DATA 420 Notes – Jan. 30, 2024

Converting between time and index

Recall that every element of a function/vector can be identified in two ways: by its index:

`f[i]`

or by its x value (which is usually a time t):

f(x)

It will sometimes be necessary to convert back and forth between these two specifications. Given an index i, what time point does that correspond to? Well, the Δx (`delta_x`) factor is what determines how much time each point is "worth" on the scale, so we need to multiply i by Δx. Also, our starting point may or may not be 0 – sometimes (like in the example for today) it will be something else. So we need to add the starting time (`start_x`) to this total in order to get everything shifted correctly. The final formula, then, is:

$$x = (i \times \Delta x) + \texttt{start\_x}$$

Convince yourself that the units work out correctly. Remember that i is "unitless."

Going the other way is just solving for i:

$$i = \frac{x - \texttt{start\_x}}{\Delta x}$$

Again, convince yourself that the units work out correctly.

In Python, therefore, we'll define these two functions:

```
def itox(i):
    return delta_x * i + start_x

def xtoi(x):
    return int((x - start_x) / delta_x)
```

which will be convenient for us.

Aliens vs. Vampires

Aliens

Suppose that in 1940, UFOs from outer space started descending on the earth and abducting humans. Let A be a function/vector that gives *the total number of abductees at a given index/time*. To make things easy, we'll use "1 year" as our Δx. This means that `A[0], A[1], A[2], ...` correspond to $A(1940), A(1941), A(1942), ...$

Constant rate of abduction

Let's start by assuming that *the rate of abduction has been constant since 1940*, say, 1000 $\frac{\text{abductees}}{\text{year}}$. In symbols:

$$A'(x) = 1000$$

and to give the first few values:

$$\begin{aligned} A'(1940) &= \texttt{A\_prime[0]} = 1000 \\ A'(1941) &= \texttt{A\_prime[1]} = 1000 \\ A'(1942) &= \texttt{A\_prime[2]} = 1000 \\ A'(1943) &= \texttt{A\_prime[3]} = 1000 \\ &\dots \end{aligned}$$

(Recall that "A'," pronounced "A-prime," means "the derivative of A," or "the rate of change of A over time.")

Now to find the *total* number of people aboard spaceships in a given year, we need not A', but A. This is just the integral, of course. Assuming no one had been abducted prior to 1940, and thus 0 is our initial condition, we have:

$$\begin{aligned} A(1940) &= \texttt{A[0]} = 0 \\ A(1941) &= \texttt{A[1]} = 1000 \\ A(1942) &= \texttt{A[2]} = 2000 \\ A(1943) &= \texttt{A[3]} = 3000 \\ &\dots \end{aligned}$$

With an equation this simple, we could actually compute an analytical, closed-form expression for the integral, which I leave as an exercise if you're interested. For now I'll just say that in general, ain't nobody got time for that.

Increasing rate of abduction

Maybe, though, once the aliens realize that the gettin's good on planet Earth, they start increasing the number of spaceships they send, and therefore the *rate* of abduction increases over time.

(Note very very carefully that the *total* number of abductees was *already* increasing over time in the previous section. So having the total be increasing is not new. What's new now is that the *rate itself* will be increasing.)

Let's say that every year, the *rate* increases by an *additional* 1000 $\frac{abductions}{year}$. Soon, the night sky is studded with mother ships. Our equation for the rate is now:

$$A'(x) = 1000 \cdot x$$

indicating that for every year that passes we have a higher abduction rate:

$$\begin{aligned} A'(1940) &= \texttt{A\_prime[0]} = 1000 \\ A'(1941) &= \texttt{A\_prime[1]} = 2000 \\ A'(1942) &= \texttt{A\_prime[2]} = 3000 \\ A'(1943) &= \texttt{A\_prime[3]} = 4000 \\ &\ldots \end{aligned}$$

The totals are now:

$$\begin{aligned} A(1940) &= \texttt{A[0]} = 0 \\ A(1941) &= \texttt{A[1]} = 1000 \\ A(1942) &= \texttt{A[2]} = 3000 \\ A(1943) &= \texttt{A[3]} = 6000 \\ A(1944) &= \texttt{A[3]} = 10,000 \\ &\ldots \end{aligned}$$

Units check: what are the units on the number 1000 in our original expression for $A'(x)$? Answer: $\frac{\text{abductions}}{\text{year}^2}$. This may seem hard to wrap your head around — what the heck is a "year squared," anyway? It's easier to understand, I think, if you see it like this:

$$\frac{\frac{\text{abductions}}{\text{year}}}{\text{year}}$$

"For every year that passes, how many additional abductions-per-year will occur?"

Vampires

Okay, vampires. These undead beings operate a little differently in that instead of being a different species that preys on humans, they actually turn humans *into* other vampires when they bite them. This means that the rate of "vampirization" is going to depend on *the number of vampires that already exist.* When there's only a few vampires, they can't bite that many additional people. But as they grow more numerous, their rate of "recruiting" additional vampires will also grow.

A reasonable equation for the rate-of-vampirization would be:

$$V' = .1 \cdot V$$

where .1 could perhaps be considered a "bloodthirstiness" parameter that models how frequently a vampire will bite a victim. Units check: what are the units for this "bloodthirstiness?" A cold, calculating unit analysis will tell you: $\frac{1}{year}$. Uh...a "per-year?"

Much easier to see this way:

$$\frac{\frac{\text{(new) vampires}}{\text{year}}}{\text{(existing) vampire}}$$

"For every vampire we already have, how many additional vampires per year will this result in?" Note that this boils down to a "per-year" since the vampires cancel out in numerator and denominator.

In the present case, we're saying that each vampire, on average, bites one victim every ten years or so.

Our V' vector is now a bit more complicated to compute, since it depends on the number of vampires V instead of just on time. Hence, we can't precompute all the V' values ahead of time. Instead, we'll alternate: computing first a $V'(x)$, then the corresponding $V(x)$.

What's our initial condition, though? If we make it 0, then this whole enterprise is fruitless – there aren't any vampires at all to bite any other vampires, so there will never be any vampires! (Some planets are actually like this.) Let's start with 1, then, to kick off the process.

$$
\begin{aligned}
V(1940) &= \texttt{V[0]} = 1 \\
V'(1940) &= \texttt{V\_prime[0]} = .1 \\
V(1941) &= \texttt{V[1]} = 1.1 \\
V'(1941) &= \texttt{V\_prime[1]} = .11 \\
V(1942) &= \texttt{V[2]} = 1.21 \\
V'(1942) &= \texttt{V\_prime[2]} = .121 \\
V(1943) &= \texttt{V[3]} = 1.33 \\
&\ldots
\end{aligned}
$$

and on it goes.

Implementation

Page 9 depicts an entire simulation in Python for the aliens vs. vampires model. Study it carefully. (To increase the resolution of the simulation, I changed Δx from 1 year to $\frac{1}{365}$ year, so we're re-computing our quantities every day.)

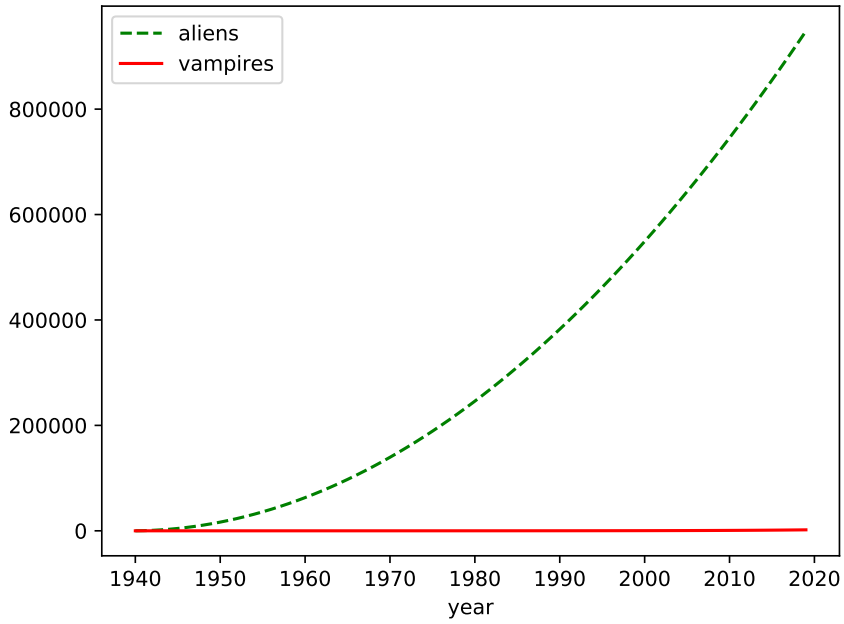
Note that in this simulation, we're computing each "prime" value (A' and V') on the fly, then throwing it away. We could, of course, keep them around in their own vectors, if we wanted to analyze the patterns of the rates of change at the end. Occasionally we'll do this.

Throwdown

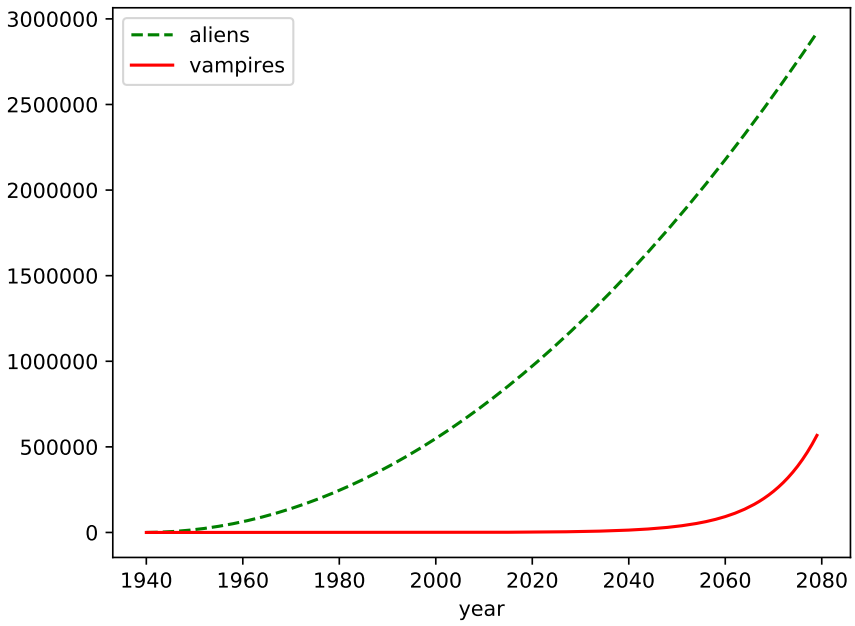
So who wins the battle for the human race? Aliens or vampires?

If you're like most people, you predict "aliens" without a second thought. After all, they're abducting *thousands* of people every year, and this rate is *increasing* every year. Those dumb vampires just can't even get off the dime – what are they think biting only one person every *ten* years, for heaven's sake?!

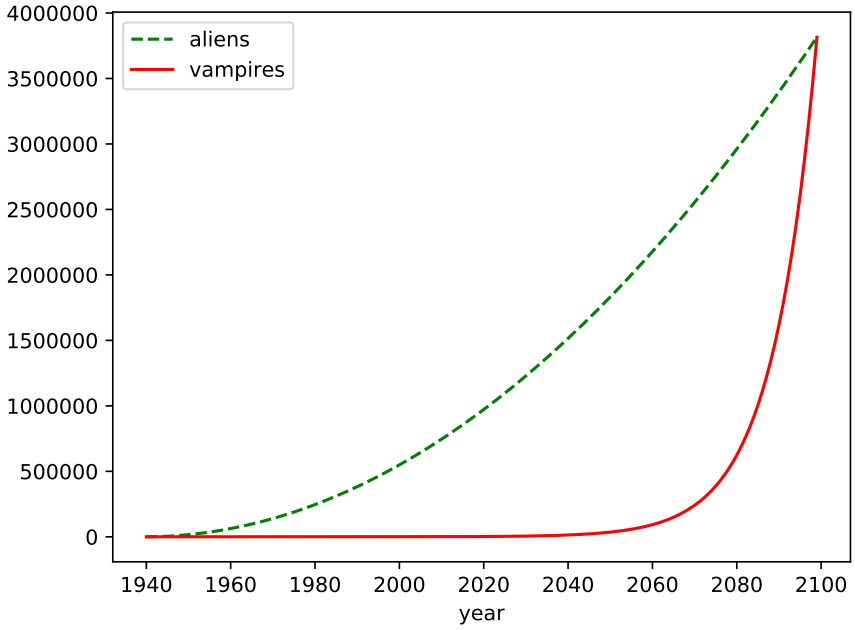
And indeed, if you plot the results over a relatively short time period, that's what you see: the aliens seem to be leaving the vampires in the dust.



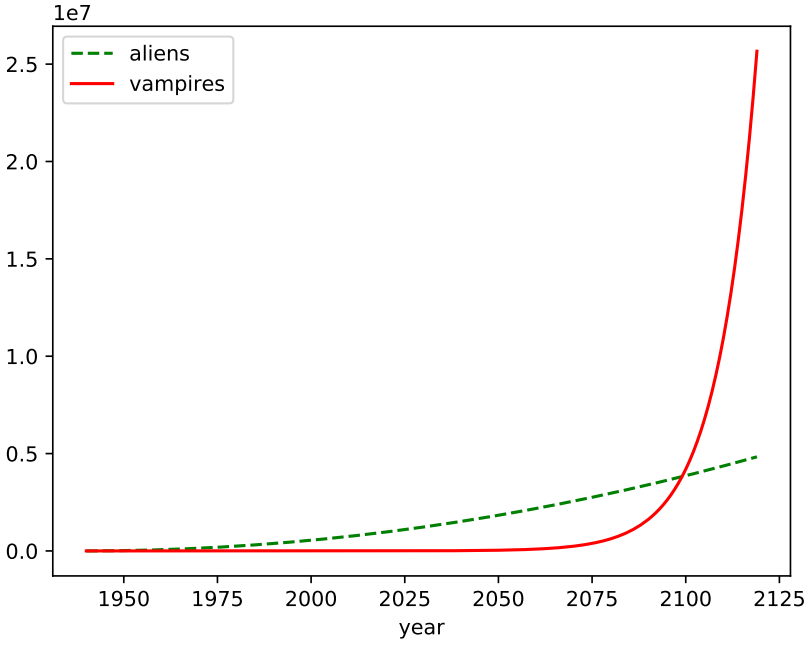
No contest. Keep running the simulation, though, and you'll see something odd happening around the year 2060...



Hold the fort, folks: the vampires seem to be showing signs of life! Let's keep simulating...



Tie ball game. And you can see where this is going:



It's a romp. In a hundred years or so, vampires take over the earth easily, leaving the aliens with no one to harvest (unless they like to abduct vampires, which I guess is okay.)

Lessons learned

Why is this so surprising? Probably because although humans are pretty good at comparing the magnitudes of *numbers* (1,000 sure looks a lot bigger than .1) they aren't so good at comparing *growth processes*. In this case, although both the aliens and the vampires presented roughly the same kind of "increasing curvy concave-up line" behavior, they are in fact totally different.

The aliens exhibited **polynomial growth**. That's what you're going to get when your "prime vector" is a constant, or a function of x (or even a power of x). Doesn't matter what the constants are, which means it doesn't matter how much your quantity increases, or *even how much you increase the rate at which it increases.* You're going to get behavior which is mathematically a polynomial: a power of x.

The vampires, on the other hand, exhibit **exponential growth**. And that is the type of growth that crushes all others in its wake. Why do the vampires grow exponentially, whereas the aliens do not? The answer is simple and worth turning over in your mind:

> Whenever the *rate* at which a quantity increases depends on *the amount that's already there*, you have exponential growth.

This was the situation with the vampires. We had V on the right-hand side of the equation for V'. That's the tell-tale sign of exponential growth. We'll see it again and again this semester.

```
import numpy as np
import matplotlib.pyplot as plt

delta_x = 1/365    # years
start_x = 1940     # year
end_x = 2122       # year

# Given an array index, return the corresponding time (year).
def itox(i):
    return delta_x * i + start_x

# Given a point in time (year) return the corresponding array index (unitless).
def xtoi(x):
    return int((x - start_x) / delta_x)

# How aggressive are the aliens? At what rate do they abduct victims for every
# year after 1940?
aggressiveness = 1000   # (abd/year)/year

# How thirsty are the vampires? How many does each one bite, on average, in a
# year?
bloodthirstiness = .1   # (vampires/year)/vampire

# Our x-values: the precise times at which we will measure and compute the
# quantities of interest.
x = np.arange(start_x, 2122, delta_x)   # years

# Our "stock variables": one for the number of UFO-abducted victims, and one
# for the number of vampires. They are arrays, of course, because we want to
# track how many of each quantity there are *at each time period*.
A = np.zeros(len(x))   # abductions
V = np.zeros(len(x))   # vampires

# Set our initial conditions. When the simulation begins at the stroke of
# midnight New Year's Day 1940, there's nobody on spaceships yet, and there's
# one lonely vampire in the world.
A[0] = 0
V[0] = 1

# The main simulation loop. For each time period...
for i in range(1, len(x)):

    # ...calculate the rate of each quantity at this point in time, and...
    Aprime = (itox(i) - start_x) * aggressiveness   # abd/year
    Vprime = V[i-1] * bloodthirstiness              # vamp/year

    # ...increment the quantities by that amount, times our time period.
    A[i] = A[i-1] + Aprime * delta_x                # abd
    V[i] = V[i-1] + Vprime * delta_x                # vamp


# Let's see what this bad boy looks like.
plt.plot(x, A/1e6, color="green", linestyle="dashed", label="alien abduction")
plt.plot(x, V/1e6, color="red", label="vampires")
plt.xlabel("year")
plt.ylabel("millions of abductees / vampires")
plt.legend()
plt.show()
```