mathematically diffusion occurs in response to a concentration gradient. In this figure the slope (gradient) is negative. So, the net flow of particles is in the positive direction.

**Method and Results**

Below is the Matlab code we developed to numerically solve the reaction diffusion equation. It makes use of the Euler (Finite Difference) method. On the right is the final frame from the animation produced by the code. An initial concentration is shown in dark blue. Eventually the distribution takes an upside down U shape.

```
plotNum = 20000;
C(:,plotNum);
max(C(:,plotNum));
min(C(:,plotNum));
figure(1);
hold on
plot(x(C(:,1)),',', 'LineWidth', 4);
plot(x(C(plotNum,:)),',', 'LineWidth', 5);
xlabel('x');
ylabel('c(x,t)');
axis tight manual
set(gca, 'nextplot', 'replacechildren');
V=VideoWriter('RD1.avi');
open(v);
for k = 1:200:plotNum
hold on
plot(x,C(k,:))
frame = getframe(gcf);
writeVideo(v,frame);
M(k) = getframe;
end
```

**Discussion And Conclusion**

We were able to solve the reaction diffusion equation numerically and create an animation showing the concentration of bacteria over time. Regardless of the initial conditions we chose (dark blue curve), in the end, the bacteria concentration would take the form of an upside-down U shape. In the above figure $L = 1$ and most of the bacteria are concentrated between $x=0.2$ to $x=0.8$. This is due to the bacteria at the ends of the tube being killed by the antibiotics.

**Acknowledgements**

Special thanks to BMCC Foundation Fund for the funding and my mentor Chris McCarthy.
Professor Chris McCarthy
BMCC CUNY
cmccarthy@bmcc.cuny.edu

Please feel free to contact me! Thanks!!!