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

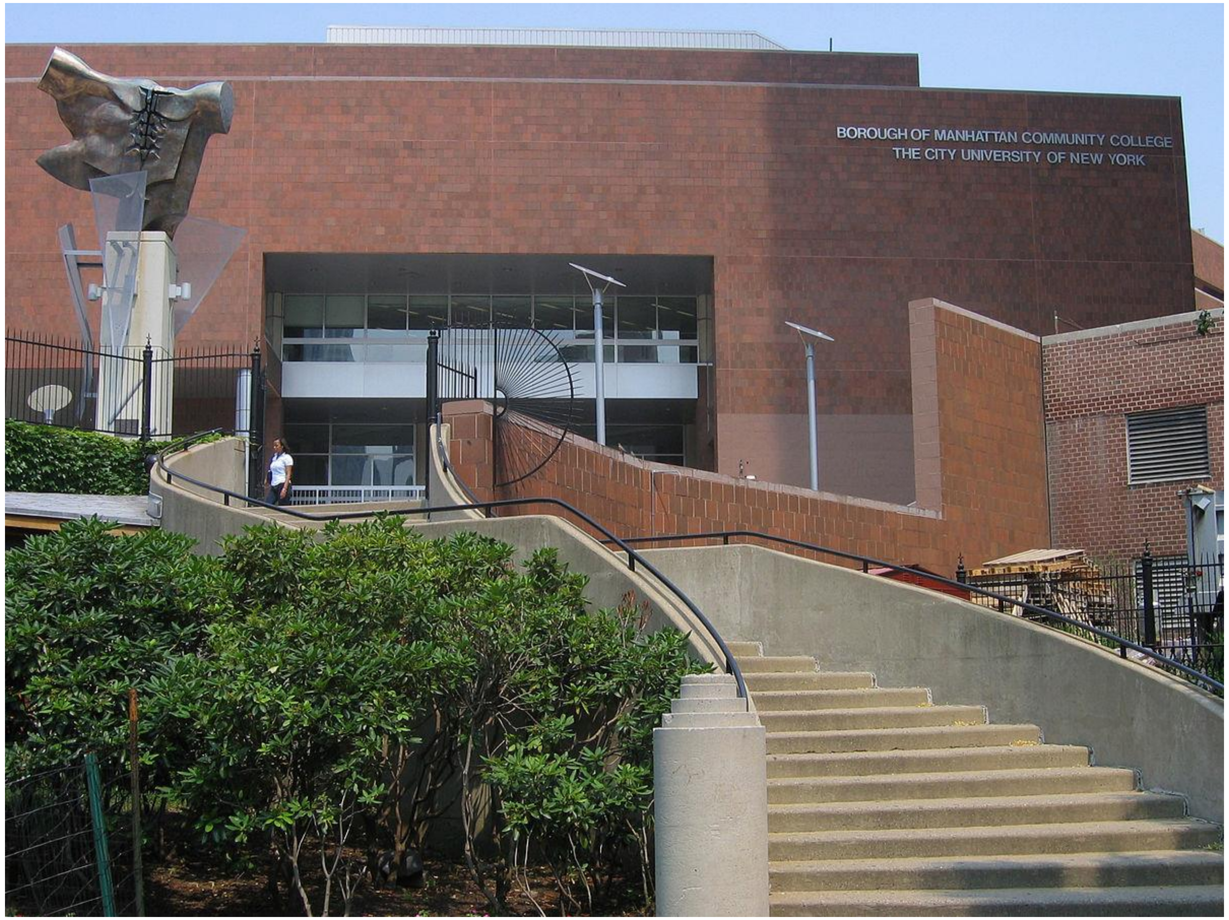
Virtual

City University of New York

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.)

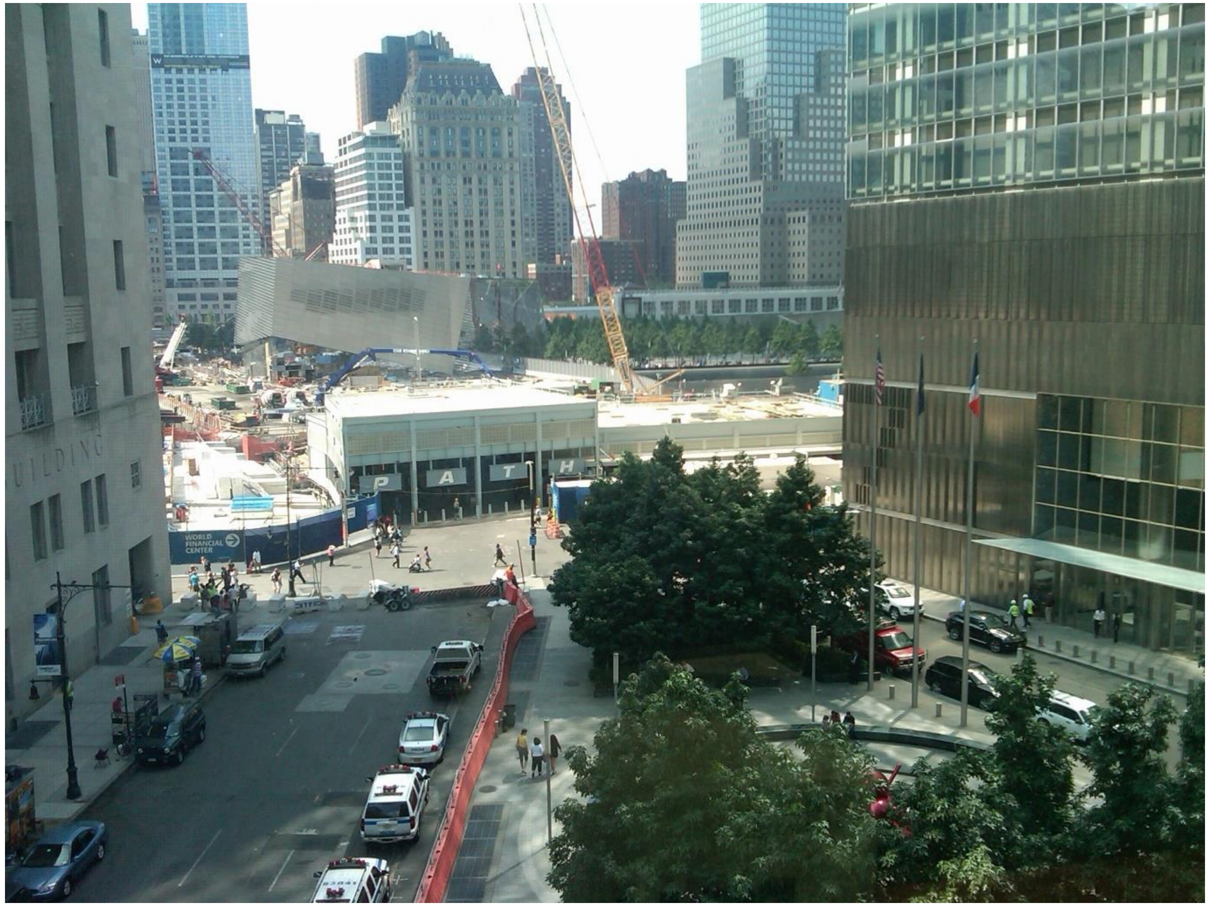
Main Campus Location:
199 Chambers Street, New York, NY 10007

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

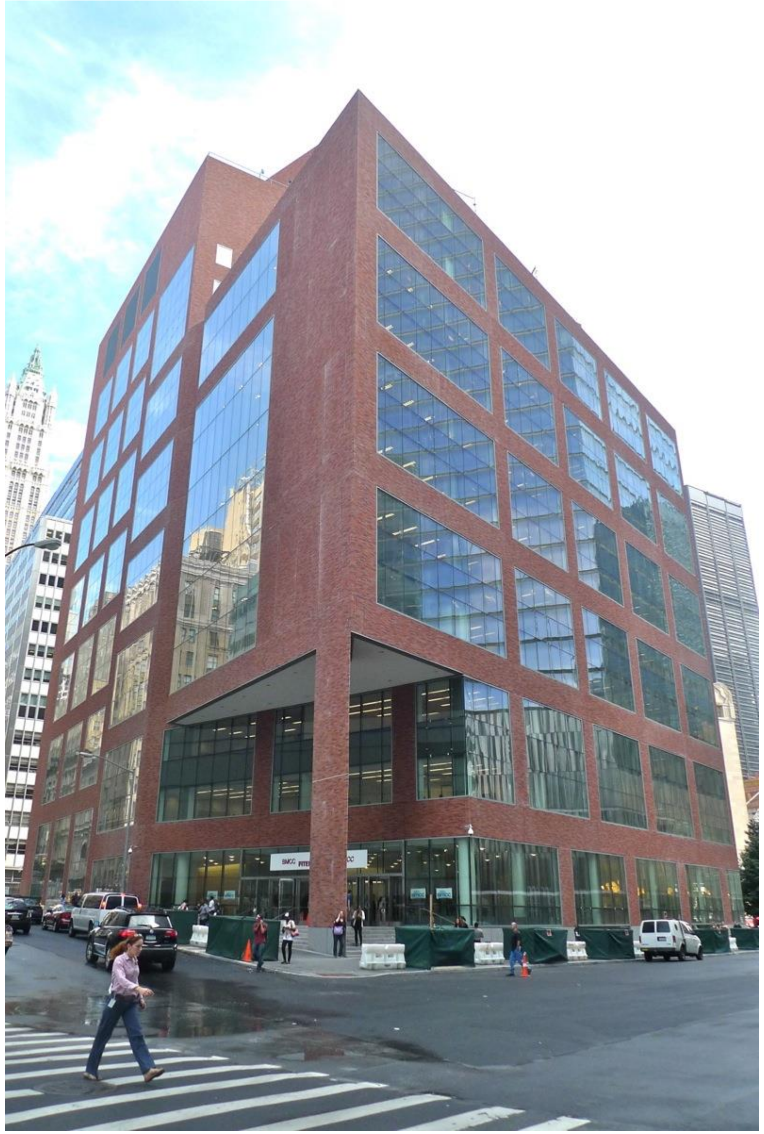
Borough of Manhattan Community College

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

The Freedom Tower (World Trade Complex) from the steps of BMCC on 9-11-2017.

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

Borough of Manhattan Community College

Borough of Manhattan Community College

BMCC Receives $230,407 from NSF for Research on Dark Matter

BMCC Professor of Science Quinn Minor

SEPTEMBER 14, 2016

"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."

STORY HIGHLIGHTS

BMCC Professor of Science Quinn Minor receives National Science Foundation (NSF) award of $235,407 to study dark matter

Funded through NSF's Division of Astronomical Sciences (AST), the project runs September 1, 2016 through August 31, 2019

Six students will receive stipends to examine computer data, present papers and more

BMCC Annual Research Symposium (BARS)

Non-BMCC Research Opportunities

Nuclear Engineering Science Laboratory Synthesis Programs ... - Chris Mccarthy https://email.bmcc.cuny.edu/owa/#viewmodel=ReadMessageItem&ItemID=AAMkADE4O...

Nuclear Engineering Science Laboratory Synthesis Programs at ORNL- Spring or Summer 2018

Science Education Programs <scienceeducationprog@orau.org>

Fri 1/5/2018 8:16 AM

To:Chris Mccarthy <cmccarthy@bmcc.cuny.edu>;

Student and Alumni Research and Technical Opportunities at Oak Ridge National Laboratory (ORNL) – Oak Ridge, TN

Appointments for Spring and Summer 2018!

Apply NOW to the Nuclear Engineering Science Laboratory Synthesis Programs (NESLS) Program at Oak Ridge National Laboratory (ORNL) – Spring or Summer 2018

Must apply at https://www.zintellect.com/Posting/Details/3645
by January 6, 2018 for Spring term

Must apply at https://www.zintellect.com/Posting/Details/3685
by February 28, 2018 for Summer term (must start by June 15 and end on or after August 10, 2018)

- Current AAS, BS, MS, and PhD students – Majors related to Engineering, Earth and Geosciences, Environmental and Marine Sciences, Life Health and Medical Sciences, Mathematics and Statistics, Nanotechnology, Chemistry, Physics, International Relations, Political Science, Government, Policy, Risk Analysis, Science Writing, Public Affairs, and Computer Sciences
- Stipend based on academic status – range from $529/week to $935/week for full-time; pro-rated for part-time
- Travel/Housing assistance (if eligible)
- Professional development activities
- Minimum GPA - 3.0/4.0
- Open to U.S. and Eligible International Citizenship

Visit http://www.orau.org/ornl or contact NESLS@orau.org for more information!

- 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.

Center for Undergraduate Research in Math

urmath.org/curm/resources-about-undergraduate-research/

curm

CURM
Center for Undergraduate Research in Mathematics

- HOME
- MINIGRANTS
- SUMMER FACULTY WORKSHOP
- APPLICATION FORM
- PROPOSAL TIPS
- RESOURCES FOR STUDENTS
- **RESOURCES ABOUT UNDERGRADUATE RESEARCH**
- PERSONNEL

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.

"Keys to Successful Mentoring of Undergraduate Research Teams with an Emphasis in Applied Mathematics Research" by Hannah L. Callender, *Proceedings of the Sixth Symposium on BEER*, 2013,http://cas.illinoisstate.edu/ojs/index.php/beer/article/view/796.

"Academic year undergraduate research: the CURM model" by Tor A. Kwembe, Kathryn Leonard and Angel R. Pineda, INVOLVE 7:3 (2014), pp. 383-394.

"Obtaining Funding and Support for Undergraduate Research" by Michael Dorff and Darren A. Narayan, Apr. 2012, pp. 1-7.

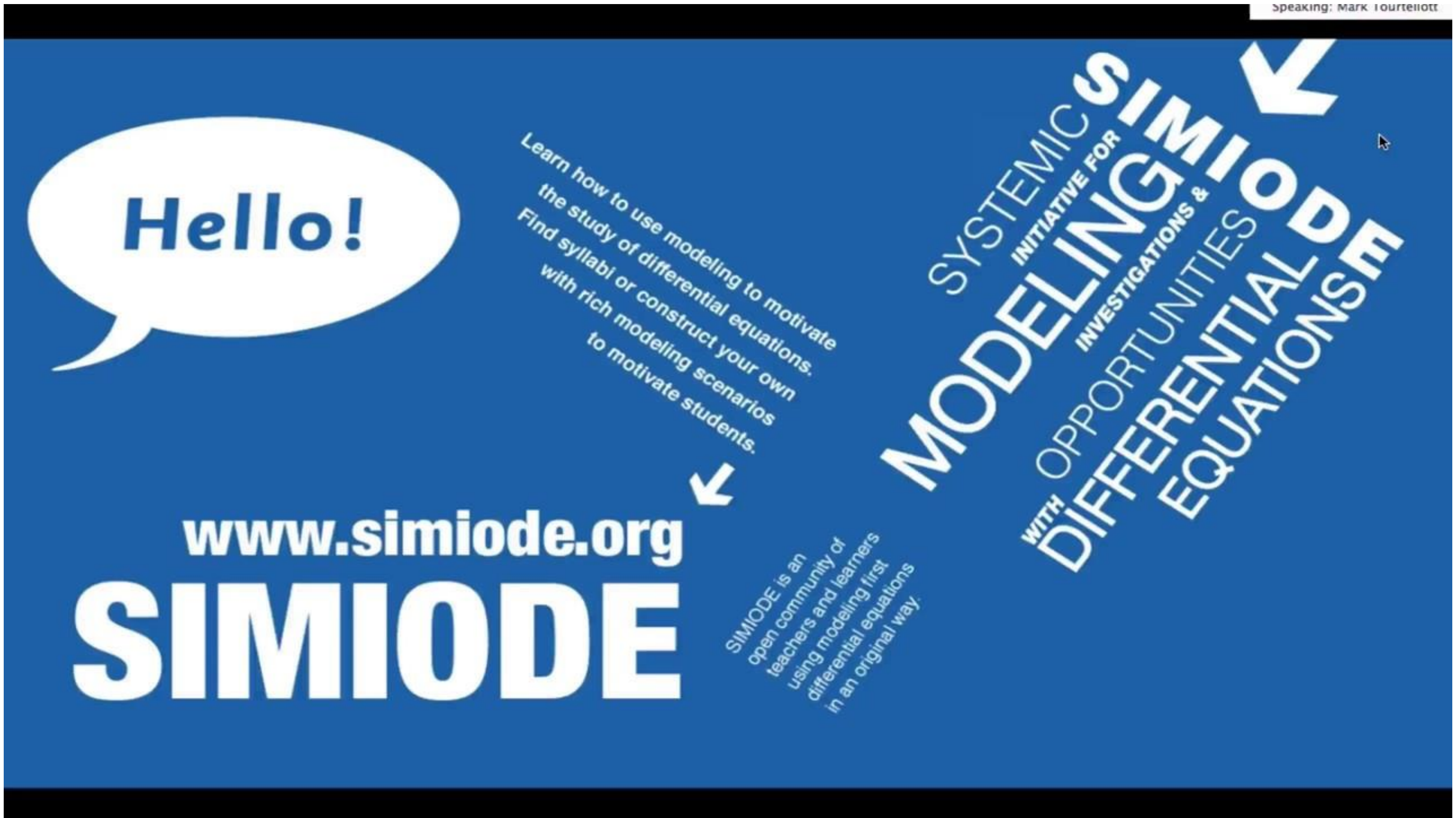
"Undergraduate Research: How Do We Begin?" by Brad Bailey, Mark Budden, Michael Dorff, and Urmi Ghosh-Dastidar, published in the MAA Focus, Jan. 2009, pp. 14-16.

"Adventures in Doing Academic Year Undergraduate Research" by Kathryn Leonard, published in the AMS Notices, Nov. 2008, pp. 1422-1426.

"Practical Tips for Managing Challenging Scenarios in Undergraduate Research" by Brad Bailey, Mark Budden, and Urmi Ghosh-Dastidar, published in the MAA Online Column Resources for Undergraduate Research, Dec. 2008.

"Assessing the impact of Undergraduate Research Experiences on Students" by Mary Crowe and David Brakke, published in the Council for Undergraduate Research Quarterly, Summer 2008, Vol. 28, Issue 4, pp. 43-50.

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
Mentor: Professor Chris McCarthy
Mathematics Department, CUNY Borough Of Manhattan Community College
Research Group Professors McCarthy, Navarro, Tesfagiorgis

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 bioremediation 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 issues. It is important for researchers to find economical and effective methods for heavy metal removal.

Figure 1

Figure 2

FILTERING MODEL (CONCEPTUAL)

Filter modeled as a one dimension strip with S_e binding sites (figure 3). Particles bind to a site with probability *p* and don't bind with probability *q =1-p*.

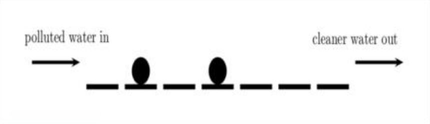


Figure 3

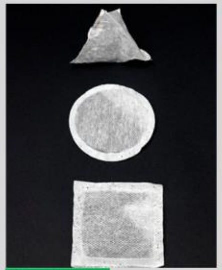


Figure 4

Figure 5

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_e binding sites. For each binding site there is probability *p* that the pollutant unit will stick to that binding site, and probability *q =1-p*, that it won't stick.

THE DIFFERENTIAL EQUATION (ODE)

Let ξ be the probability that a particle, entering the filter along with the m^{th} mL of waste water, will escape the filter. We want to know ξ as a function of m.

S = the number of particles stuck to the filter's binding sites, with S_T being the total number of binding sites in the filter.

$1 - \frac{S}{S_T}$ = the fraction of the filter's binding sites that are unoccupied. Hence, as a function of S, the escape probability ξ is:

$$\xi = q^{S_e\left(1-\frac{S}{S_T}\right)}$$

and so:

$$\frac{d\xi}{dS} = -\frac{S_e}{S_T}\,(\ln q)\, q^{S_e\left(1-\frac{S}{S_T}\right)} = -\frac{S_e}{S_T}\,(\ln q)\,\xi$$

Let C = the concentration of the particles entering the filter in units of $\frac{\text{particles}}{\text{mL}}$.

Recall S = the number of particles bound to the filter. So:

$$dS = (1-\xi)\, C\, dm \text{ particles,}$$

$$\frac{dS}{dm} = (1-\xi)\, C$$

Applying the chain rule we get the ODE:

$$\frac{d\xi}{dm} = \frac{d\xi}{dS}\frac{dS}{dm} = \left(-C\,\frac{S_e}{S_T}\ln q\right)\xi(1-\xi) \qquad (1)$$

Letting $\kappa = \left(-C\,\frac{S_e}{S_T}\ln q\right)$ and using the IC (initial condition) $\xi(0) = q^{S_e}$, Equation (1) becomes the IVP (initial value problem)

$$\frac{d\xi}{dm} = \kappa\xi\,(1-\xi),\ \xi(0) = q^{S_e}, \qquad (2)$$

The IVP (2) is easily solved by separation and then applying partial fraction expansion to the resulting integral. Using the IC and the definition of κ:

$$\xi(m) = \frac{q^{S_e}}{q^{S_e} + (1-q^{S_e})\,(q^{S_e})^{\frac{C}{S_T}m}} \qquad (3)$$

In Equation (3) $\xi(m)$ = faction of heavy metal particles remaining in the m^{th} mL of waste water output by the filter. Note. q^{S_e} is the probability that the first particle escapes the filter.

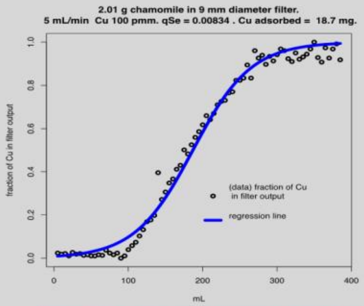


Figure 6. We apply nonlinear regression to find $q^{S_{e,1}}$

Let α = the filter's cross sectional area and its mass=$\mathbb{M}$.

The total adsorption capacity of the filter S_T is proportional to its mass.

So, for two filters: $S_{T,2} = S_{T,1}\frac{\mathbb{M}_2}{\mathbb{M}_1}$ and, by a non trivial argument:

$$q^{S_{e,2}} = \left(q^{S_{e,1}}\right)^{\frac{\mathbb{M}_2}{\mathbb{M}_1}\frac{\alpha_1}{\alpha_2}}$$

Using these substitution with equation (3) allows us to predict the escape probabilities $\xi_2(m)$ (for 2nd filter) if we know $q^{S_{e,1}}$ (from the 1st filter).

VERIFICATION OF MODEL'S PREDICTION USING LAB DATA

$$\xi_2(m) = \frac{\left(q^{S_{e_1}}\right)^{\frac{\mathbb{M}_2}{\mathbb{M}_1}\frac{\alpha_1}{\alpha_2}}}{\left(q^{S_{e_1}}\right)^{\frac{\mathbb{M}_2}{\mathbb{M}_1}\frac{\alpha_1}{\alpha_2}} + \left(1-\left(q^{S_{e_1}}\right)^{\frac{\mathbb{M}_2}{\mathbb{M}_1}\frac{\alpha_1}{\alpha_2}}\right)^{\frac{C_2}{S_{T,1}\frac{\mathbb{M}_2}{\mathbb{M}_1}}m}} \qquad (4)$$

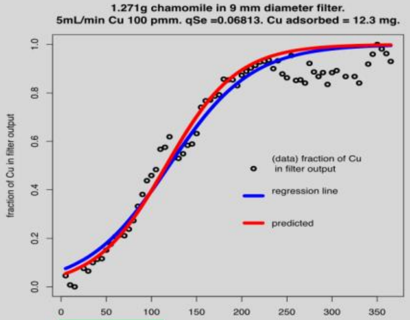


Figure 7

CONCLUSION

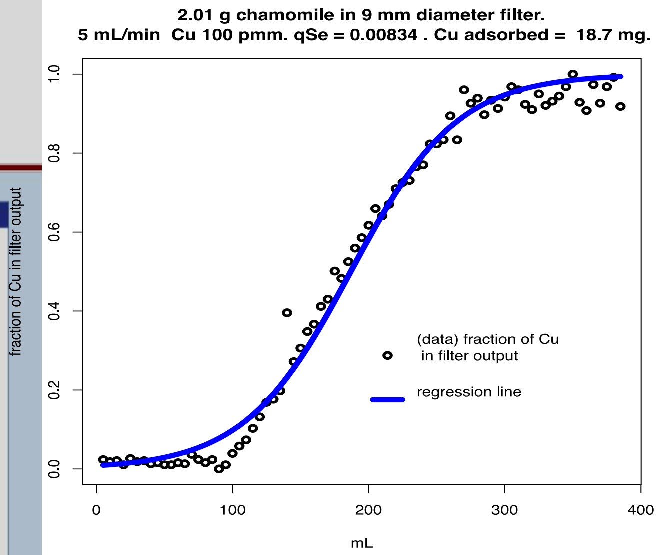
Our model fits the data. It also allows the responses of other filters (of the same material, but different masses, diameters, etc.) to be predicted, based upon data from lab experiments conducted on a single first (standard) filter.

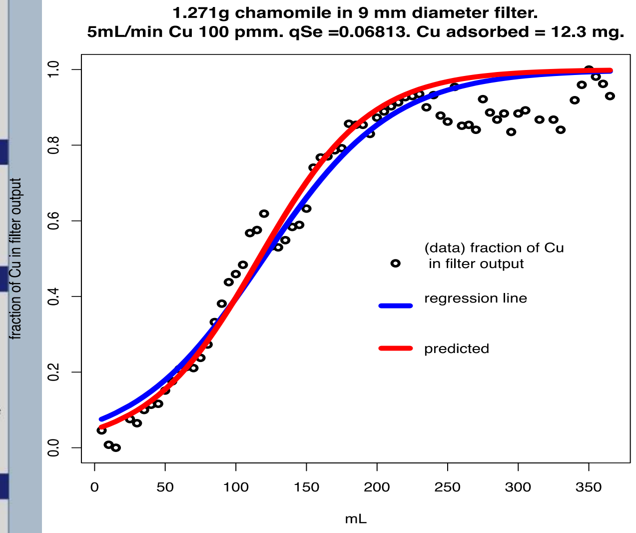
ACKNOWLEDGMENTS

Dr. Helene Bach, (BMCC Director of Research). Professors Abel Navarro and Kibrewossen Tesfagiorgis (BMCC Science). Research assistants: Ai Ngo, Xin Liu, and Mayumy Cordova Lozano; Jie Lan and Jieying Li; Seonin Cho, Min Yeong Hong, and Kwangmin Kim. Funding: Round 12, City University of New York, Community College Collaborative Incentive Research Grant (C3 IRG); Borough of Manhattan Community College Faculty Development Grant; the CUNY Research Scholars Program (CRSP).

REFERENCES

1. K. L. Wasewar, Adsorption of metals onto tea factory waste: a review, Int. J. Res. Rev. Appl. Sci 3 (3) (2010) 303. 6
2. T. Kim, D. Yang, J. Kim, H. Musaev, A. Navarro, et al., Comparative adsorption of highly porous and raw adsorbents for the elimination of copper (II) ions from wastewaters, Trends in Chromatography 8 (2013) 97–108.
3. J. T. Nwabanne, P. K. Igbokwe, Adsorption Performance of Packed Bed Column for the removal of Lead (ii) using oil Palm Fibre, International Journal of Applied Science and Technology 2 (5) (2012) 106 – 115.
4. Z. Xu, J.-G. Cai, B.-C. Pan, Mathematically modeling fixed-bed adsorption in aqueous systems, Journal of Zhejiang University-Science A (Applied Physics & Engineering) 14 (3) (2013) 155 – 176.





Senayit Menasche & Abdulai Jalloh (2017)

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

ONLINE https://mccarthymat501.commons.gc.cuny.edu/newtonian-cooling/


METRIC
HABOR
TIMER
0:00 00
H
M
S
ST/SP
MODE
CLEAR
HABOR
77.0
33

Students copy and modify the R script. They run it on online (RexTester.com) or on their computer.

R is open source!

https://rextester.com/l/r_online_compiler

Run Code | Code Wall | Users | Misc | Feedback | About

compile R online

Language: R Editor: CodeMirror Layout: Vertical

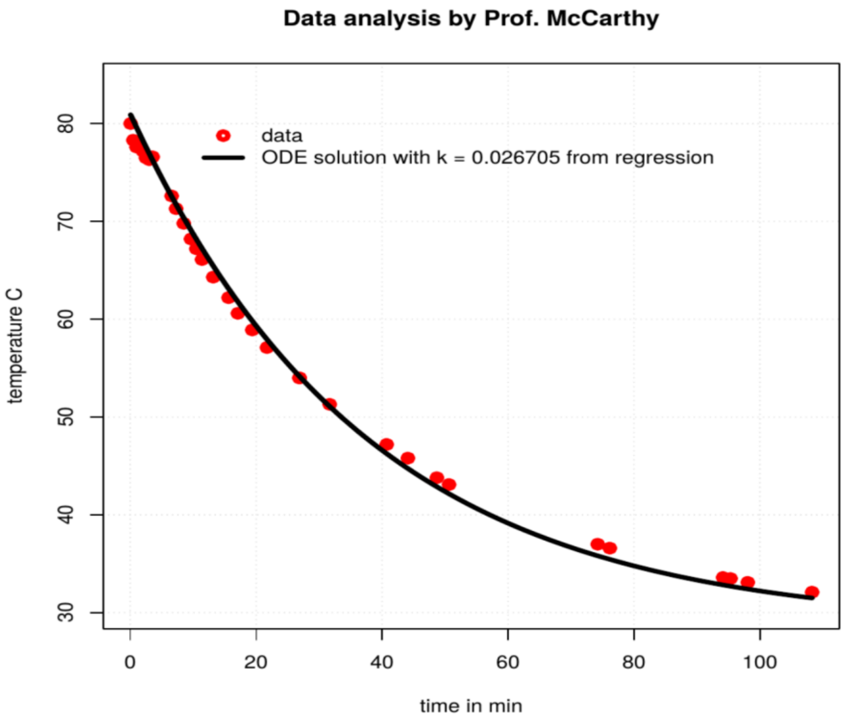
```
#  RcodeNewtonLawOfCoolingRegPlot.R
#  Does regression on data from Newton's Law of Cooling Experiment and
#  plots results. For Mat 501 notes.

# Experiment notes:    Date: Feb 9, 2018
# 500 mL of near to boiling water in a pyrex measuring cup.
# about 50 mL evaporated eventually.

# air and initial water temp (C)
T = 28.6  # air temp (78 deg F)
y0 = 81.0 # initial water temperature (177.8 deg F)

# data from experiment taken manually
# (t_i, y_i) = (time in min, water temp in C);

# t_i time data (min) Note a measurement like 60+34+9/60 means 1 hr 34 min 9 sec
t_i = c(4/60, 30/60,  1,   1+31/60, 2  ,2+30/60, 3+2/60, 3+34/60,
 6+35/60, 7+17/60,8+30/60, 9+39/60,10+30/60, 11+24/60, 13+11/60, 15 + 37/60, 17+6/60,
 19+23/60,21+43/60, 26+51/60, 26+55/60,31+41/60, 40+44/60,44+6/60, 48+42/60, 50+38/60,
 60+14+12/50, 60+16+10/60, 60+34+9/60, 60+35+20/60, 60+38+4/60, 60+48+17/60 );

# y_i water temperature data (C)
y_i =  c(80.0, 78.3,  77.6, 77.6,  77.2 , 76.5 ,  76.3,  76.6,
 72.6, 71.3, 69.8, 68.2, 67.2, 66.1, 64.3, 62.2, 60.6,
 58.9, 57.1, 54.0, 54.0, 51.3, 47.2, 45.8,43.8, 43.1,
 37.0, 36.6, 33.6, 33.5, 33.1, 32.1);

# display data in console.
t_i;
y_i;

# approximate k
```

Run it (F8) Save it [+] Show input Live cooperation Put on a wall F ?

Absolute running time: 2.89 sec, cpu time: 0.88 sec, memory peak: 34 Mb, absolute service time: 2,96 sec

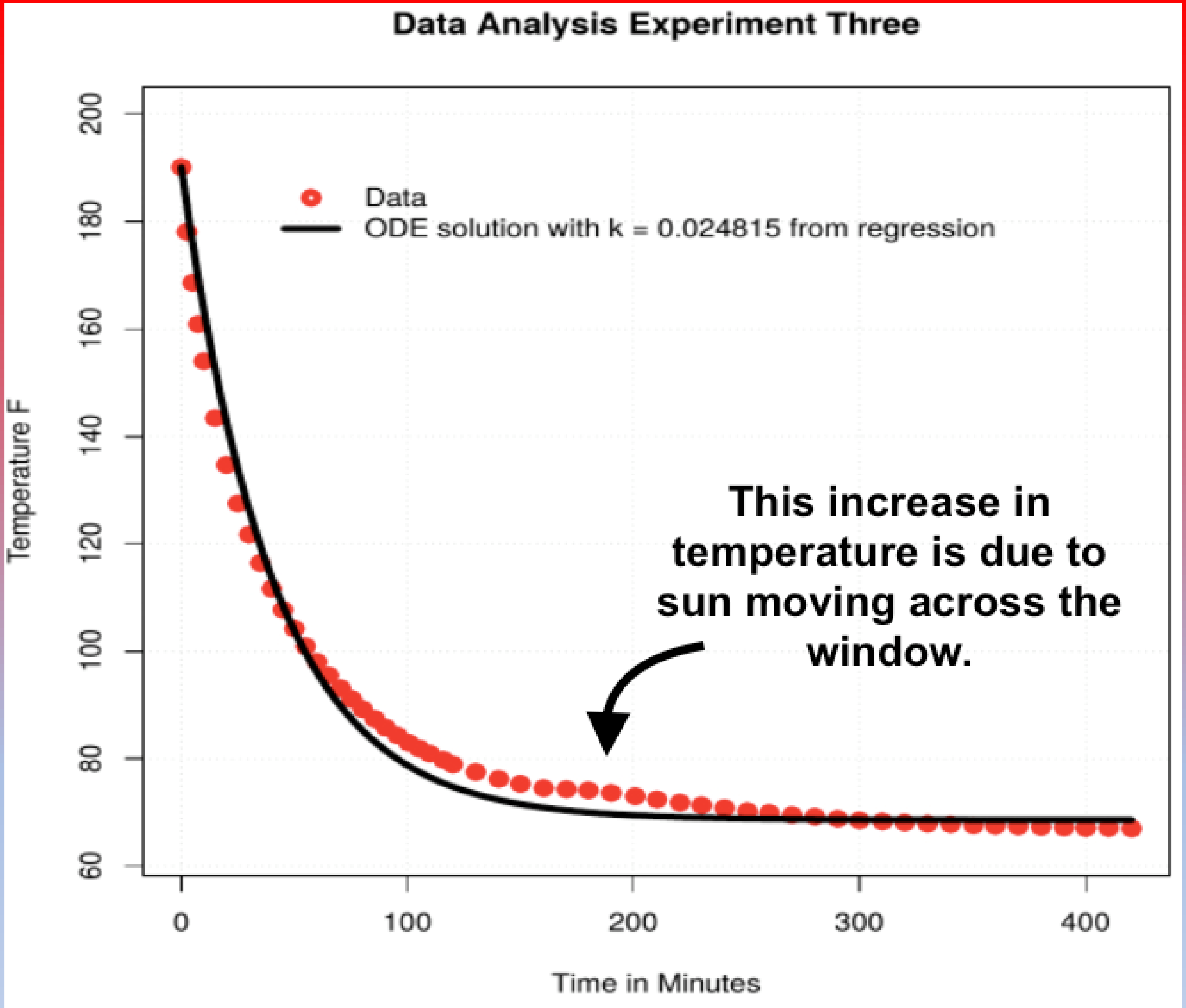
```
 [1]  0.06666667  0.50000000  1.00000000  1.51666667  2.00000000
 [6]  2.50000000  3.03333333  3.56666667  6.58333333  7.28333333
[11]  8.50000000  9.65000000 10.50000000 11.40000000 13.18333333
```

```
Residual standard error: 0.9992 on 31 degrees of freedom

Number of iterations to convergence: 3
Achieved convergence tolerance: 9.936e-06

         k
0.02670549
```



Open ended modeling question for students

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



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

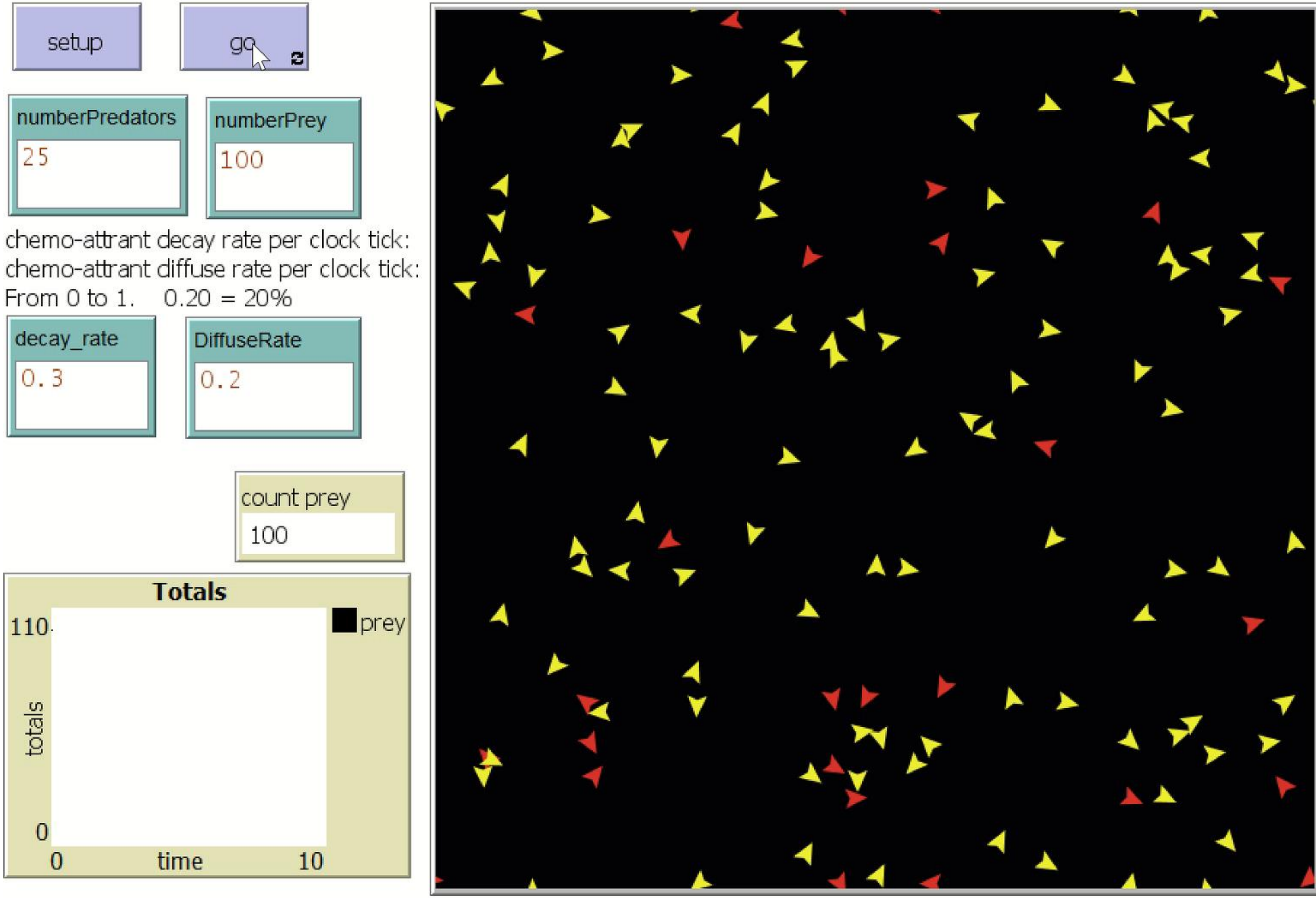
OER Warrior Extraordinaire

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**).



CRSP

2018 – 2019

Students Presented at various conferences including the 2019 Joint Mathematics Meetings in Baltimore

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

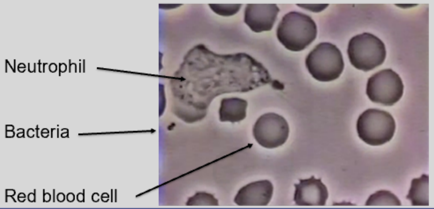
ACTIVE MATTER

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].

A flock of starlings. John Holmes CC

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).

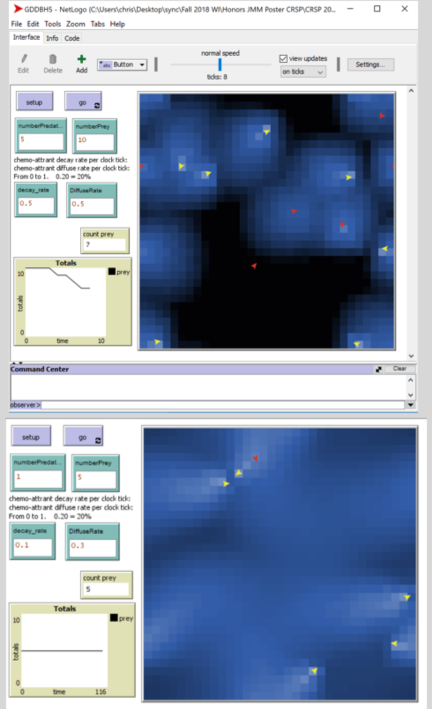


AGENT BASED MODELING OF CHEMOTAXIS

Our research involved developing agent based simulations of chemotaxis. These simulations were developed in NetLogo. We modeled a predator (red triangles) hunting down prey (yellow triangles). In each time step the prey excretes a chemoattractant (blue color) which diffuses and decays. The amount of chemoattractant present at a location is indicated by the shade of blue. Black = no chemoattractant. As the chemoattractant level increases the blue becomes lighter. The predator senses the chemoattractant and follows its gradient (hoping) to find its prey. When the prey is found it is killed by the predator. We varied the diffusion and decay rates, and the number of prey and predators, and recorded the number of time steps till extinction of the prey.

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.



Images of the NetLogo Interface running simulations.

FUTHER RESEARCH

In the future, we hope to accomplish

1. Understanding the uptick 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 lifelike simulations. For example, where both species reproduce and die; where species are more biologically accurate.

RESULTS (for Diffusion Rates)

Figure 1

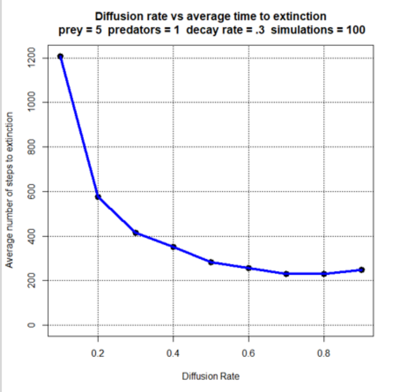
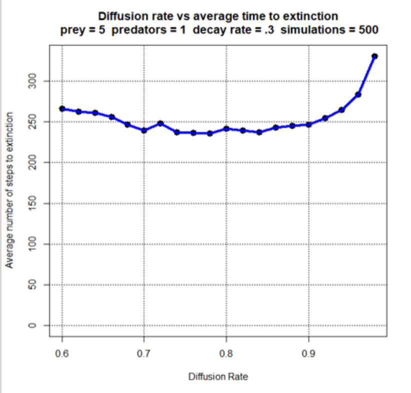


Figure 2



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 0.9 results in it taking longer for the predators to capture the prey. Figure 2 required 10,000 simulations (500 simulations x 20 different diffusion rates).

In Figures 3 we see that the shape of the Decay Rate vs. Average Extinction Time graph seems to make a sort of "U" 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 chemoattractant isn't decaying, and the predator is misled by chemoattractant remnants. If the decay rate is close to 1, the chemoattractant decays too quickly to be of use to the predator.

RESULTS (for Decay Rates and Prey Numbers)

Figure 3

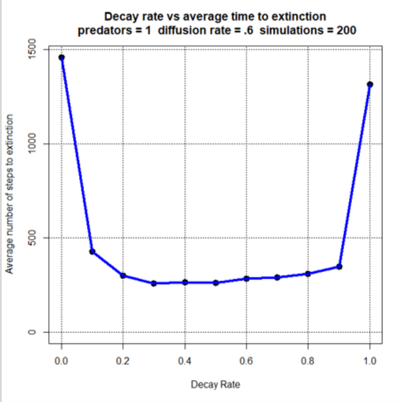
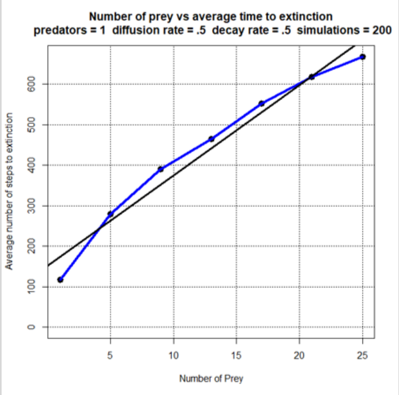


Figure 4



Black linear regression line has equation y = 22x + 152

ACKNOWLEGMENTS

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

REFERENCES

1. Skylar Tibbits (Editor), Active Matter. The MIT Press (2017)
2. T. Vicsek, A. Czirók, E. Ben-Jacob, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.
3. Michael Eisenbach. Chemotaxis. World Scientific Publishing (2004)

CRSP

2018 – 2019

Students Presented at various conferences including the 2019 Joint Mathematics Meetings in Baltimore

CUNY RESEARCH SCHOLARS PROGRAM

Active Matter: Predator Prey Interactions

Jorwyn Medina, Muhammad Hannan, Adama Sene

Mentor: Professor Chris McCarthy
BMCC Mathematics

BMCC

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 pretty.

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].

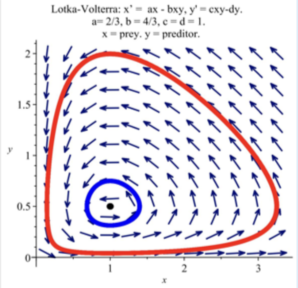
Lotka Volterra

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):

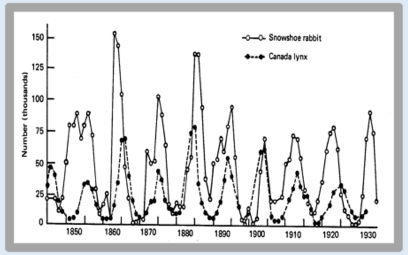
$$dg/dt = a(K - g) - bsg$$

$$ds/dt = cgs - ds - ews$$

$$dw/dt = fsw - hw$$ .where *a, b, c, d,e,f,h* >0



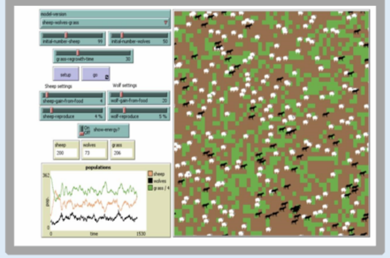
One of the classic predator prey relations modeled by the Lotka Volterra equations is the relationship of the arctic lynx and snowshoe hare populations.



The population size of lynx and hare can be estimated from the commercial records of the Hudson Bay Company of how many lynx and hare pelts they purchased [3].

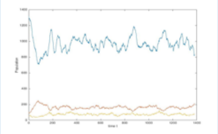
Netlogo and Matlab

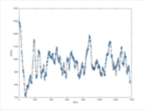
1. Using NetLogo we simulated Lotka Volterra predator prey type system. The Wolf-Sheep simulation we used was created by U. Wilensky [4].

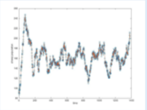


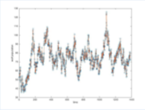
2. We export the wolf, grass and sheep population data to a spreadsheet. We then import the data into MATLAB. Below, we plotted the regular data taken from Netlogo simulation. In order to have better approximation for the differentiation, we use the csaps function to smoothen the data. However, plot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

time	sheep	wolves	grass
0	1000	100	13578
1	948	102	14117
2	948	102	14117
3	913	106	14654
4	913	106	14654
5	912	103	15236
6	912	103	15236
7	914	110	15836
8	914	110	15836
9	910	114	16401

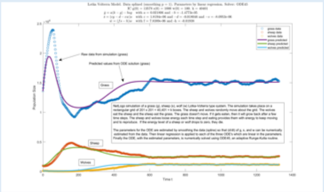








3. We numerically differentiate the splined data in matlab to estimate dg/dt, ds/dt and dw/dt where s = sheep and w=wolves and g = grass. The derivative is obtained by using the fnder function.
4. The parameters a, b, c, d, e, f and h are linear in the Lotka Volterra differential equations. We apply matlab's linear regression routine to the splined data and the numerical estimates of dg/dt, ds/dt and dw/dt to get estimates for a, b, c, d, e, f and h.
5. Using Runge Kutta and the estimates from a,b,c,d, e, f and h we numerically solve the Lotka Volterra system and plot the results.



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

6. Stability

Our Lotka Voltera system of ODE's has an equilibrium point(where gdot=sdot=wdot=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-yI)=0.

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

```
# import data from excel to matlab using upload from the home tab and then import it into the command window. To access it, do the following:
 A = importdata("TimeSheepWolvesGrass1.csv")
#------------------------------------------------------------------------
# Smoothing: create spline object and turn it into a function.
P=1
Sp_g = csaps(t, g, p) ; sp_s = csaps(t, s, p); sp_w = csaps(t, w, p);
% In order to plot it, we have to turn the spline object into a function using the fnval function.
Plot(t, fnval(sp_g, t)); plot(t, fnval(sp_s, t)); plot(t, fnval(sp_w, t))
#------------------------------------------------------------------------
# numerical differentiation applied to the splined data
Dg/dt= fnval(fnder(sp_g,1), t); ds/dt = fnval(fnder(sp_s,1), t);
Dw/dt = fnval(fnder(Sp_w,1), t);
#------------------------------------------------------------------------
# we estimate parameters using linear regression#. we then use Runge 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: [t,gsw] = ode45(@(t,gsw) [a*(K-gsw(1))+b*gsw(2)*gsw(1);c*gsw(1)*gsw(2)+d*gsw(2)+e*gsw(3)*gsw(2); f*gsw(2)*gsw(3)+h*gsw(3)], [to tf], [g0; s0; w0].
```

Future Research

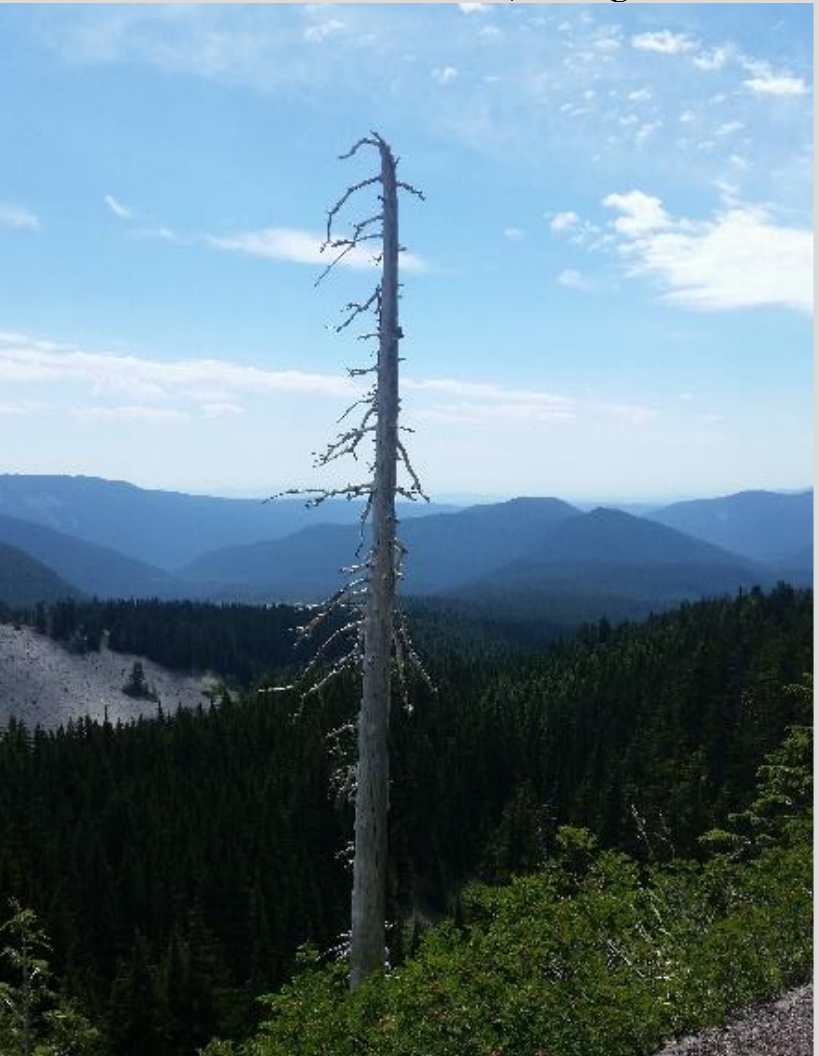
1. Further improve the algorithm to estimate the parameters.
2. Understand the changes in our parameters.

References

1. Lotka, A. J. (1925). Elements of physical biology. New York: Dover.
2. Volterra, V. (1926, October 16). Fluctuations in the abundance of a species considered mathematically. Nature, 118, 558–560.
3. Trophic Links: Predation and Parasitism. (n.d.). Retrieved from https://globalchange.umich.edu/globalchange1/current/lectures/predation/predation.htm
4. Wilensky, U. (1997). NetLogo Wolf Sheep Predation model. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Differential Equations
Model And Resource
Creators Workshop

View from Mt. Hood, Oregon



SIMIODE DEMARC Workshop

George Fox University
Oregon, July 2019

SYSTEMIC INITIATIVE FOR MODELING INVESTIGATIONS & OPPORTUNITIES WITH DIFFERENTIAL EQUATIONS
SIMIODE

NSF

GEORGE FOX UNIVERSITY
1891
CHRISTIANITY AND CULTURE

DEMARC Goal

To Develop Diff Eq Modeling Projects

(That are good for students)

Leonhard Euler 1707 - 1783

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

The drag on a ball

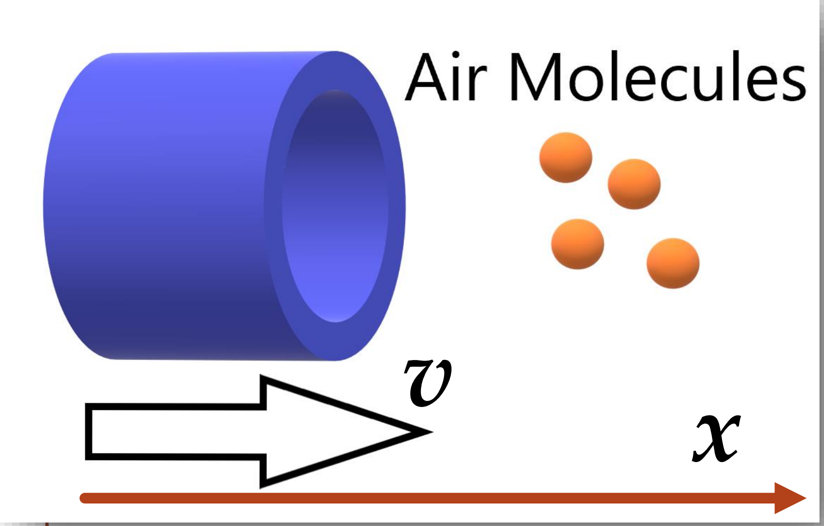
Heuristic argument: drag force proportional to v^2

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

$\rho =$ density of air. $\quad v =$ velocity of projectile

$A =$ cross sectional area. $\quad \Delta x = v\Delta t$

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



$$\Delta v_{air} \propto v_{projectile} = v$$

$$F_{drag} \propto \frac{\overbrace{\rho A\Delta x}^{mass\ air} \quad \Delta v_{air}}{\Delta t} \approx \rho A v\, v = \rho A v^2$$

F_{drag} is in opposite direction of v.

Drag Equation

$$F_{drag} = \tfrac{1}{2} C_D\, \rho A v^2$$

$C_D =$ drag coefficient

Euler recursive relation including drag

$$\begin{pmatrix} x \\ y \\ v_x \\ v_y \\ t \end{pmatrix}_0 = \begin{pmatrix} 0 \\ 2 \\ 12\cos\theta \\ 12\sin\theta \\ 0 \end{pmatrix}$$

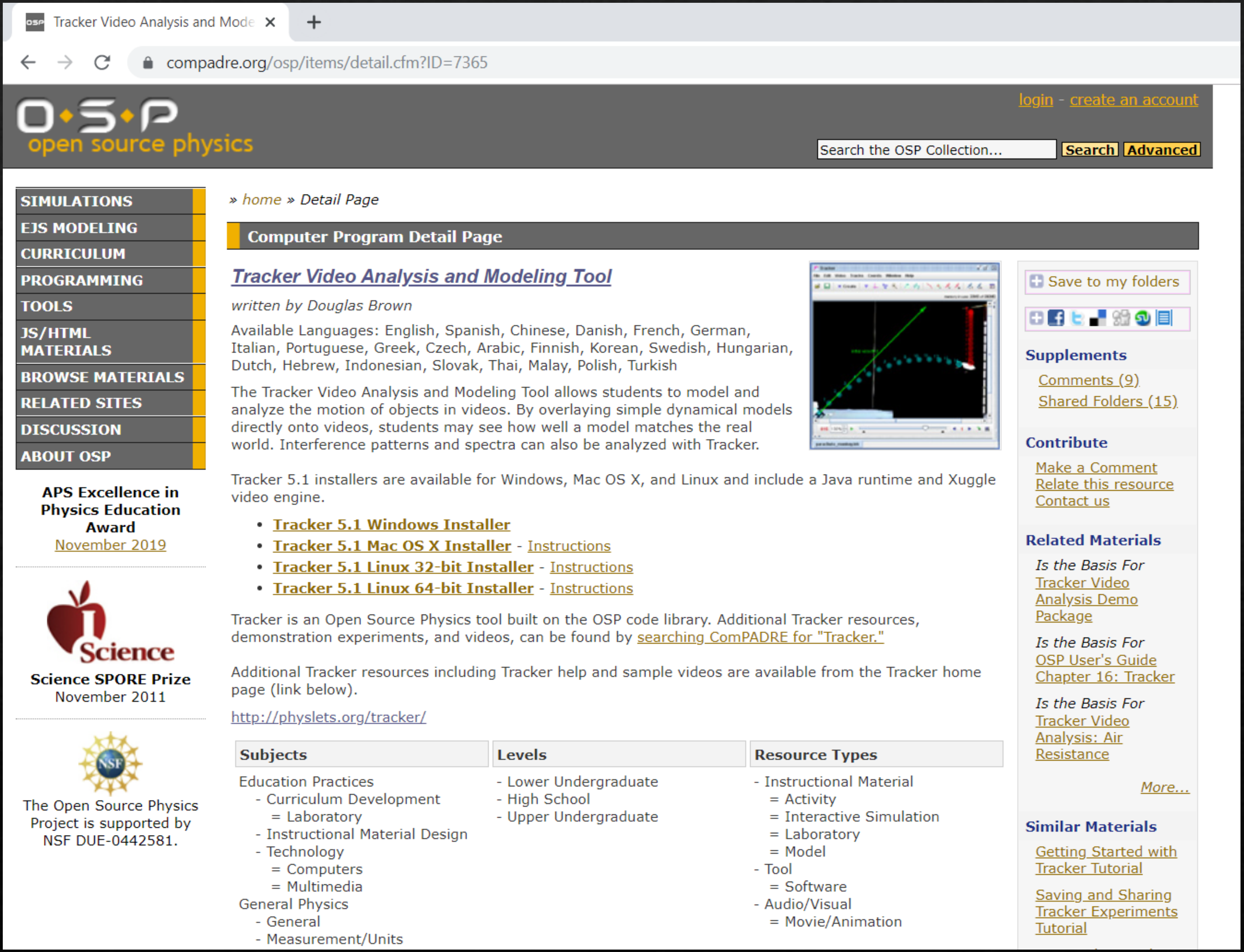
Initial conditions
Position 2 meters up
Speed 12 m/s
Launch angle θ varies

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

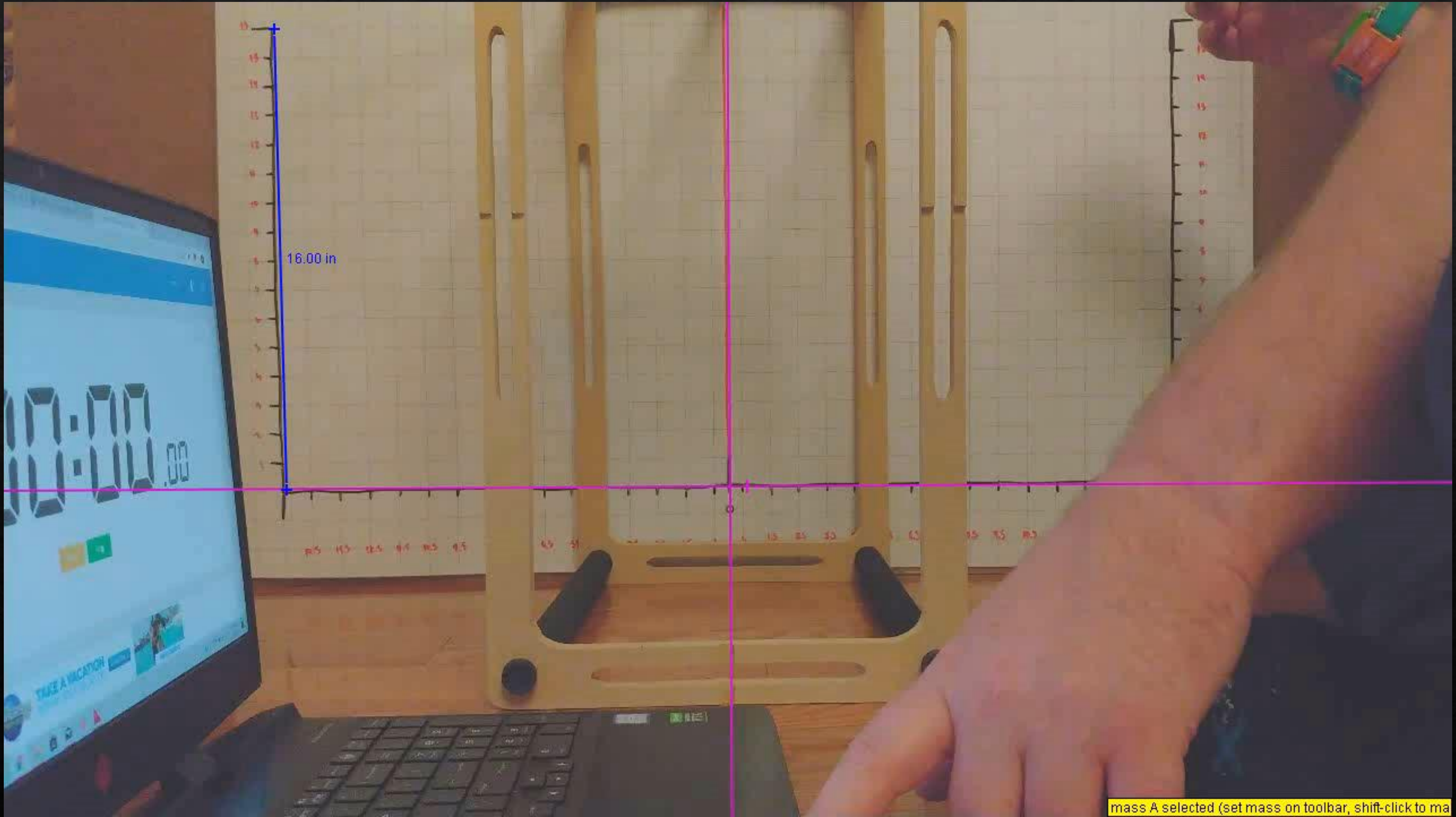
My honors student Kujtim Bardhyll worked with me to test the drag model on a real pendulum.

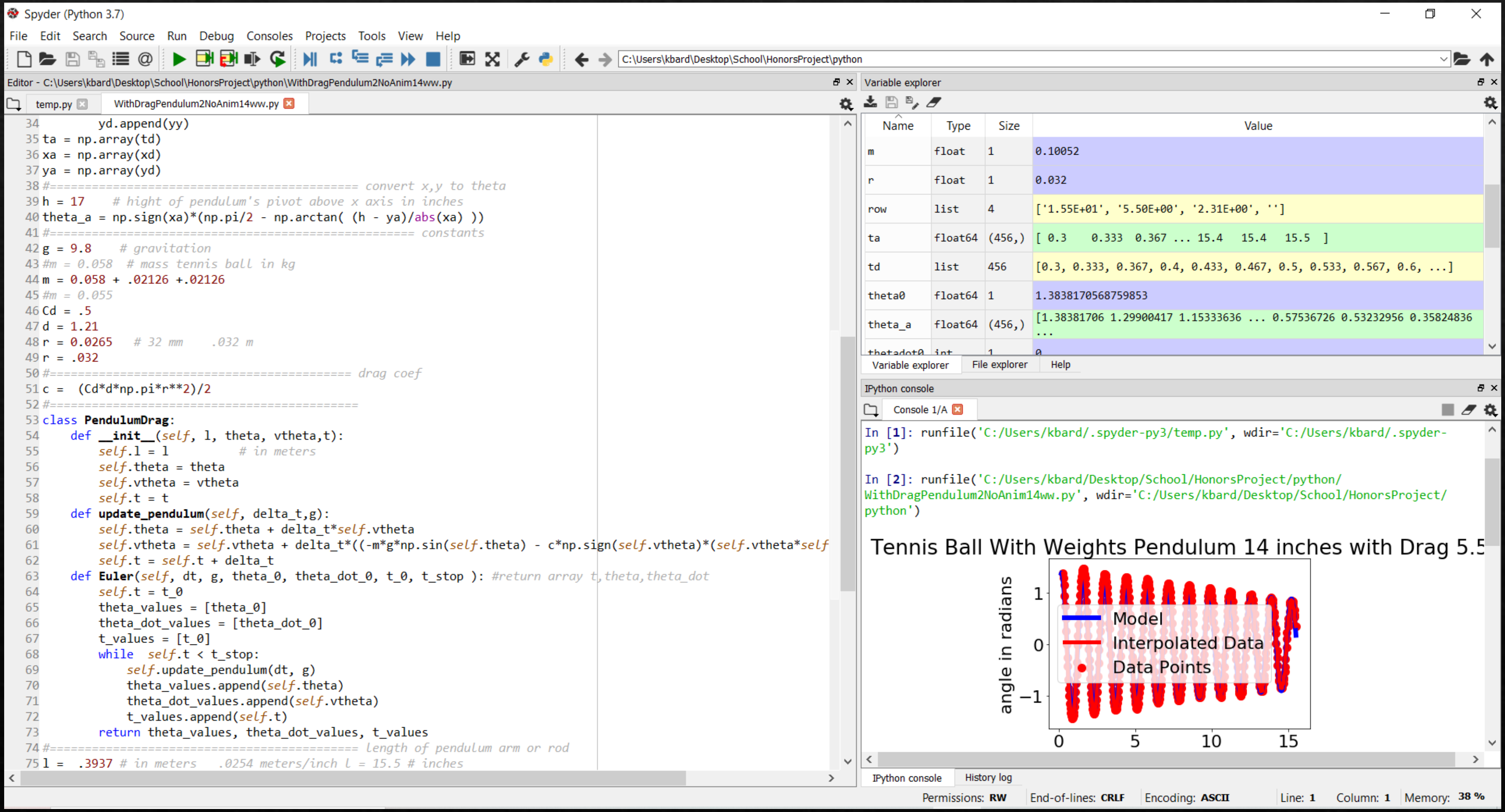
From Kutjim's presentation

- 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





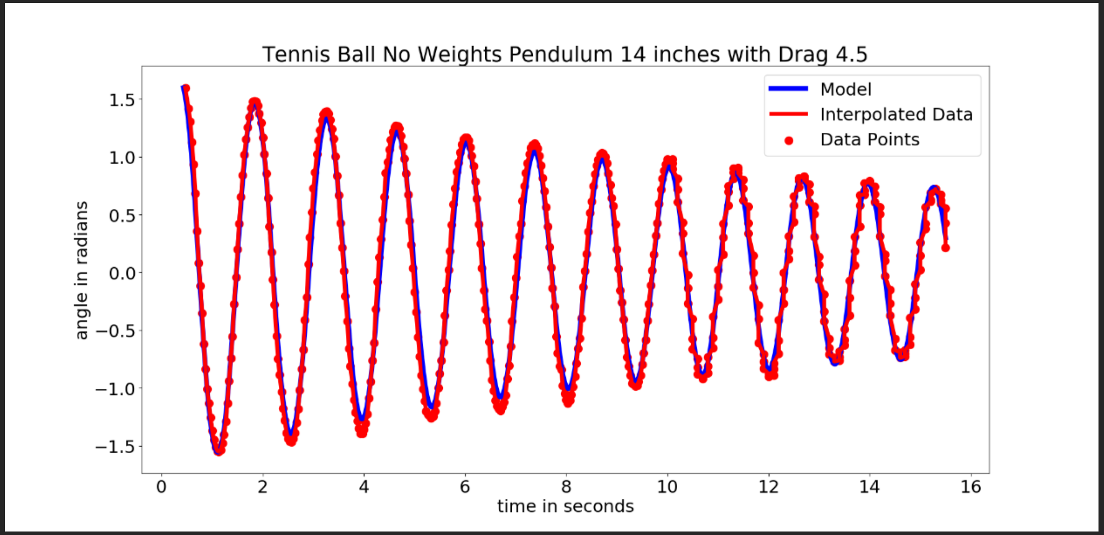
The pendulum data from the Tracker software was imported into Python where it was combined with our ODE model, which was solved using Euler's method.

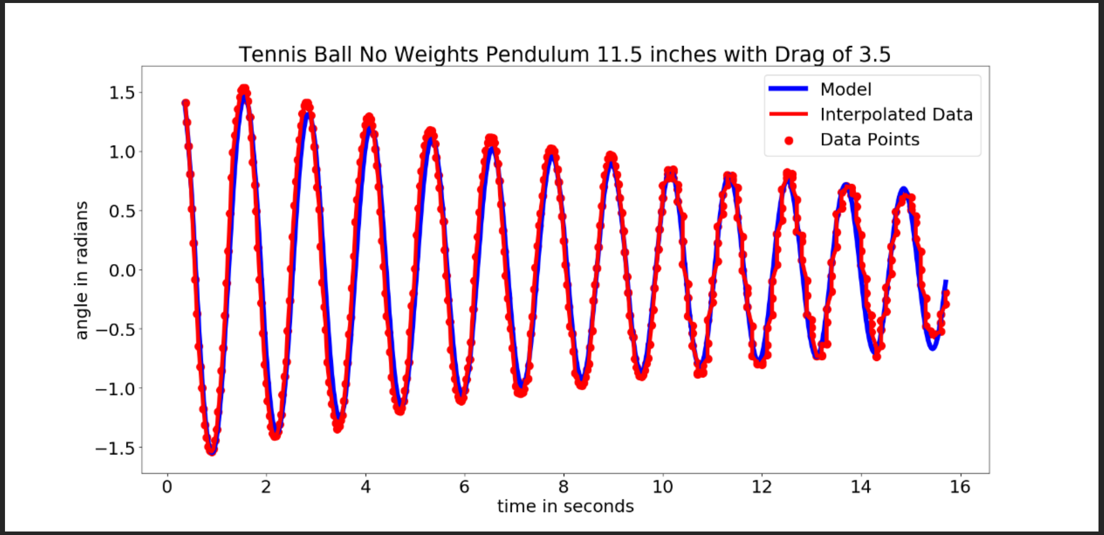
From Kutjim's presentation

L = 14 inches

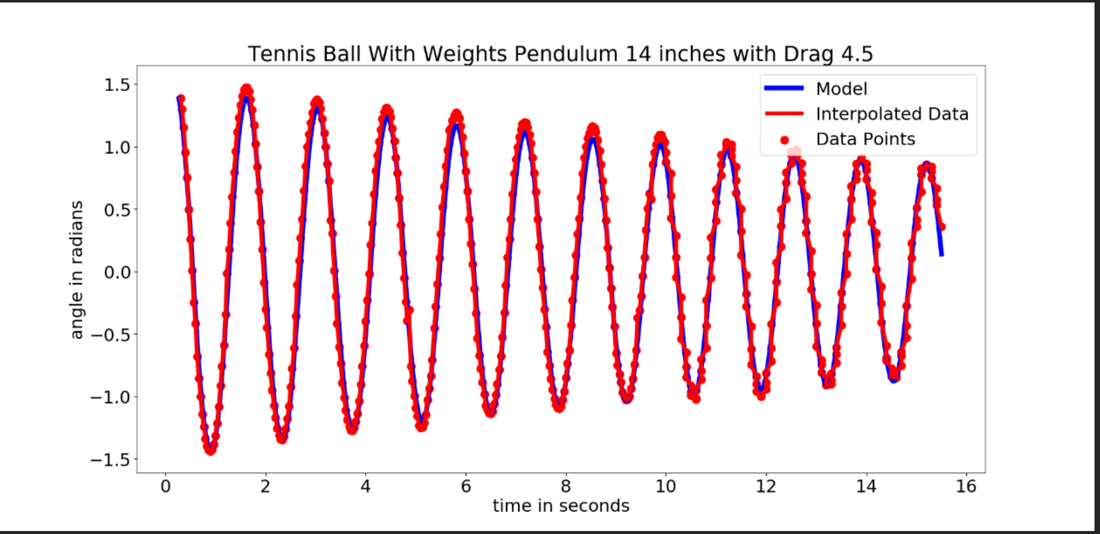
L = 11 (top)/11.5 (bottom) inches

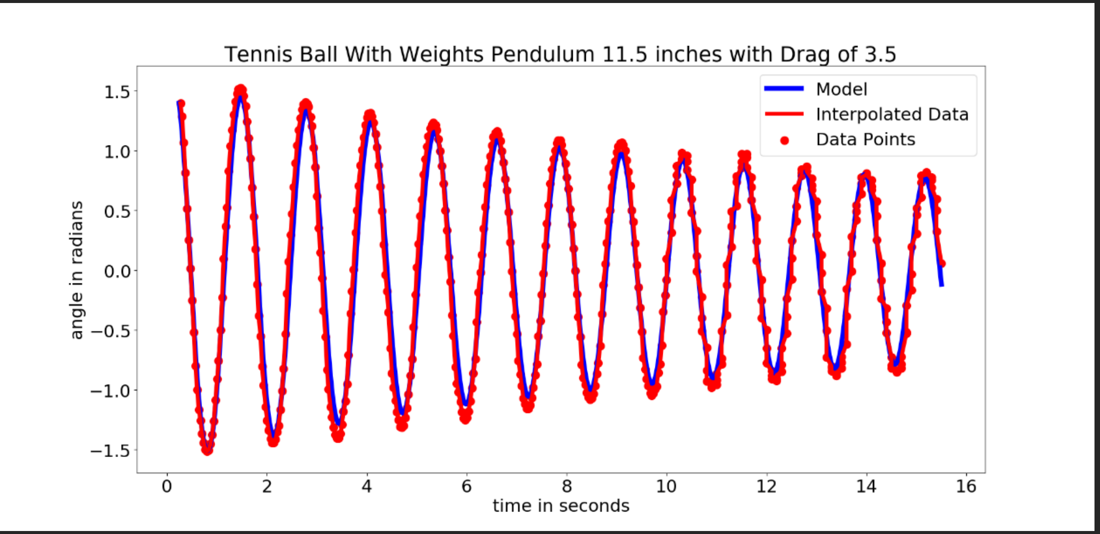
No added weight





With added weight





$$T \approx 2\pi\sqrt{L/g}$$

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

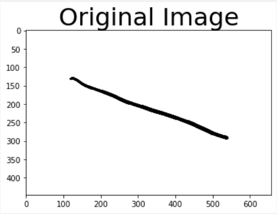


Figure 12 : Hand drawn image of a line with negative slope

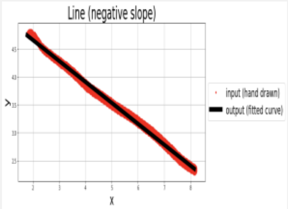


Figure 13 : Best fitting (regressed) line (black) is superimposed on the hand drawn line from Figure 12 (now colored red).

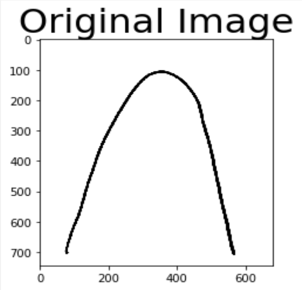


Figure 14 : Hand drawn image of a parabola concave down

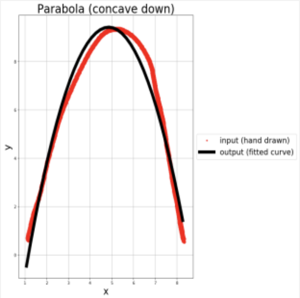


Figure 15 : Best fitting (regressed) parabola (black) is superimposed on the hand drawn parabola from Figure 14 (now colored red).

Training process of the neural network

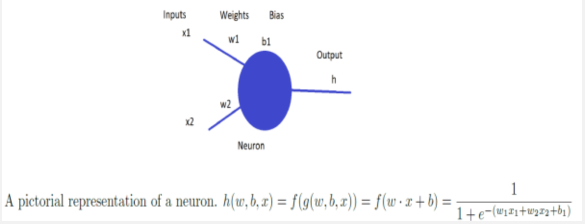
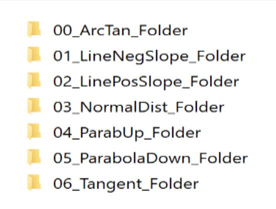


Figure 16 : Process of training our neural net

(a)



(b)

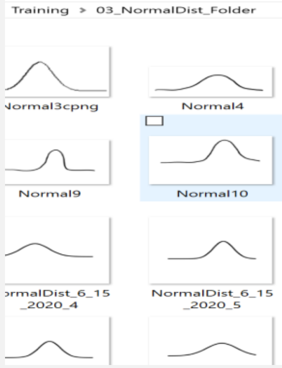


Figure 17 a: Folder structure. Each folder holds training images of a single category of function.
Figure 17b: Contents of the 03_NormalDist_Folder is a mixture of computer and hand drawn images of the normal distribution.

Model Summary

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 126, 126, 64)      1792
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 124, 124, 64)      36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 62, 62, 64)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 62, 62, 64)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 246016)            0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                15745088
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 455
=================================================================
Total params: 15,784,263
Trainable params: 15,784,263
Non-trainable params: 0
```

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 x 62 x 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.

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

Python function that produced the superposed images

```python
def fitFile(imgDir, fileName, modelName, classNamesOfFunctions): #fileName, imageDir, sx,sy, model_name, (
    pred = predictionForFile(imgDir, fileName, modelName, classNamesOfFunctions)
    LoadedImage = load_img(imgDir + fileName) # color_mode="grayscale",
    plt.imshow(LoadedImage)
    plt.title('Original Image', fontsize=32)
    LoadedImage = load_img(imgDir + fileName, color_mode="grayscale") # color_mode="grayscale",
    ImArray = img_to_array(LoadedImage)
    numRows = ImArray.shape[0] # rows
    numCols = ImArray.shape[1] # cols
    ImArrayNorm = 1 - (ImArray/255)
    ImArrayNorm2d = np.reshape(ImArrayNorm, (numRows, numCols), order = 'C')
    IndicesWhereCurveIs = np.where(ImArrayNorm2d > .1) #.5 # row_array, col_array
    xCoords = IndicesWhereCurveIs[1]*(10/numCols)
    yCoords = ((numRows - 1) - IndicesWhereCurveIs[0])*(10/numCols)
    p_guess = (pred[2].pmin + pred[2].pmax)/2
    popt, pcov = curve_fit(pred[2].ff, xCoords, yCoords, p0=p_guess)
    fig =plt.figure(figsize=(12,12))
    ax = fig.add_subplot(111)
    ax.set_aspect(1.0/ax.get_data_ratio(), adjustable='box')
    fontsizeLegend = 20
    fontsizeAxis = 25
    fontsizeTitle = 30
    plt.plot(xCoords,yCoords, 'r.', label='input (hand drawn)')
    t = np.arange(min(xCoords), max(xCoords), 0.2)
    plt.plot(t, pred[2].f(popt, t) , 'k-',label= 'output (fitted curve)', linewidth=8)
    plt.xlabel('x', fontsize = fontsizeAxis)
    plt.ylabel('y', fontsize = fontsizeAxis)
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5), fontsize = fontsizeLegend)
    ax.set_title(pred[2].functionName[0], fontsize = fontsizeTitle)
    ax.grid()
    plt.show()
    return popt
```

Python function that 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.

References

[1]. Tutorials Point. Artificial Intelligence. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm
[2]. Expert System. What's Machine Learning? https://expertsystem.com/machine-learning-definition/
[3. Wikipedia. Machine Learning. https://en.wikipedia.org/wiki/Machine_learning
[4]. http://neuralnetworksanddeeplearning.com/index.html
[5]. Francois Chollet. 2018). Deep Learning with Python. Mannig Publications CO.
[6]. http://neuralnetworksanddeeplearning.com/chap1.html
[7]. https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9
[8]. https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/
[9]. https://aishack.in/tutorials/image-convolution-examples/

July 2020

CUNYwide

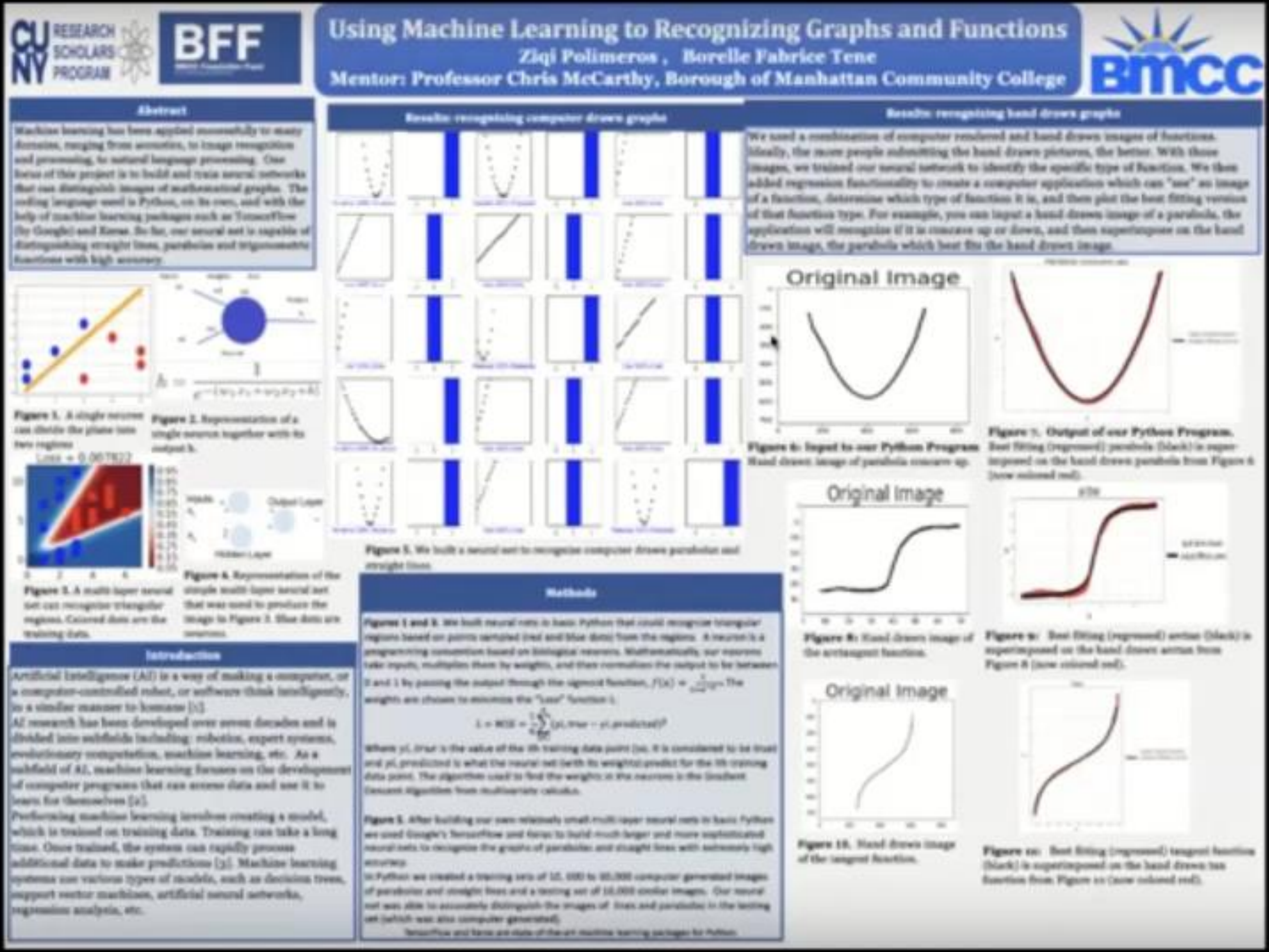
CRSP Symposium

July 2020

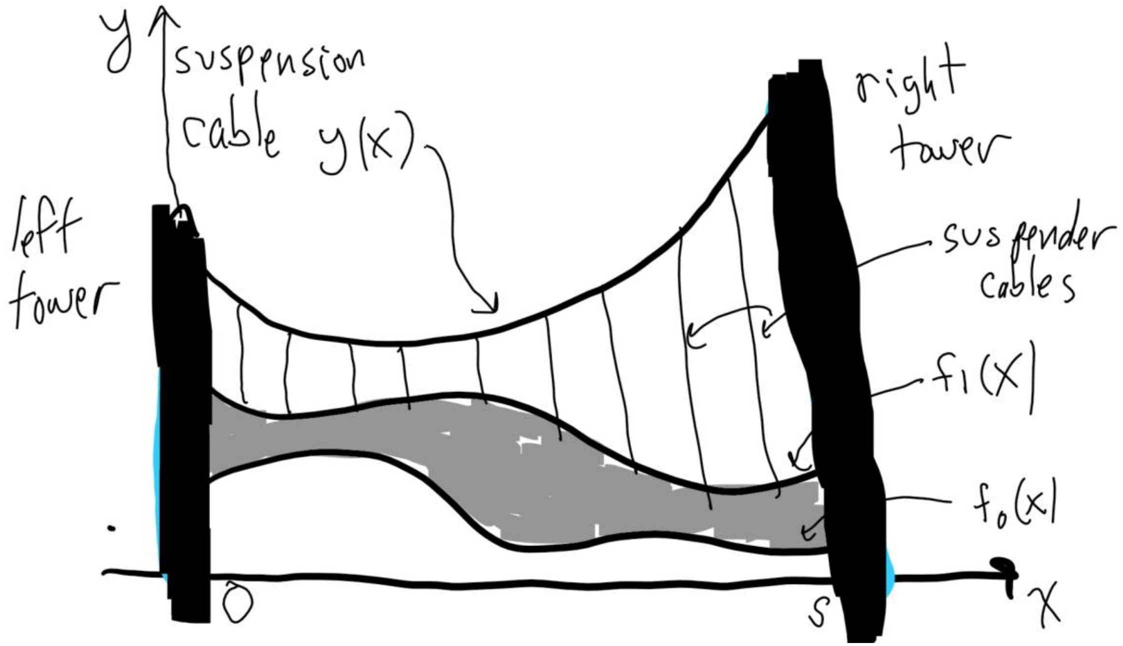
Virtual Due to COVID19

From Borrelle Fabrice Tene's

Presentation



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 *To*.



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) $y(x) = \int\int \frac{w(x)}{T_0}\, dx\, dx + c_1 x + c_0$

(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.

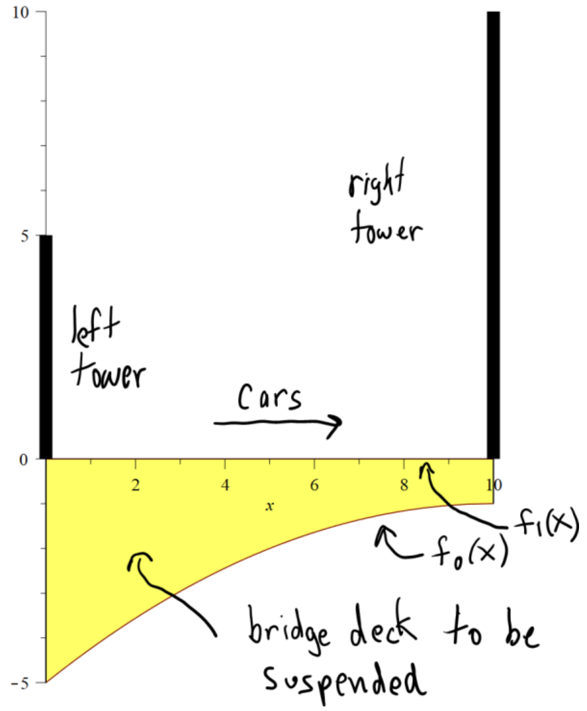
(5) Find T_0 so cable low point $myT(T_0)$ is at desired height. (Newton's Method)

(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 *To*.



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 *To*.

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) $y(x) = \int\int \frac{w(x)}{T_0}\, dx\, dx + c_1 x + c_0$

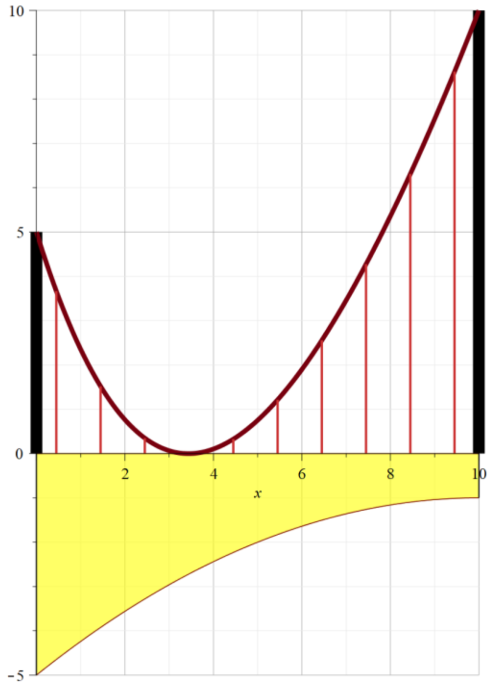
(3) Using BC (tower attachment heights) solve for c_1 and c_0 in terms of T_0

(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.

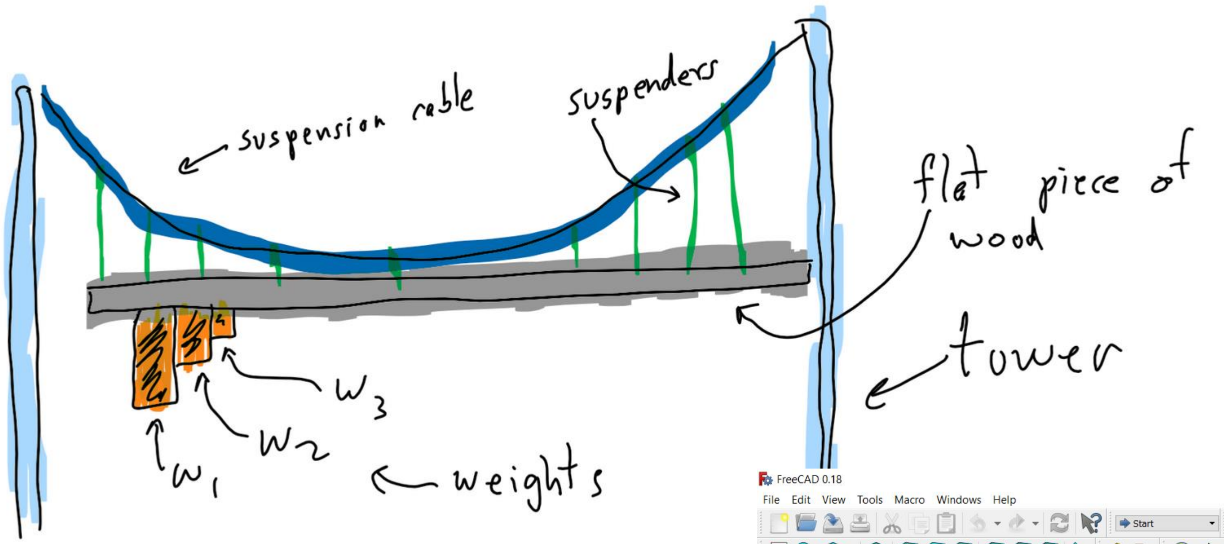
(7) Calculate where suspender cables are attached.

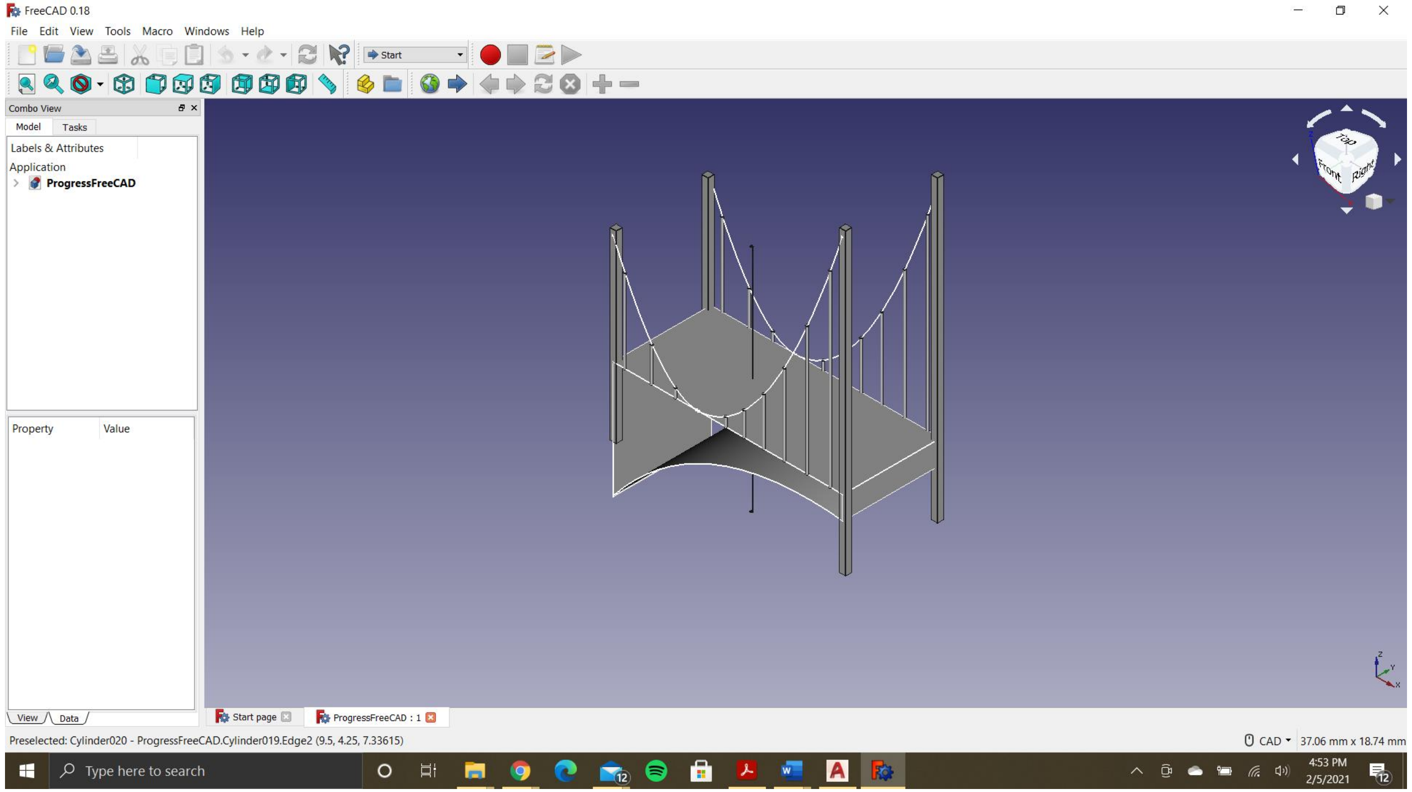


Solution

Bridges with unusual geometries
CAD and Building It

Spring 2021

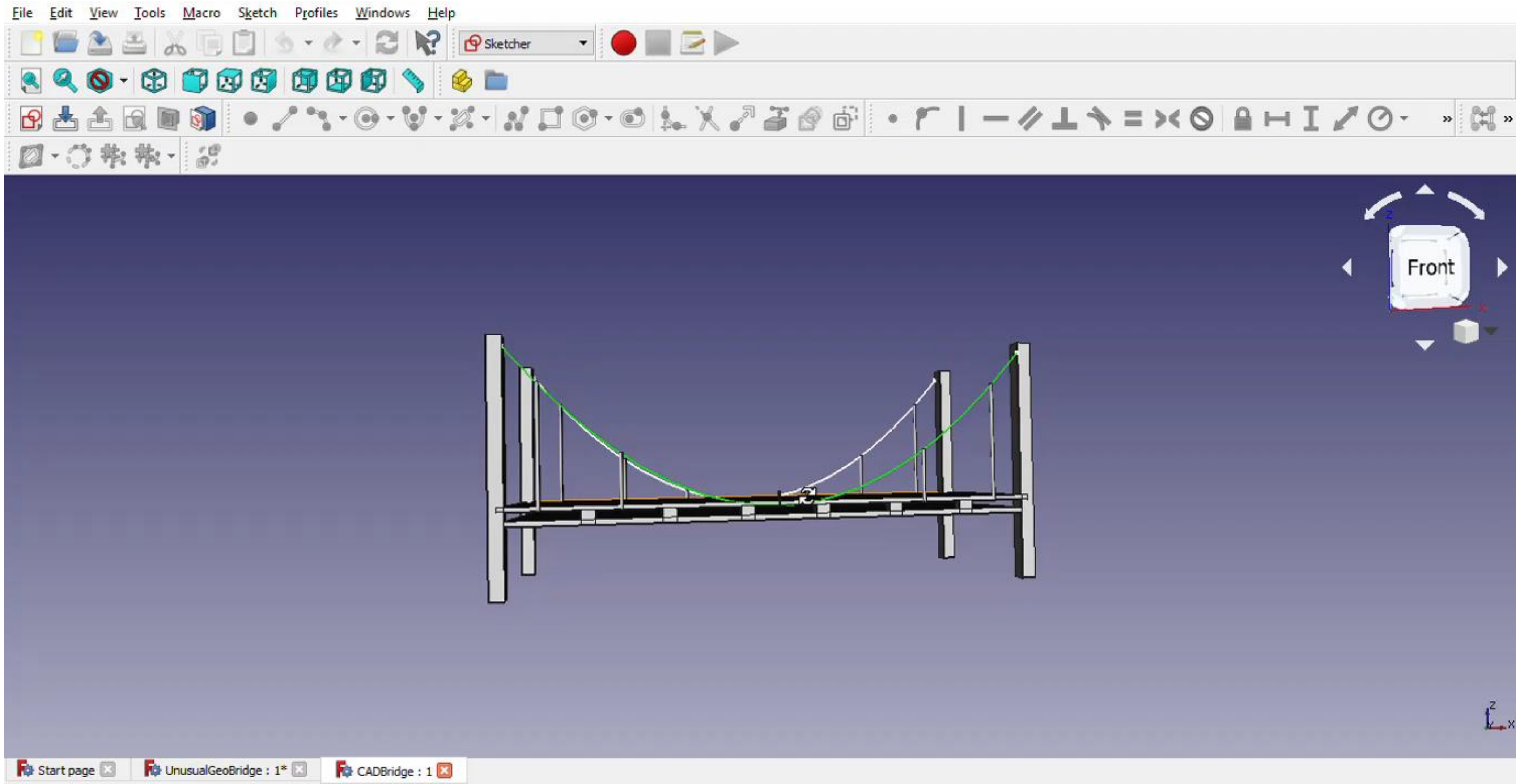




Michael L.
(CRSP mentee 2020 – 2021)

FreeCad Model Builder Extraordinaire

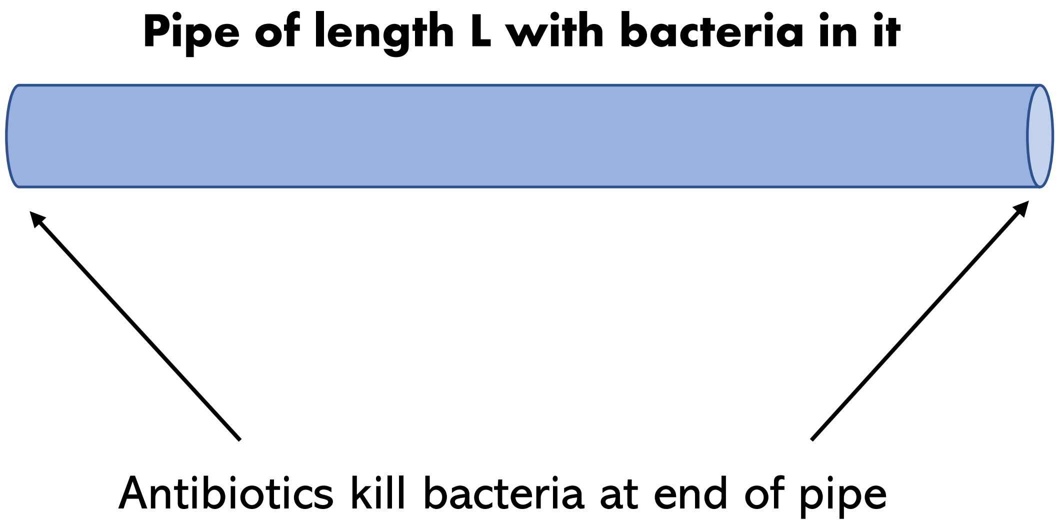
Michael L. (CRSP mentee 2020 – 2021)
FreeCad Model Builder Extraordinaire



Reaction Diffusion Model

Spring 2021 BMCC Foundation Fund
R.A. Samuel Boadu Amoako

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.



Samuel Boadu Amoako
Kaplan Leadership Program Scholar
POISE Program Scholar
Currently studying
Environmental Engineering (2022)

Reaction Diffusion Model

Spring 2021 BMCC Foundation Fund

R.A. Samuel Boadu Amoako

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:

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

Boundary Conditions

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

$c(x,t) =$ cocentration 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

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

Reaction Diffusion Equation

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

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

$$\approx \frac{\frac{c(x+\Delta x,t) - c(x,t)}{\Delta x} - \frac{c(x,t) - c(x-\Delta x,t)}{\Delta x}}{\Delta x}$$

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

We approximate $\frac{\partial^2 c}{\partial x^2}$ using Finite Differences

Reaction Diffusion Model

Numerical Solution Euler's Method

Reaction Diffusion Equation

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

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

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

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

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

Reaction Diffusion Model

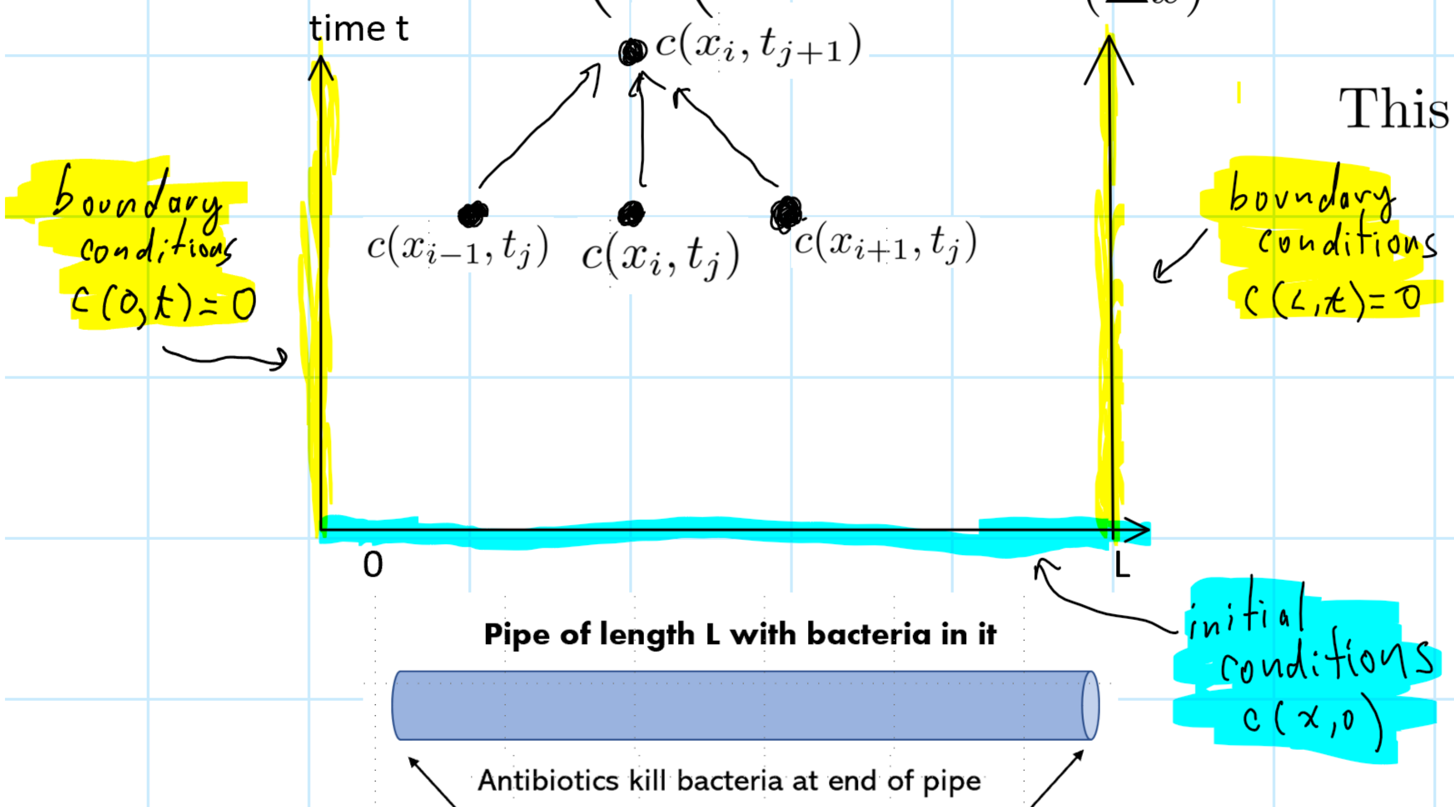
Numerical Solution Euler's Method

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

Reaction Diffusion Equation

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1-\frac{c}{K}\right)}_{\text{reaction term}}$$

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



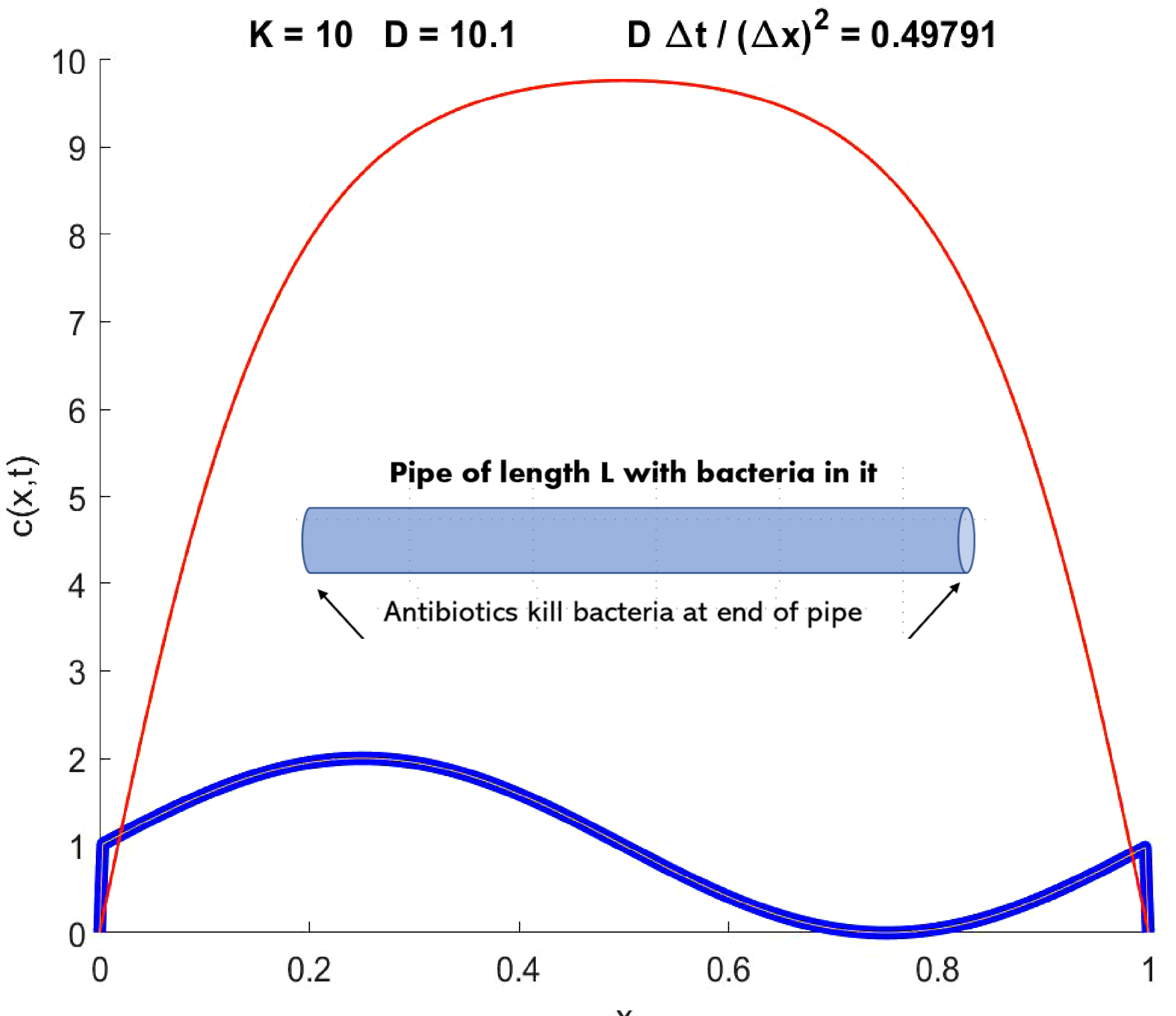
This finite difference numerical method works provided the Courant number:

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

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.

“convergent” Courant number = 0.49791 < 0.5

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

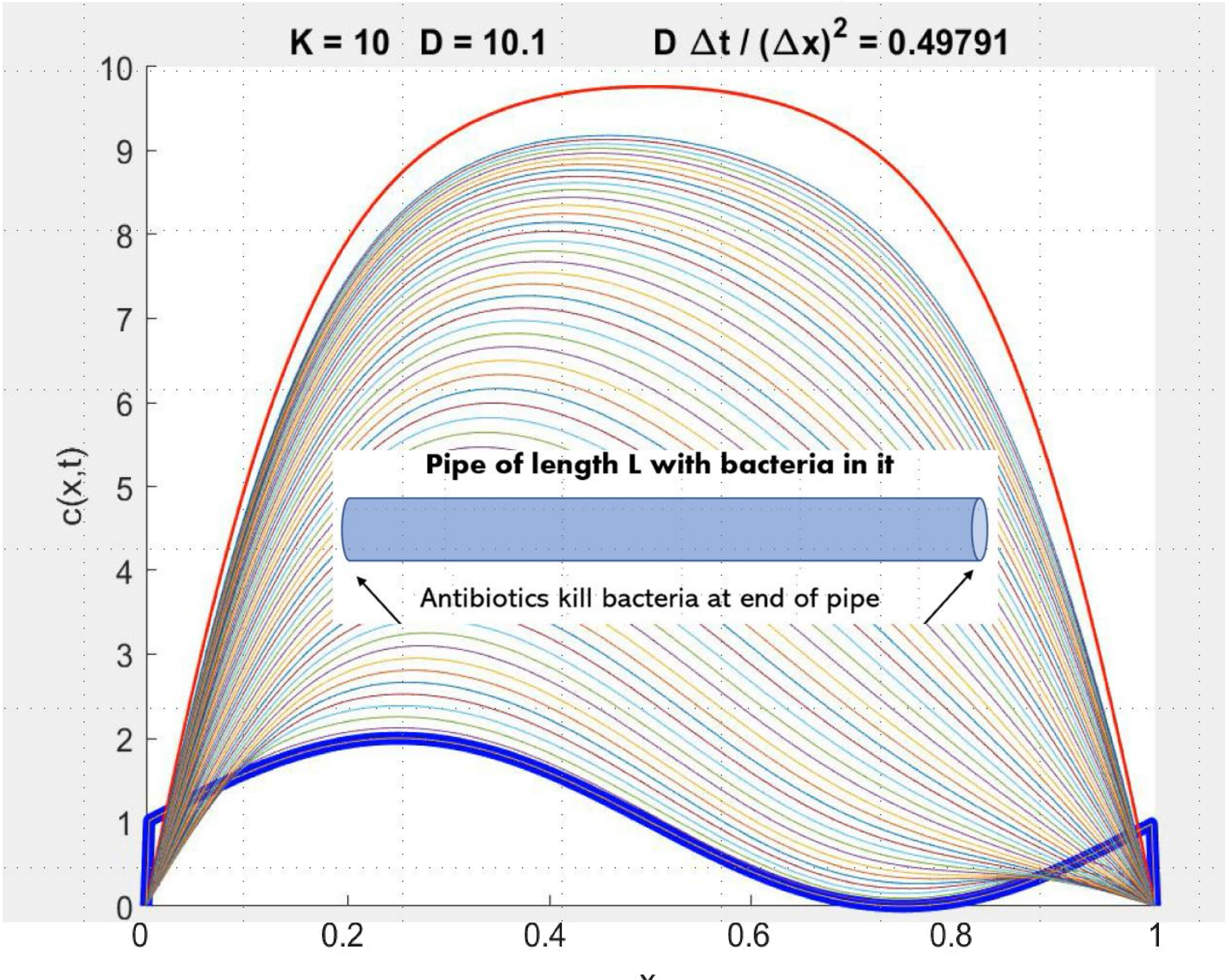
Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

"convergent" Courant number = 0.49791 < 0.5



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

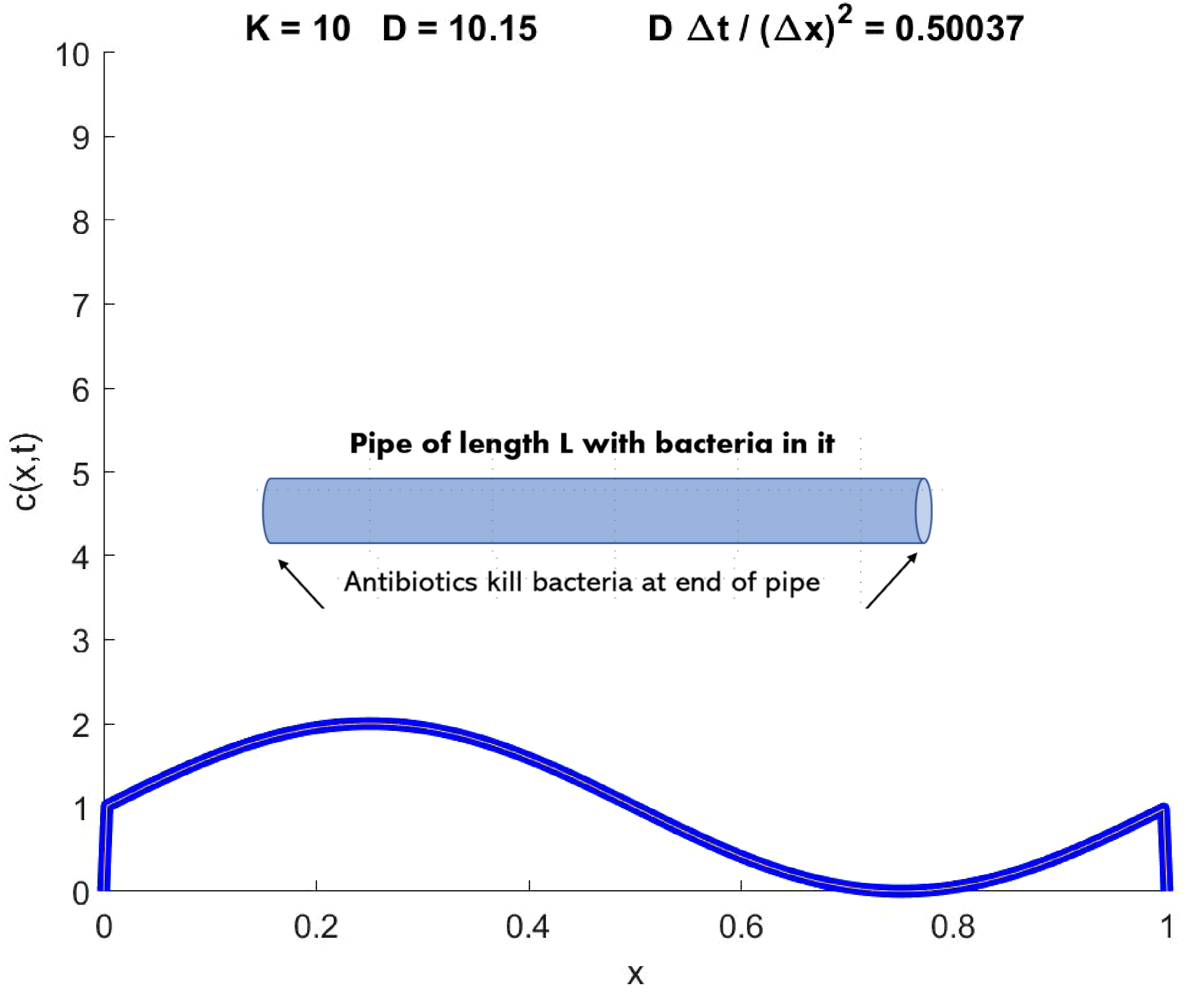
Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

"NOT convergent" Courant number = 0.50037 > 0.5



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

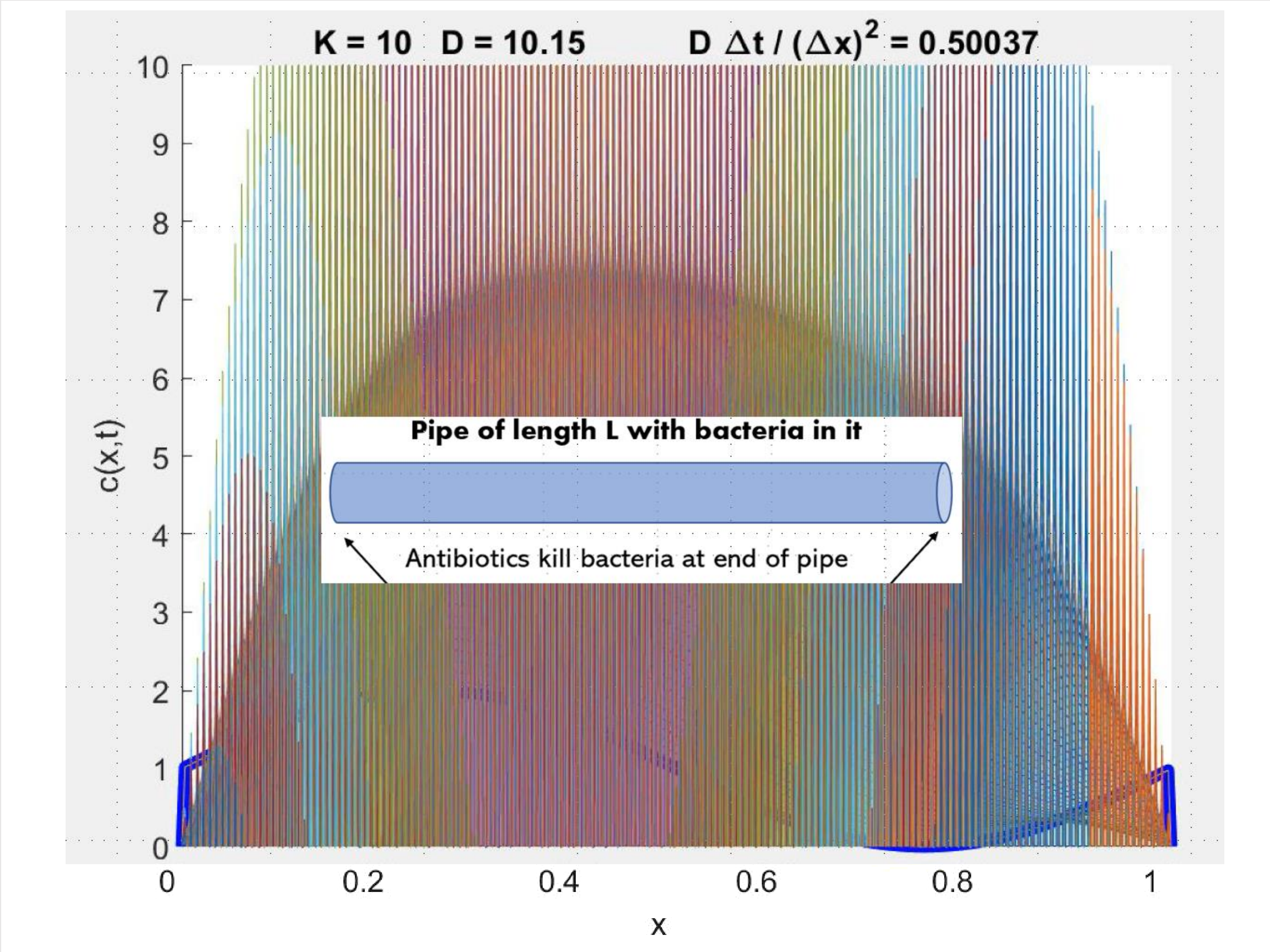
Blue = initial concentration of bacteria

Red = steady state solution

Animation produced in Matlab using Euler FD numerical method

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$

“NOT convergent” Courant number = 0.50037 > 0.5



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

Blue = initial concentration of bacteria

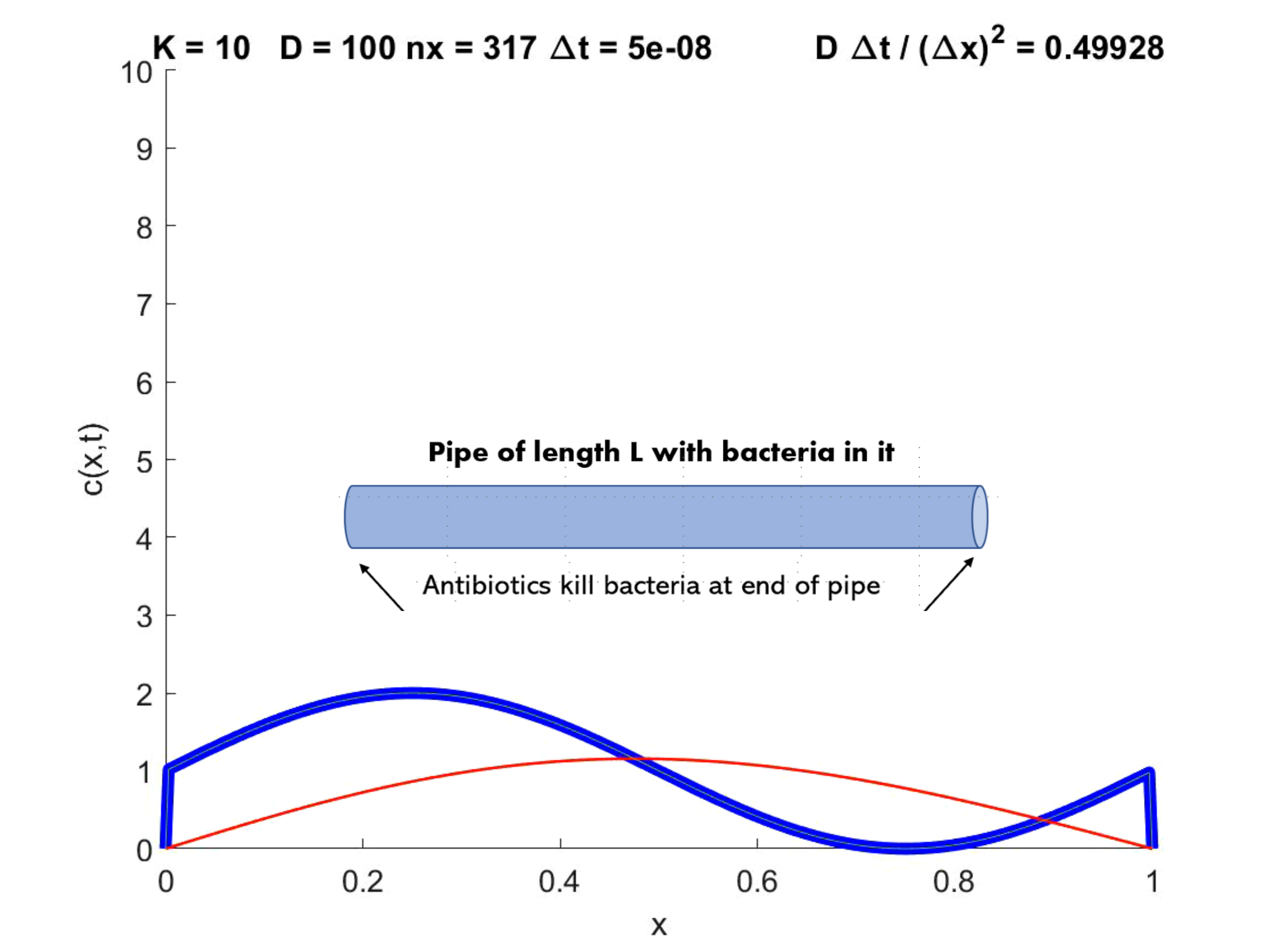
Red = steady state solution

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

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1-\frac{c}{K}\right)}_{\text{reaction term}}$$



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

Blue = initial concentration of bacteria

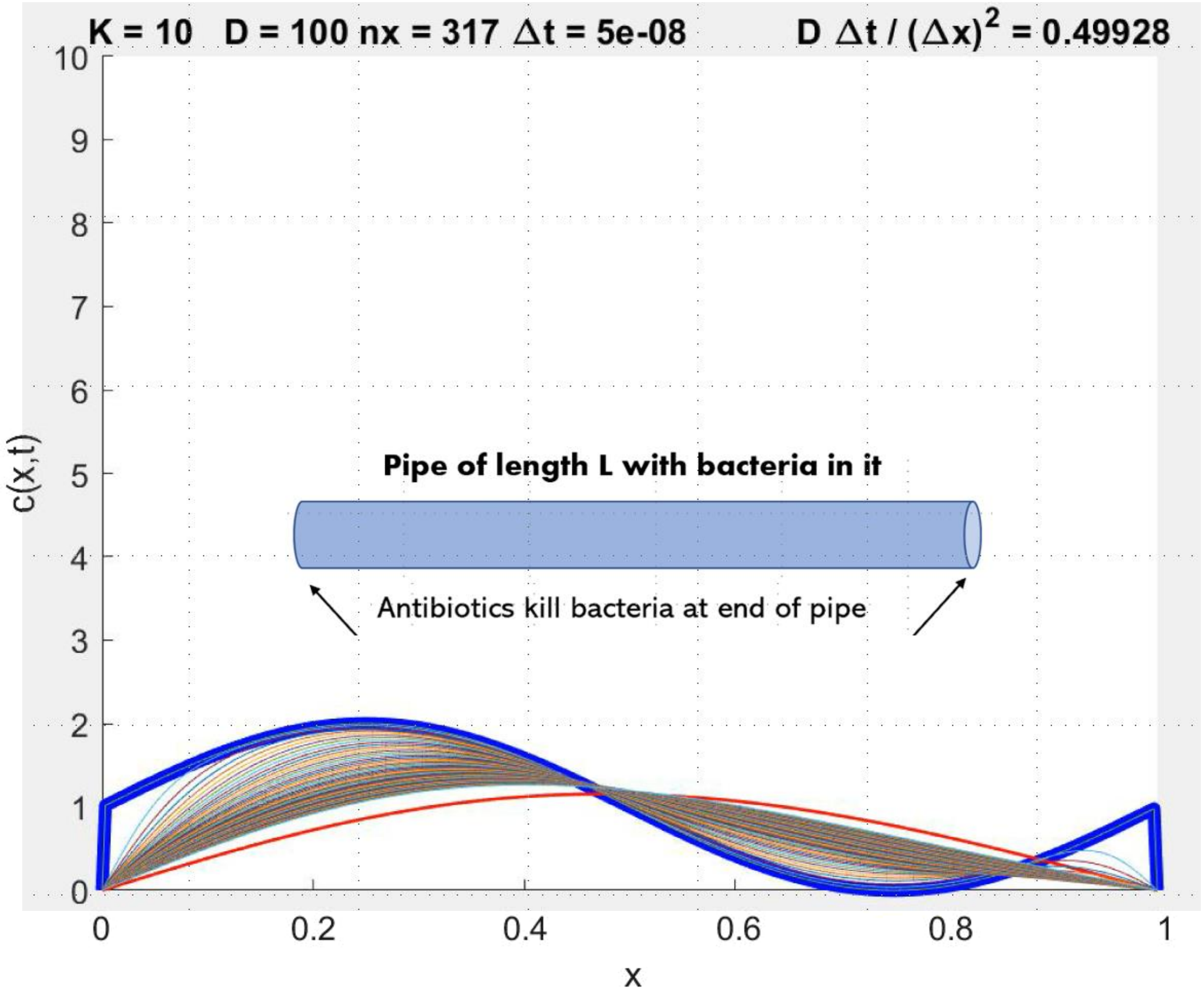
Red = steady state solution

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

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} + \underbrace{r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

Blue = initial concentration of bacteria

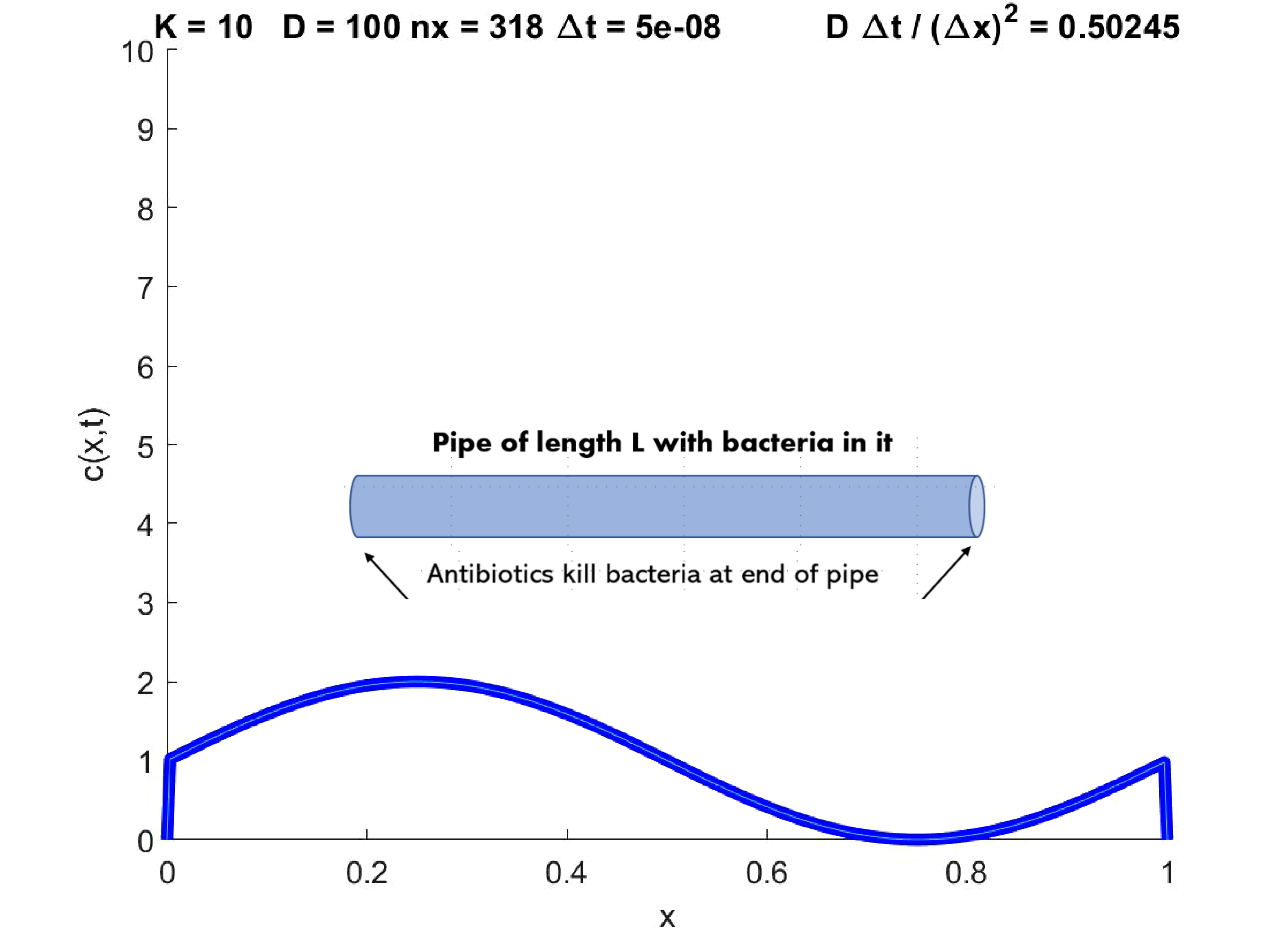
Red = steady state solution

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

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} \underbrace{+ r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$



Reaction Diffusion Equation Solution

Tube of length L = 1.

K = 10 = carrying capacity

Blue = initial concentration of bacteria

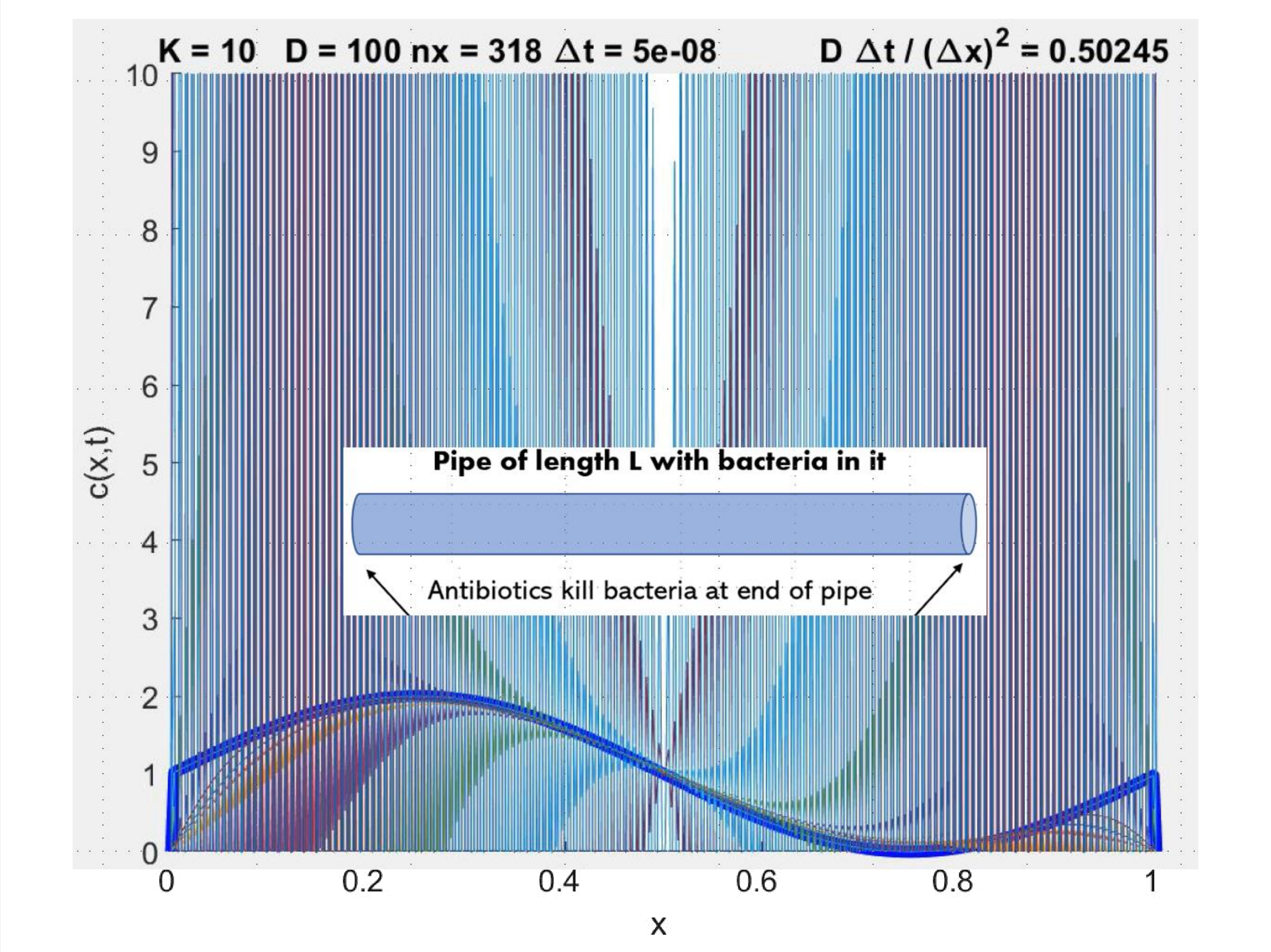
Red = steady state solution

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

$$\frac{\partial c}{\partial t} = \underbrace{D\frac{\partial^2 c}{\partial x^2}}_{\text{diffusion term}} \underbrace{+ r_0\, c\left(1 - \frac{c}{K}\right)}_{\text{reaction term}}$$



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 Equation

Samuel Boadu Amoako & Professor Chris McCarthy
Borough of Manhattan Community College/ Mathematics Dept.

Our Project

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:

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

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.

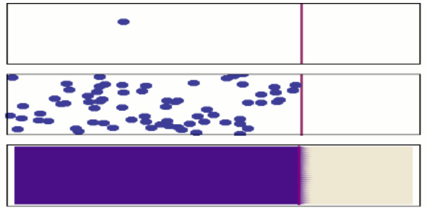
$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial t^2} + \frac{r_0}{K}c(K-c)$$

Boundary Conditions:
$u(0,t) = u(0,L) = 0$
bacteria are killed by antibiotics at the ends of tube $x = 0$ and $x = L$

c = concentration
x = position
t = time
D = diffusivity constant
K = concentration carrying capacity
r_0 = 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.

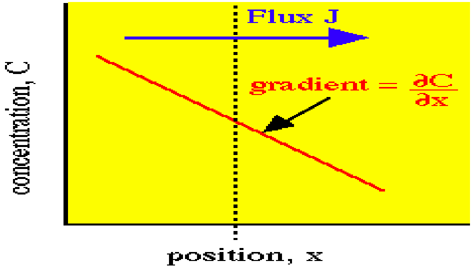
Background



The animation above shows random motion leads to diffusion

Diffusion is the movement of particles from a region of higher concentration to lower concentration. Mathematically diffusion occurs in response to a concentration gradient.

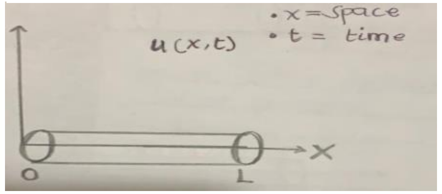
The figure below shows Fick's first law of diffusion: that the net flux (or flow of particles) is proportional to the negative gradient. The gradient is the slope of the concentration function. In this figure the slope (gradient) is negative. So, the net flow of particles is in the positive direction.



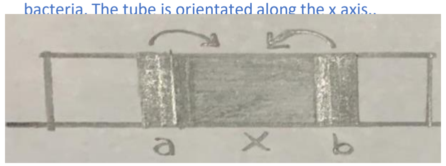
The logistic growth model for bacteria assumes that bacteria are less successful at reproducing as the concentration (density) of bacteria increases due to overcrowding and competition for resources. If the bacteria concentration exceeds the carrying capacity *K*, the bacteria will start to die off more quickly than they reproduce. As a result, in the logistic growth model, the carrying capacity is a stable equilibrium: the concentration of bacteria will tend to the carrying capacity.

Photo of e coli bacteria

Method and Results



The figure above represents the tube with the bacteria. The tube is orientated along the x axis..

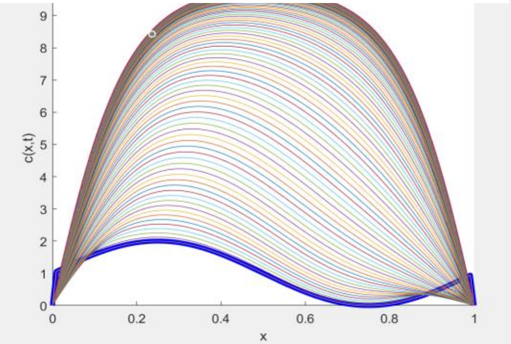


The figure above represents a section of the tube with diffusing bacteria.

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.

```
numx = 111;
numt =  20000;
dx = 1/(numx - 1);
dt = 0.0000005;
dt/dx^2
x = 0:dx:1;
C = zeros(numx,numt);
t(1) = 0;
C(1,1) = 0;
C(1,numx) = 0;
mu = 0.5;
sigma = 0.05;
for i=2:numx-1
   C(i,1) = exp(-(x(i)-
mu)^2/(2*sigma^2)) /
sqrt(2*pi*sigma^2);
end

for i=2:numx-1
   C(i,1) = 1+ 1*sin(x(i)*2*pi);
end
k=10; r = 100;
for j=1:numt
   t(j+1) = t(j) + dt;
   for i=2:numx-1
      C(i,j+1) = C(i,j) +
10*(dt/dx^2)*(C(i+1,j) - 2*C(i,j) +
C(i-1,j)) + r*C(i,j)*(k - C(i,j))*dt;
   end
```

```
plotNum = 20000;
C(:,plotNum);
max(C(:,plotNum));
min(C(:,plotNum));
figure(1);
hold on;
plot(x,C(:,1),'b', 'LineWidth',4);
plot(x,C(:,plotNum),'r' ,'LineWidth',1);
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

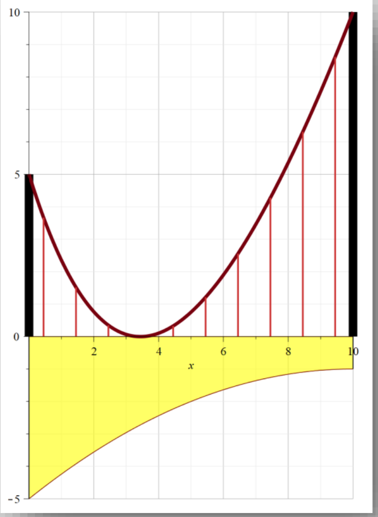
We were able to solve the reaction diffusion equation numerically and create an animation showing the concentration of bacteria over time. Regardless of the initial conditions we chose (dark blue curve), in the end, the bacteria concentration would take the form of an upside-down U shape. In the above 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

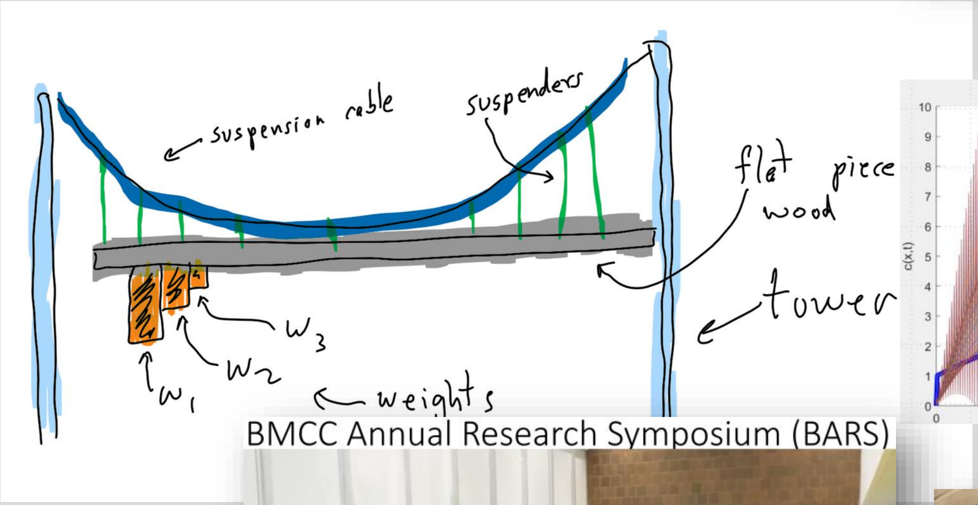
Special thanks to BMCC Foundation Fund for the funding and my mentor Chris McCarthy.

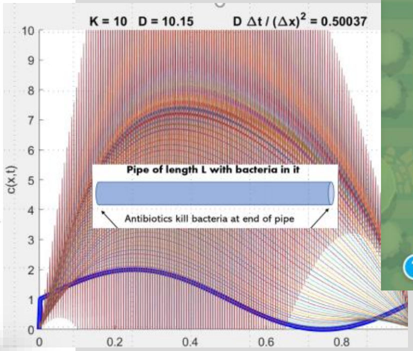
Professor Chris McCarthy
BMCC CUNY
cmccarthy@bmcc.cuny.edu

Please feel free to contact me! Thanks!!!



SIMIODE EXPO 2022





BMCC Annual Research Symposium (BARS)