Project Ideas

This page collects starting points for your project. The Project Specification describes what the project involves and when everything is due.

The ideas below are starting points. You are welcome to suggest your own topic in pure mathematics, applied mathematics, chemistry, biology, physics, climate, data science, or another area connected to differential equations. We can work together to turn an interest into a question of a manageable size.

A project can also be primarily computational. You might use Python to investigate a model, compare numerical methods, analyze measurements, or build a simulation that helps explain a mathematical idea. The interesting part is what the computation lets you discover and explain. The computational directions below are possibilities you can develop, not required tasks.

Pure Mathematics

Heat flow and Fourier series

What does the cooling of a metal rod have to do with adding sine waves? Explore how separation of variables turns the heat equation into ODEs, and how their solutions combine to describe an initial temperature profile. This gives a route into Fourier series through the problem that motivated Fourier's work. You could investigate the mathematics of these expansions, their historical development, or, if you like, use Python to visualize heat flow and compare Fourier solutions with numerical approximations.

Fourier series from vibrating chains

A chain of masses connected by springs has finitely many modes, and every motion of the chain is a sum of them. What happens to these modes as the chain becomes finer? Consider a chain of equal masses with both ends fixed, and find its eigenvectors explicitly: the displacements in each mode follow a sine curve sampled at the positions of the masses. As the number of masses grows, these samples fill in the functions $\sin(k\pi x)$. Use this to derive Fourier sine series. Expand an initial displacement of the chain in its eigenvectors, find the coefficients using the fact that the eigenvectors are perpendicular, and see what the expansion and the coefficient formulas become in the limit, where a sum over the masses turns into an integral. Which functions can be written this way, and in what sense does the series converge? You could also compare the chain's frequencies with those of a continuous string, or use Python to watch the expansions approach a given shape as the chain becomes finer.

Nonlinear equations in new variables

Linear differential equations come with powerful tools for finding and combining solutions. Nonlinear equations are usually harder. But could an equation's nonlinear appearance come from our choice of variable? The logistic equation contains a population squared; surprisingly, the reciprocal of that population satisfies a linear equation.

Explore how this substitution works and what it lets us solve. Begin by recovering the familiar logistic solution, then allow the growth rate and carrying capacity to vary with time. Separation of variables generally stops working, but the reciprocal substitution still produces a linear equation. Use this to find solutions beyond the constant-environment model studied in class.

Then investigate other nonlinear equations that become linear after a substitution. Explain how to choose the new variable, derive the transformed equation, and translate its solutions back. Pay attention to where that translation fails: taking a reciprocal leaves out the zero-population solution, and a transformed solution passing through zero can correspond to the original solution becoming unbounded. How can a linear equation whose solutions exist for all time describe a nonlinear equation whose solutions blow up?

Matrix Lie groups

Possible directions to revisit:

- Connect linear ODEs and the matrix exponential to matrix Lie groups and their Lie algebras, using concrete examples such as rotations.
- Investigate how conditions on the coefficient matrix determine which geometric quantities its flow preserves, such as lengths or volume.

Numerical Methods

Adaptive Euler methods

How should a numerical method decide how large a step to take? Small steps can improve accuracy, but they also mean more computation. Explore how comparing one Euler step with two half-sized steps lets a method estimate its own error and adjust its step size as it goes. You could build an adaptive solver in Python and investigate when it improves on fixed-step Euler, using equations with known solutions to compare accuracy and computational effort. Does controlling the error in each step keep the accumulated error small? You could also investigate what adaptive stepping does to an oscillator's energy, or compare it with an improved step formula such as Heun's method.

Computing the matrix exponential

The series defining e^A converges for every square matrix, so why not compute the exponential by adding up its terms? Try it in Python, beginning with a single number: what happens when you sum the series for e^{-25}? Large terms of opposite signs cancel, and the computer's rounding errors

are all that survive. Explore why this happens and what to do instead. Investigate other ways to compute the exponential: diagonalizing the matrix, which fails when its eigenvectors are nearly parallel; taking many Euler steps, as in the limit $(I + A/n)^n$; and scaling and squaring, which approximates $e^{A/2^k}$, whose argument is now small, and then squares the result k times. Compare the accuracy and cost of these methods on matrices whose exponentials you know exactly. Why does scaling and squaring work so well, and how should the number of squarings be chosen? You could read Moler and Van Loan's classic paper "Nineteen Dubious Ways to Compute the Exponential of a Matrix" and compare your methods with the one built into SciPy.

Data Science

Recovering model parameters from data

How can we determine the constants in a differential equation when all we can measure is its solution? Explore two approaches: estimating derivatives from observations and fitting the equation to those estimates, or solving the equation repeatedly and adjusting its parameters until the solutions fit the observations. Choose a model and compare how well these approaches recover its parameters. Begin with simulated measurements, where you know the correct answers, and investigate what happens when you add noise or take measurements less frequently. Then apply the methods to experimental data and test their predictions against observations left out of the fit. Which approach gives more reliable predictions, and what measurements does it need?

Learning a differential equation with a neural network

Can a neural network learn a law of motion from observed trajectories? Instead of training it to predict position directly, train it to predict the derivative of the state. An ODE solver then turns that learned rule into a trajectory, which can be compared with the observations.

Begin with simulated position and velocity measurements from a damped oscillator, where the true equation is known. Train a small network to reproduce several trajectories, then test it on initial conditions excluded from training and over longer time intervals. Does fitting the observations also recover the correct direction field and eventual decay? Investigate how the number of training trajectories affects these predictions. Finally, repeat the predictions with smaller solver steps: has the network learned a useful differential equation, or does its apparent accuracy depend on the numerical method used during training?

This computational project is intended for students who have taken the Math 370 series and have experience training neural networks and using automatic differentiation.

Computer Science

Ranking a network with differential equations

How can we rank webpages when a link from an important page should count more than a link from an obscure one? Explore PageRank by imagining visitors moving through a network, sometimes following links and sometimes jumping to another page. Derive differential equations for the fractions of visitors on each page, and investigate how their equilibrium determines a ranking. On small networks, compare this ranking with simply counting incoming links. Explain why the scores settle down and investigate how changing links or the frequency of independent jumps changes the result. Show how Euler's method turns the differential equation into an algorithm for computing the ranking.

Physics

Earthquakes and buildings

Why can the same earthquake produce very different motions in different buildings? Explore a building's vibrational modes and how ground motion excites them. If you like, Python could help you simulate the response, animate the building, or work with recorded earthquake data. You could also investigate an engineering technique such as a tuned mass damper, flexible bearings at the foundation, or reinforcement of selected parts of the structure. Incorporate the technique into a model and study its effect theoretically or through simulation.

The nonlinear pendulum

Does a pendulum take the same time to swing back and forth when released from a larger angle? Use conservation of energy to express its period as an integral, and investigate how the period changes with amplitude. Compare the result with the small-angle approximation and explain what happens as the release angle approaches the upright position. Then add friction and study how the amplitude and the time between successive swings change as the pendulum loses energy. You could compare friction proportional to velocity with drag proportional to the square of speed, use Python to simulate the motion, or test the predictions against measurements from a real pendulum.

Light near a black hole

Gravity bends the path of light, and near a black hole the bending can be extreme. Light can be deflected through large angles, captured entirely, or even sent around the black hole in a circle. Explore these possibilities using the equations general relativity gives for light passing a nonrotating black hole. These equations describe the light ray's distance r from the black hole and its angle ϕ, both as functions of a parameter that marks progress along the ray.

Begin by changing variables. Use the angle in place of the parameter and the reciprocal $u = 1/r$ in place of the distance. Show that the path satisfies a first-order equation relating u to its derivative

and, after differentiating, a nonlinear oscillator equation in which the angle plays the role of time. What are its solutions when the black hole has no mass, and what does the new nonlinear term do to them? Then use these equations to analyze the motion of light. Find the circular orbit for light, called the photon sphere. Use the first-order equation to decide which incoming rays escape and which fall in; the boundary between them determines the size of the black hole's shadow. For rays passing far from the black hole, find an approximate solution by guessing and checking, and use it to predict how much starlight is bent as it passes the Sun. Compare your prediction with the 1919 eclipse measurements that first tested general relativity.

You could use Python to compute the exact deflection and see where the approximation fails, to follow rays that pass close to the photon sphere, or to render an image of a black hole in front of a field of stars.

Meeting another spacecraft in orbit

How do you catch up with a spacecraft orbiting ahead of you? The obvious plan, pointing toward it and firing your engine, does not work the way you might expect: you rise above the target and, before long, begin falling behind it. Explore why, using the Clohessy–Wiltshire equations, which describe one spacecraft's motion relative to another in a nearby circular orbit. These equations come from Newton's law of gravity, written in a rotating frame and approximated near the target, and you can take them as given.

Within the orbital plane, the relative motion is a constant-coefficient linear system in four variables. Find its invariant planes: in one the motion rotates, and in the other it shears. Use them to explain the strange effect of a forward thrust, and to find which starting states return to where they began after each orbit and which drift away. Then use the flow to plan a rendezvous. Given your current position and the time allowed, solve a linear system for the change in velocity that brings you to the target, and find the second change that stops you there. How does the required thrust depend on the time you allow? You could use Python to simulate and animate different approaches, or read about how this effect contributed to Gemini 4's failed attempt to fly alongside its booster rocket in 1965.

Traffic jams without a cause

In 2008, researchers in Japan asked 22 drivers to circle a track 230 meters around at a steady speed. Within a few minutes a traffic jam formed and moved backward around the track, although there was no accident, no merging lane, and nothing in the road. Explore how such phantom jams can arise from the way drivers respond to the car ahead. In the optimal velocity model, each driver accelerates toward a preferred speed that depends on the gap to the next car. Evenly spaced cars all moving at the same speed form an equilibrium. Replace the preferred-speed rule by its tangent line at the equilibrium gap to obtain a linear system for small disturbances to this steady flow.

Because the cars sit on a ring and all follow the same rule, this system has special structure. As for a ring of identical masses joined by springs, it separates into patterns that travel around the ring. But each driver responds only to the car in front, so these patterns lie in invariant planes where disturbances rotate while they grow or shrink. Find these planes and determine when one of them

grows, making steady traffic unstable. How does the answer depend on how quickly drivers react, how strongly their preferred speed depends on the gap, and how many cars share the road? Then use Python to simulate the full model from a state near steady flow, and watch whether a jam forms and how it moves. You could compare your simulations with videos of the experiment.

Chemistry

Molecular vibrations

Why do molecules absorb particular frequencies of infrared light? Their atoms vibrate in collective patterns, much like the coupled springs we study in class. Explore models of carbon dioxide or water, calculate their modes, and compare predicted frequencies with experimental measurements. Investigate why some modes interact with light while others do not, and how this connects molecular motion to the greenhouse effect. If you like, Python could help you calculate and animate the modes or compare the model with data.

Chemical oscillations

How can chemical concentrations oscillate when the supply of reactants is steady? Explore an idealized reaction model such as the Brusselator, using the law of mass action to turn its reaction steps into differential equations. Determine when its equilibrium is stable, then investigate how changing the supplied reactant concentrations can produce sustained oscillations. Use Python to follow the concentrations through time and study the attracting cycle in the phase plane.

Biology

Understanding an epidemic

What determines how large an outbreak of an infectious disease becomes? The SIR model divides a population into people who are susceptible, infectious, and recovered, and describes how these groups change as susceptible people catch the disease and infectious people recover. Explore what these equations predict about an epidemic. Begin by eliminating time to obtain an equation relating the susceptible and infectious populations. Integrate it to find a quantity that stays constant along each solution, and use its level curves to trace the possible paths of an outbreak. Determine the peak number simultaneously infectious and derive an equation for the total number infected over the outbreak. Why can an epidemic end while susceptible people remain?

Investigate how these predictions change with the initial populations and the transmission and recovery rates. You could study vaccination before an outbreak, assuming it gives complete immunity: how does moving people from the susceptible group to the immune group change the solution curve, its peak, and its final size? Explain what the conserved quantity lets you predict and what still requires information about time, such as the date of the peak.

Ecology

How much can we harvest?

How can measurements of a fish population help us choose a fishing quota without emptying the pond? Starting from the logistic growth model, generate simulated population measurements and use them to estimate the growth rate and carrying capacity. Add a constant harvesting rate and use your estimates to recommend a quota. Find the population threshold below which that quota leads to collapse: even a harvest smaller than the maximum natural growth rate can be too much for a depleted population. Then test your recommendation when the fish grow more slowly, or the initial population is smaller, than your measurements suggested. How much should you reduce the quota to allow for those errors? You could use Python to fit the model and compare the resulting population histories, or investigate when to take measurements to learn more about the growth parameters. The goal is a harvesting recommendation supported by your model, together with an explanation of when it would fail.

Climate Science

Seasonal temperature lag

Why does the warmest part of the year usually come after the summer solstice? Build a heat-balance model with seasonally varying sunlight, heat storage, and heat loss. Solve the resulting differential equation to predict both the size of the temperature variation and its delay behind the solar input. Investigate how heat capacity changes these two predictions, and use the model to explore differences between land and ocean temperatures. You could compare its predictions with observed seasonal cycles, or use measured amplitudes and delays to estimate the model's heat capacity and heat-loss coefficient.

When do emissions cuts lower atmospheric CO_2?

Reducing the amount of CO_2 we emit each year does not necessarily reduce the amount already in the atmosphere. Some of each year's emissions remain while more are added. How large a reduction would make the atmospheric amount start falling?

Begin with a published model describing how an isolated release of CO_2 is gradually absorbed by land and ocean. Its response combines several exponential decays with different timescales and a component that persists over the model's time horizon. Express these contributions as solutions of linear differential equations, then include continuing emissions. Compare what happens when emissions remain constant, fall by half, or stop. Determine when the atmospheric amount begins decreasing and investigate how the answer depends on the earlier emissions history. Explain why replacing the different removal timescales with a single exponential changes the predictions.

Warming after CO_2 stabilizes

If atmospheric CO_2 stops increasing, does Earth immediately stop warming? The ocean stores heat, and its deeper layers adjust much more slowly than the surface. Explore how this difference affects the warming that remains after CO_2 concentrations stabilize.

Build a model with two temperatures, one for the surface and one for the deep ocean, using heat balances to describe their exchange of energy. Compare a sudden increase in greenhouse heating with a gradual increase to the same final level. Solve the resulting linear system to determine the eventual temperature and how far the surface remains from it when the heating stops increasing. Investigate how ocean heat capacity and heat exchange change the adjustment times. Can two temperature histories approach the same equilibrium while predicting substantially different warming over the next several decades?