Math 42 HW 6

stewartben02

May 2022

1

I found the method of using slope fields to model unsolvable differential equations very interesting considering that most differential equations cannot be solved. By simply following the path of the derivative we are able to accurately model a situation without explicitly solving the differential equation. By picking a starting point on a slope field, we can map the progress of a model by moving along the direction of the slope field of the given differential.

2

$$\frac{dx}{dt} = rx(1 - \frac{x}{K})$$

$$\frac{dx}{x(1 - \frac{x}{K})} = rdt$$

$$\int (\frac{1}{x} + \frac{1}{K - x})dx = \int rdt$$

$$\int \frac{dx}{x} + \int \frac{dx}{K - x} = rt + c$$

$$ln|x| - ln|K - x| = rt + c$$

$$ln|K - x| - ln|x| = -rt - c$$

$$ln\left|\frac{K - x}{x}\right| = -rt - c$$

$$\left|\frac{K - x}{x}\right| = e^{-rt-c}$$

$$\frac{K - x}{x} = -Ce^{-rt}$$

$$x = \frac{K}{1 + Ce^{-rt}}$$

3

$$\frac{dx}{dt} = sin(x)$$

$$csc(x)dx = dt$$

$$\int csc(x)dx = \int dt$$

$$ln\left|tan(\frac{x}{2})\right| = t + C$$

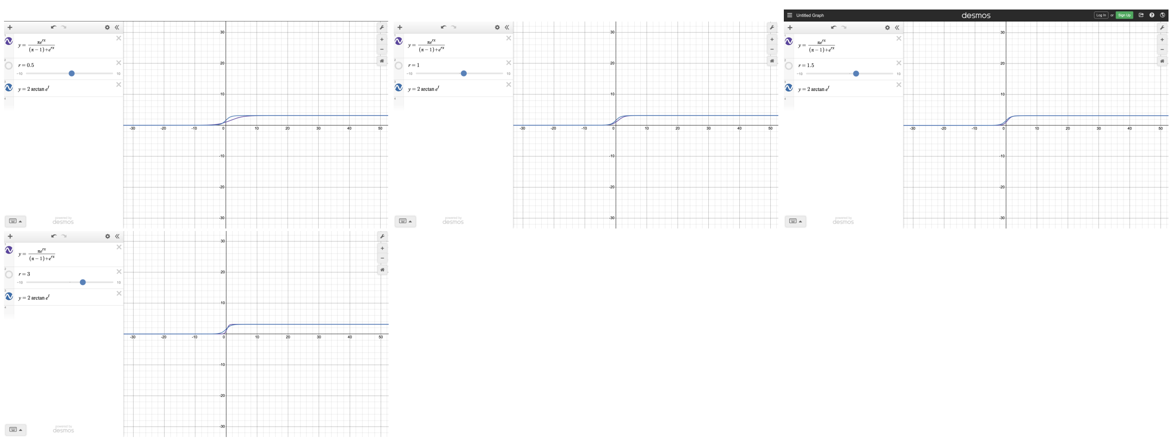
$$x = 2tan^{-1}Ce^{t}$$

$$\frac{dx}{dt} = rx(1 - \frac{x}{\pi})$$

$$x = \frac{\pi}{1 + Ce^{-rt}}$$

$$x(0) = 1 = \frac{\pi}{1 + Ce^{-rt}}$$

$$x(t) = \frac{\pi e^{rt}}{(\pi - 1) + e^{rt}}$$



4

$$\frac{dx}{dt} = ax + by$$

$$\frac{dy}{dt} = cx + dy$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Source:

$$ad - bc > 0, a + d > 0$$

$$a = 1, b = 0, c = 0, d = 1$$

Sink:

$$ad - bc > 0, a + d < 0$$

$$a = -2, b = -2, c = 2, d = 1$$

Spiral Source:

$$ad - bc > 0, a + d > 0, (a + d)^2 - 4(ad - bc) < 0$$

$$a = 1, b = -2, c = 2, d = 0$$

Spiral Sink:

$$ad - bc > 0, a + d < 0, (a + d)^2 - 4(ad - bc) < 0$$

$$a = -2, b = -2, c = 2, d = 1$$

Saddle:

$$ad - bc < 0$$

$$a = -1, b = 1, c = 1, d = 1$$

5

5.1

Region	Birth Rate	Death Rate	Migration Rate
California	0.1315060	0.0472744	0.0865414
Rest of U.S.	0.1282137	0.0487687	-0.0073908

5.2

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} B = \begin{bmatrix} 0.1315060 & 0 \\ 0 & 0.1282137 \end{bmatrix} D = \begin{bmatrix} 0.0472744 & 0 \\ 0 & 0.0487687 \end{bmatrix} M = \begin{bmatrix} 0.0865414 & 0 \\ 0 & -0.0073908 \end{bmatrix}$$

$$I + B - D + M = \begin{bmatrix} 1.170773 & 0 \\ 0 & 1.0720542 \end{bmatrix} = G$$

$$\begin{bmatrix} 1.170773 & 0 \\ 0 & 1.0720542 \end{bmatrix} \begin{bmatrix} 12988 \\ 152082 \end{bmatrix} = \begin{bmatrix} 15206 \\ 163040 \end{bmatrix}$$

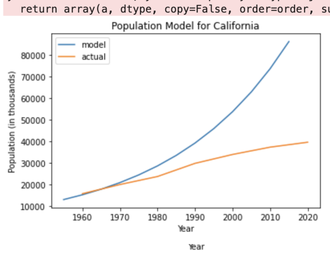
The identity matrix I is needed because the birth, death, and migration rates represent an increase or decrease in amount of the people relative to the population, so we must take those changes and add them to the initial population to get the new population. The matrix G multiplied by the matrix of the population of 1955 correctly yields the population of 1960.

5.3

```
[102]: actual_pop_cal = (1 / 1000) * np.array([15717204, 19953134, 23667902, 29760021, 33871648, 37253956, 39538223], dtype = float)
       years_by_10 = np.array([1960, 1970, 1980, 1990, 2000, 2010, 2020], dtype = float)
```

```
[103]: plt.plot(years, pop_model_cal)
       plt.plot(years_by_10, actual_pop_cal)
       plt.title("Population Model for California")
       plt.xlabel("Year")
       plt.ylabel("Population (in thousands)")
       plt.legend(labels = ["model", "actual"])
       plt.show()
```

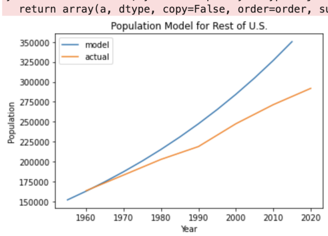
```
/opt/anaconda3/lib/python3.9/site-packages/numpy/core/_asarray.py:171: VisibleDeprecationWarning: Creating an ndarray from ragg
d nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If
you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return array(a, dtype, copy=False, order=order, subok=True)
```




```
[90]: pop_rest = (1 / 1000) * np.array([179323175, 203211926, 226545805, 248709873, 281421906, 308745538, 331449281]) - actual_pop_cal
```

```
•[91]: plt.plot(years, pop_model_rest)
       plt.plot(years_by_10, actual_pop_rest)
       plt.title("Population Model for Rest of U.S.")
       plt.xlabel("Year")
       plt.ylabel("Population")
       plt.legend(labels = ["model", "actual"])
       plt.show()
```

```
/opt/anaconda3/lib/python3.9/site-packages/numpy/core/_asarray.py:171: VisibleDeprecationWarning: Creating an ndarray from ragge
d nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If
you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return array(a, dtype, copy=False, order=order, subok=True)
```



The model for the rest of the US population is more accurate than the California one, but both seem to diverge as net population growth does not stay constant in the real world.

5.4

```
[76]: np.linalg.eig(G)
```

```
[76]: (array([1.170773 , 1.0720542]),
       array([[1., 0.],
              [0., 1.]]))
```

```
[77]: np.linalg.eig(G)[1][:,0]
```

```
[77]: array([1., 0.])
```

The Californian population is growing at 17% per 5 years and the rest of the U.S. population is growing at 7% each year. The end behavior as time goes on is the proportion of California population to the rest of the U.S. population being

1:0. In other words, California would have 100% of the population and the rest 0%. This is because California is growing at a higher exponential rate than the rest of the country, and as time reaches infinity the Californian population will grow exponentially faster.

5.5

```
[138]: T = np.array([[12174, 1938], [814, 150144]])
```

```
[139]: T = (1 / T.sum(axis=0)) * T
```

```
[142]: print(T)

[[0.93732676 0.01274313]
 [0.06267324 0.98725687]]
```

```
[147]: (1/ np.sum(np.linalg.eig(T)[1][:,1])) * np.linalg.eig(T)[1][:,1]

[147]: array([0.1689703, 0.8310297])
```

This proportion is very different, as the chance of being in California as time goes on because 0.17 while being from the rest of the US is 0.83. This is due to the fact that we are not using a constant growth rate where as time goes infinity one population grows exponentially more, but rather using using a Markov chain to model real probabilities of growth and decline.

5.6

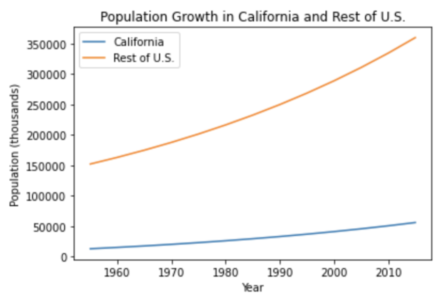
```
[161]: G @ np.array([[12988], [152082]])

[161]: array([[ 15206.00067634],
              [163040.15383446]])
```

The model population in 1960 is 15,206,000 Californians and 163,040,000 Americans excluding Californians.

```
[148]: G = np.array([[0.1315060-0.0472744+0.93732676, 0.01274313], [0.06267324, 0.1282137-0.0487687+0.98725687]])
```

```
[149]: pop_model_cal_2 = [np.array((12988), dtype=float)]
       pop_model_rest_2 = [np.array((152082), dtype=float)]
       initial = np.array([[12988], [152082]])
       for i in range(12):
           pop_model_cal_2.append(population_model(initial)[0])
           pop_model_rest_2.append(population_model(initial)[1])
           initial = population_model(initial)
```

```
[158]: plt.plot(years, pop_model_cal_2)
       plt.plot(years, pop_model_rest_2)
       plt.xlabel("Year")
       plt.ylabel("Population (thousands)")
       plt.title("Population Growth in California and Rest of U.S.")
       plt.legend(labels=["California", "Rest of U.S."])
       plt.show()
```




```
[159]: np.linalg.eig(G)
```

```
[159]: (array([1.00796193, 1.0802983 ]),
        array([[-0.68384059, -0.21200988],
               [ 0.72963145, -0.97726752]]))
```

```
[160]: (1 / np.sum(np.linalg.eig(G)[1][:,1])) * np.linalg.eig(G)[1][:,1]
```

```
[160]: array([0.17826781, 0.82173219])
```

The long range proportions are very similar to the proportions found in part 5, with around 17% in California and 83% outside of California.