Modelling the Economy and Immigration with a Discrete Dynamical System

Eugene Quan

Introduction

In present-day society, immigration control and undocumented workers have constantly been the center of social discussion. Debates have ranged from how undocumented workers help the economy to how they disrupt the social hierarchy of the country. We attempt to shed some light on the issue by using some mathematics to model how undocumented workers affect the economy. More specifically, we represent the amount of money the government has, the amount of money it spends on deporting undocumented workers, the country's number of legal residents, and its number of undocumented workers using a discrete dynamical system, and look at its behavior over time.

Assumptions, Definitions, and Derivation

We begin with some definitions:
• G_n denotes the amount of money the government has at the end of the nth year.
• X_n denotes the amount of money the government will spend on deporting undocumented workers during the $n+1$st year. All that money will disappear by the end of the nth year.
• L_n denotes the country's number of legal residents at the end of the nth year.
• U_n denotes the country's number of undocumented workers at the end of the nth year.

These are the four variables we will be examining for our model. We now list our assumptions and derive the equations for the aforementioned variables.

Equations for G_n and X_n

• The only money the government spends every year is for deporting undocumented workers. Hence, the government only spends X_{n-1} during the nth year.
• Every year, every undocumented worker, except for those who arrived into the country or will die in that year, contributes m_U dollars to the government. The number of undocumented workers who will die during the nth year is $d_U U_{n-1}$, where d_U is a constant between 0 and 1. As a result, the undocumented workers contribute $m_U(1-d_U)U_{n-1}$ dollars during the nth year.
• Every year, every legal resident, except for those who were born, those who were legalized, those who arrived in the country, or those who will die in that year, contributes m_L dollars to the government. If $d_L L_{n-1}$ legal residents will die during the nth year, where d_L is also a constant between 0 and 1, then the legal residents contribute $m_L(1-d_L)L_{n-1}$ during the nth year.
• During the nth year, the government plans to spend k dollars per undocumented worker in the country at the start of that year; hence it wishes to spend kU_{n-1} dollars. However, if

spending this amount results in the government's money becoming negative at the end of the nth year, then this amount is adjusted so that the government's money is 0 at the end of the year.

In other words, let $G_{desired} = G_{n-1} + m_L(1 - d_L)L_{n-1} + m_U(1 - d_U)U_{n-1} - kU_{n-1}$ denote the amount of money the government plans to spend. If $G_{desired} < 0$, then

$$\begin{aligned} X_{n-1} &= m_L(1 - d_L)L_{n-1} + m_U(1 - d_U)U_{n-1}, \\ G_n &= G_{n-1} + m_L(1 - d_L)L_{n-1} + m_U(1 - d_U)U_{n-1} - X_{n-1} \\ &= 0. \end{aligned}$$

Otherwise,

$$\begin{aligned} X_{n-1} &= kU_{n-1}, \\ G_n &= G_{desired}. \end{aligned}$$

Equation for L_n

• Any legal resident who gives birth to a child cannot die or leave the country in that same year. Any undocumented worker who gives birth to a child cannot die or become legalized in that same year.
• Any child of an undocumented worker is a legal resident.
• b_L and b_U represent the birth proportions for legal residents and undocumented workers respectively every year; hence, there are $b_L L_{n-1} + b_U U_{n-1}$ people born as legal residents during the nth year.
• The proportion of undocumented workers that become legal residents every year is g_U, so $g_U Un - 1$ undocumented workers become legal residents during the nth year.
• The number of people who enter the country as legal residents is a constant, n_L.
• The number of legal residents who will leave the country is a proportion, l_L. No one who will leave the country can have a child or die in that same year. As a result, $l_L L_{n-1}$ legal residents leave during the nth year.
Put $A = L_{n-1} + (b_L - d_L - l_L)L_{n-1} + n_L + (g_U + b_U)U_{n-1}$. Then:

$$L_n = \begin{cases} A, & \text{if } A > 0; \\ 0, & \text{otherwise.} \end{cases}$$

Equation for U_n

• The number of people who enter the country as an undocumented worker is a constant, n_U.
• There exists an efficacy constant, e, which denotes the number of undocumented workers removed from the country per dollar spent. Hence, the number of undocumented workers removed in the nth year is eX_{n-1}.
Set $B = -(g_U + d_U)U_{n-1} + n_U - eX_{n-1}$. It follows that

$$U_n = \begin{cases} B, & \text{of } B > 0; \\ 0, & \text{otherwise.} \end{cases}$$

How Much Should the Government Spend?

Now that the model has been derived, the natural question is how much the government should spend for deporting each undocumented worker, or in other words, how the government should pick k. We now examine the behavior of the system for different values of k, and see if there is an optimal value.

Appendix A contains the constants and initial conditions we use for the model. We use Matlab .m files for all the functions in this project; we first run **quanprojectmath164_part1(k)** which takes in a value for k, and plots the G_n, X_n, L_n, and U_n versus n from 0 to 70. Plots are shown in Appendix B for various values of k. By the plots, when $k = 0, 100$, or 1000, it makes sense that U_n increases initially because the money spent is not sufficient to reduce the overall number of undocumented workers. When $k = 5000$, a different trend occurs; the number of undocumented workers decreases initially, since the amount of money the government is spending is now enough. Also, notice that U_n seems to approach a steady number as time increases; that number decreases as k increases.

What is interesting, however, is that when k increases, U_n tends to have more oscillations before approaching a steady-state value. In fact, Figure 1 shows that for $k = 19500$, the oscillations are huge, and the steady-state is reached for a value of n much larger than 70. A logical question to ask would be if there existed a large enough value for k where oscillations would continue on forever.

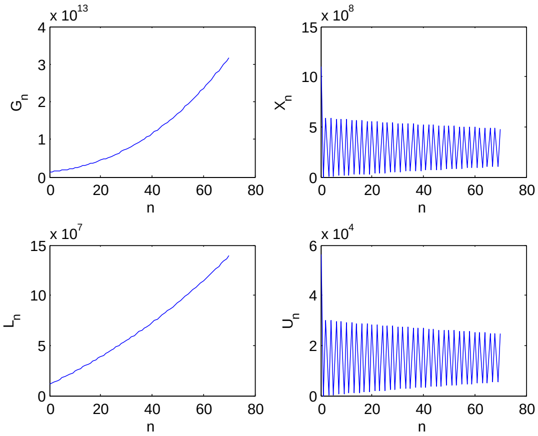

Figure 1: Model Behavior when $k = 19500$

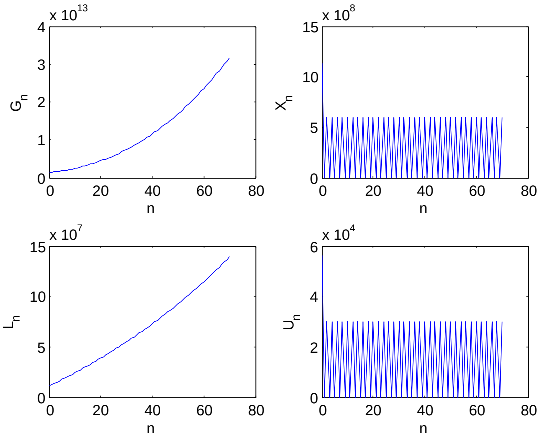

Figure 2: Model Behavior when $k = 20000$

The answer is yes; when $k = 20000$, oscillations in U_n continue forever, as demonstrated by Figure 2. (This is confirmed by Figure 20 in Appendix B which contains a plot of U_n versus n from 0 to 1000). An intuitive explanation for this infinite oscillatory behavior is because $k = 20000$ is enough to immediately remove all undocumented workers; however, when there are no undocumented workers, the government spends no money, and as a result, the number of undocumented workers rises again. This pattern continues forever. Notice also that increasing k past 20000 doesn't decrease the average value of U_n; U_n still oscillates between the same values, as demonstrated by Figure 21, which looks at the system's behavior for $k = 40000$. By this example, the government should never spend more money than the minimal value of k that causes the infinite oscillation of U_n.

Let $n*$ be the last value of n for the plots; here $n* = 70$. We proceed to plot G_{n*} and U_{n*} for different values of k, and see if there an optimal value of k can be obtained this way. Using **plot_K_G_U**(**lowerlimit**, **upperlimit**), where **lowerlimit** and **upperlimit** dictate the respective lower and upper ranges for k to plot G_{n*} and U_{n*}, Figure 3 shows that increasing k (for $0 \leq k \leq 10000$) decreases the final G_n and U_n with diminishing returns:

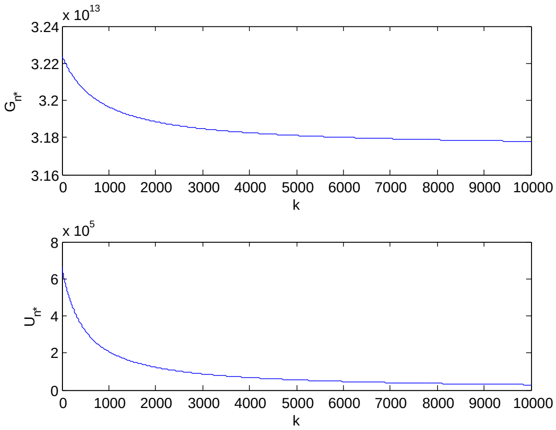

Figure 3: G_{n*} and U_{n*} vs n, $0 \leq n \leq 10000$

We also run the same program for $10000 \leq k \leq 25000$. Since the number of undocumented workers, however, oscillates between 0 and some fixed constant for the larger values of k on this range, we create plots for G_{n*} and U_{n*}, and for G_{n*-1} and U_{n*-1}.

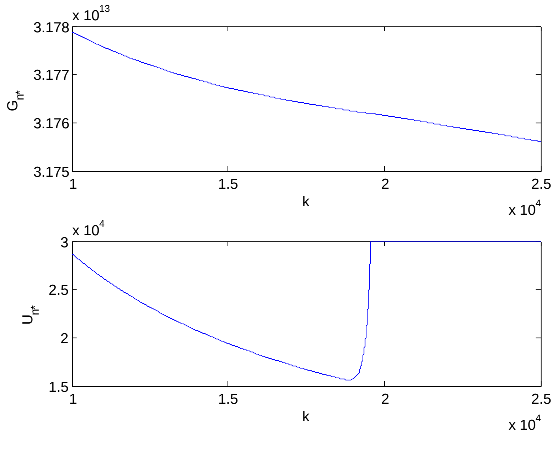

Figure 4: G_{n*}, U_{n*} for $10000 \leq k \leq 25000$

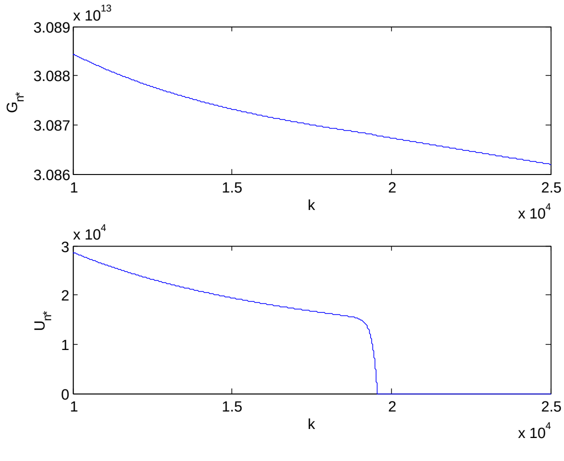

Figure 5: G_{n*-1}, U_{n*-1} for $10000 \leq k \leq 25000$

The plots indicate that increasing k decreases the final number of undocumented workers, but hurts the economy as well. There is no clear optimal value of k that maximizes G_{n*} and minimizes U_{n*}. The only thing that is obvious is that the government should never pick a value greater than the minimum k that results in the infinite oscillation, for the number of undocumented workers oscillate between 0 and the same constant for all those values. As a result, the value of k that should be chosen depends on whether the government wants the number of undocumented workers to quickly approach a higher steady state, or for it to oscillate for a long time, then approach a steady state, even though the steady state of the latter is lower. It also depends on whether the government is willing to spend the money for this tradeoff.

Expanding the Model

Perhaps no clear value for k could be determined because of some simplifying assumptions of the model. First of all, the number of legal residents that entered the country was constant, and the general behavior of L_n remained the same for all values of k. It would make more sense if the number of legal residents that entered was somehow proportional to the economy; in fact, if the economy was poor, legal residents could even leave the country as a result. We now expand and revise the L_n equation to reflect this. Let

$$c_n = \begin{cases} \sqrt{G_n/G_0}, & \text{if } G_n \geq G_0; \\ -\sqrt{G_0/G_n}, & \text{if } 1 * 10^{-2} < G_n < G_0; \\ -\sqrt{G_0/10^{-2}}, & \text{otherwise} \end{cases}$$

denote the proportion of people that enter or exit as legal residents during the nth year based on the economy at the start of the nth year compared to the initial economy. Note that the 10^{-2} is the smallest non-zero value for G_n, and has units of dollars. It merely serves as a lower bound to avoid division by 0 if $G_0 = 0$. Also assume that no legal resident who leaves can have a child or die in that same year. Put $C = L_{n-1} + (b_L - d_L - l_L)L_{n-1} + c_{n-1}n_L + (g_U + b_U)U_{n-1}$. Then

$$L_n = \begin{cases} C, & \text{if } C > 0; \\ 0, & \text{otherwise.} \end{cases}$$

Another simplifying assumption was the number of undocumented workers was linear with the amount of money spent. However, as a rule of economics, the amount spent should follow the law of diminishing returns; every extra dollar spent should really decrease U_n by a smaller amount. Let v be a constant between 0 and .35 inclusive, and define

$$e_n = e/(X_n)^v$$

to be the new efficacy function that determines the number of undocumented workers removed per dollar spent; therefore, every extra dollar spent reduces the efficacy. e, which will take on the same value as in the previous section, will now denote the number of undocumented workers removed for every $1/(1-v)$th root of a dollar spent. Put $D = -(g_U + d_U)U_{n-1} + n_U - e_n X_{n-1}$; the U_n equation is now

$$U_n = \begin{cases} D, & \text{if } D > 0; \\ 0, & \text{otherwise.} \end{cases}$$

We now proceed to re-examine the behavior of the system for different values of k, and for values of v as well.

Spending Money as a Function of Efficacy

We begin by using a function **quanprojectmath164_part2**($\mathbf{k}, \mathbf{v}$) that plots the behavior of the system by taking values for k and v. After trying some sample values, we find that many values of k that caused oscillations for $v = 0$ barely cause any oscillations for higher values of v. For example, when $v = .05$, $k = 20000$ barely even creates an oscillation, as shown in Figure 25 in Appendix B, even though it led to an infinite oscillation when v was 0. By the time $v = .15$, $k = 20000$ isn't even sufficient to cause a decrease in the number of undocumented workers.

After some more experimentation, for $v < .3$, the system behaves exactly the same way as under the old model, except that much higher values of k are needed to cause oscillations for U_n. (More examples plots for $v = .05$ are shown in Appendix B). We use a function **plot_k_v**() which, for $v = 0, .05, .10, .15, .20, .25$ and $.3$, determines the minimum k such that the infinite oscillation is created. The function does this beginning with a number less than the minimum k, which was found by experimentation, and increments k until both U_{n*} and U_{n*-2} or U_{n*-1} and U_{n*-3} equal 0. Running **plot_k_v**() creates Figure 6.

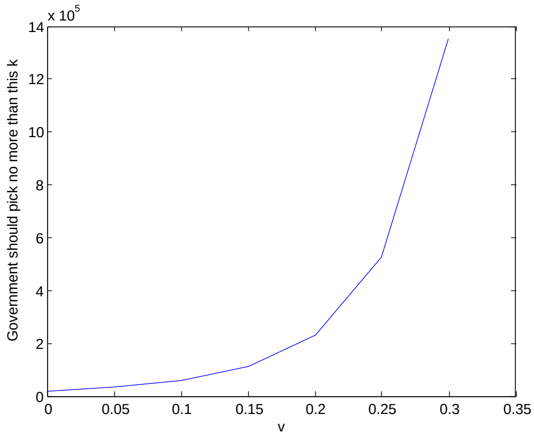

Figure 6: Minimum k versus v

It is not difficult to see that as v increases, the minimum k necessary to force an infinite oscillation increases somewhat exponentially; hence poor efficacy costs the government vast amounts very quickly. Now, the insightful reader may wonder why we didn't write the function to include $v = .35$; it is because that there is no value of k that will create an infinite oscillation, as we demonstrate by running **quanprojectmath164_part2**($\mathbf{k}, \mathbf{v}$) with $k = 2381770$ and $k = 2381870$, both for $v = .35$. The former produces Figure 7, while the latter reveals Figure 8.

What is shocking about Figures 7 and 8 is that $k = 2381770$ produces only slight oscillations, then a steady state, while increasing k by only 100 leads to a sudden collapse in the economy. In addition, the collapse forces all the legal residents to leave the country, and drastically increases the number of undocumented workers because there is no money avaiable to deport them.

As a result, large values of v imply that the only option the government can exercise is to induce initial oscillations for the number of undocumented workers before that number quickly

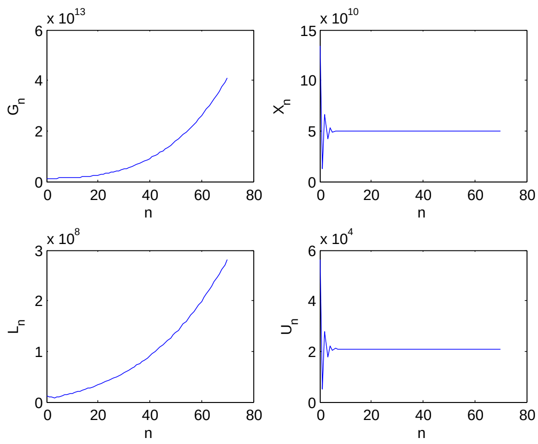

Figure 7: Model Behavior when $k = 2381770$ and $v = .35$

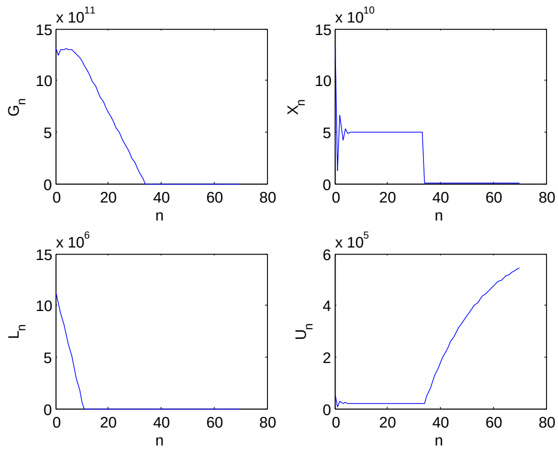

Figure 8: Model Behavior when $k = 2381870$ and $v = .35$

settles into a steady state if it wishes to preserve its economy. This, however, is not necesarily a negative thing if the government wishes to quickly settle U_n into a steady state anyways; this merely means that the number of options for the government is severly reduced, and implies that a slight mistake can lead to its economical collapse.

One Last Revision

One could argue that the last plot, however, is unrealistic with respect to the behavior of U_n. The last model states that legal residents leave when G_n is less than G_0; the same argument could be made for undocumented workers. Using the same definition of c_n as in the previous section, and letting e_U represent a constant proportion of the undocumented workers that leave as a result of poor economy, we define

$$E = \begin{cases} -(g_U + d_U)U_{n-1} + n_U - e_n X_{n-1}, & \text{if } c_n > 0; \\ -(g_U + d_U)U_{n-1} + n_U - e_n X_{n-1} + c_n e_U, & \text{otherwise.} \end{cases}$$

We can now revise U_n to be

$$U_n = \begin{cases} E, & \text{if } E > 0; \\ 0, & \text{otherwise.} \end{cases}$$

Again, we assume that no undocumented workers who leave the country will die, have a child, or legalize in that same year.

Is there a Change in the Behavior?

We begin by seeing if there is a change in the behavior for $k = 2381870$ when $v = .35$. We use **quanprojectmath164_part3(k, v)** to create Figure 9.

Figure 9 shows that the economy no longer collapses. An intuitive reason is that the economy first declines due to a huge spending; this drop is sufficient to force many undocumented workers

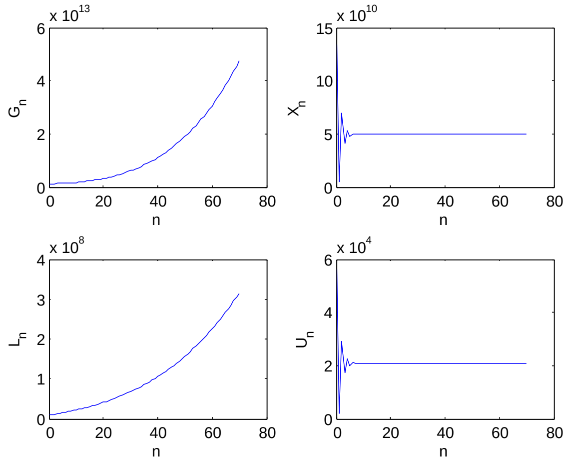

Figure 9: Model behavior when $k = 2381870, v = .35$

to leave. As a result, the amount spent on fighting the remaining undocumented workers is much smaller and not large enough to hurt the economy, leading to a recovery instead. The insightful reader will ask if this is always the case, no matter what the value of k is. The answer is no, as demonstrated with $k = 2664170$ and $k = 2664270$ when v is again .35. When k= 2664170, oscillations occur with U_n before it levels off as shown in Figure 10, like in the previous models. However, in Figure 11, when $k = 2664270$, the same type of collapse occurs, except this time, U_n decreases to 0 as well.

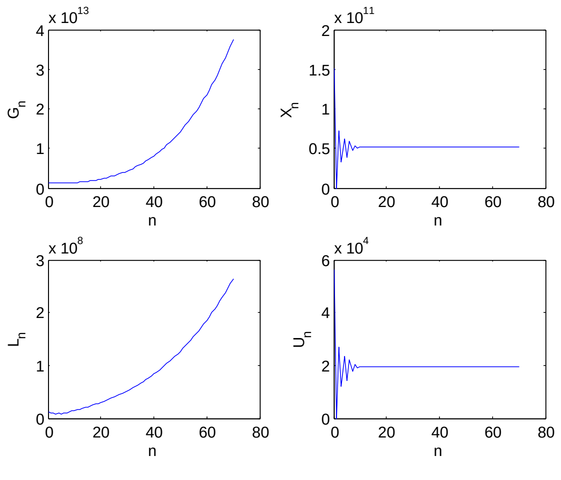

Figure 10: Model behavior when $k = 2664170, v = .35$

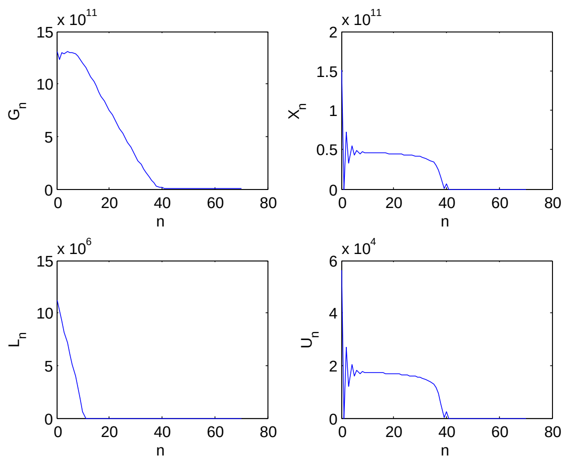

Figure 11: Model behavior when $k = 2664270, v = .35$

These plots, therefore, indicate that this model exhibits the same type of behavior as the previous one, except that a higher value of k is required to force an economic collapse. In addition, when the economy collapses, both the legal residents and undocumented workers leave, leaving the country to be an unpopulated piece of land. Therefore, if the government isn't careful with how it spends it money at large levels, it can very easily destroy the very entity of its own country.

Conclusion and Implications

Even after revising the model twice, there is no clear answer for how much the government should spend to deport undocumented workers. The answer mostly depends on whether the government wants to have a lower average value of undocumented workers, even though that takes more money, time, and oscillation in the undocumented worker population to accomplish, or a higher constant number of undocumented workers that is achieved more quickly and smoothly. What the models do tell is that when efficacy is high (for low values of v), the government has the option of choosing to fulfill either of the aforementioned goals. However, when efficacy decreases, the amount of money the government must spend in deporting undocumented workers increases drastically, and for very poor levels of efficacy, the government can only settle for the latter goal if it wishes to preserve its economy. Therefore, even though the models don't provide one grand solution that applies for every case, they do give a very clear indication of what the government should never spend and provide some informative guidance depending on its goals.

References

'Birth Rates Among Immigrants in America: Comparing Fertility in the U.S. and Home Countries.' Center for Immigration Studies.
<http://www.cis.org/articles/2005/back1105.html>

'Estimates of the Unauthorized Immigrant Population Residing in the United States: 1990 to 2000.' Office of Policy and Planning U.S. Immigration and Naturalization Service.
<http://www.uscis.gov/graphics/shared/aboutus/statistics/Ill_Report_1211.pdf>

Historical Tables, Budget of the United States Government, Fiscal Year 1999.
<www.fas.org/man/docs/fy99/historical/019-020.htm>

'Illegal immigration invasion numbers.' Desert Invasion - U.S.
<http://www.desertinvasion.us/data/invasion_numbers.html>

'Population by Sex, Age, and Citizenship Status: March 2000'. United States Census.
<http://www.census.gov/population/socdemo/foreign/p20-534/tab0101.pdf>

Appendix A - Constants and Initial Conditions

• $G_0 = 1.308608 * 10^{12}$ dollars
• $L_0 = 1.1271743 * 10^7$ legal residents
• $U_0 = 5.6488 * 10^4$ undocumented workers
• $m_L = 6815$ dollars contributed per legal resident
• $m_U = 1976$ dollars contributed per undocumented worker
• $b_L = .0286$ - birth proportion for legal residents
• $b_U = .07$ - birth proportion for undocumented workers
• $d_L = .03938$ - death proportion for legal residents
• $d_U = .02$ - death proportion for undocumented workers
• $g_U = .0235$ - legalization proportion for undocumented workers
• $n_L = 1.12 * 10^6$ - number of legal residents that enter per year
• $n_U = 3.00 * 10^4$ - number of undocumented workers that enter per year
• $e = 1 * 10^{-4}$ - number of undocumented workers deported per dollar spend
• $e_U = 3 * 10^3$ - scaling number of undocumented workers who leave when the economy is poor (final model only)

Appendix B - Figures

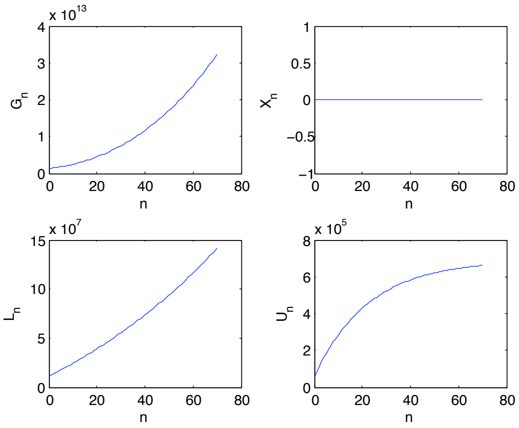

Figure 12: Model 1, $k = 0$

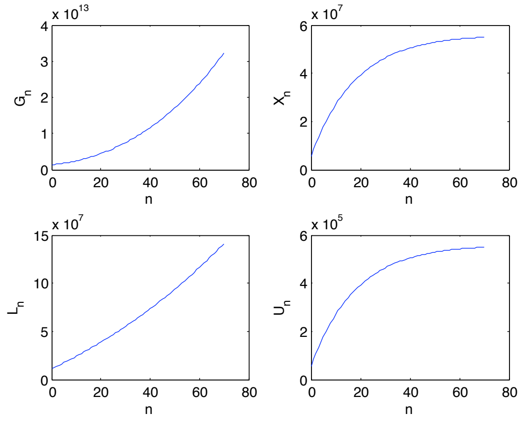

Figure 13: Model 1, $k = 100$

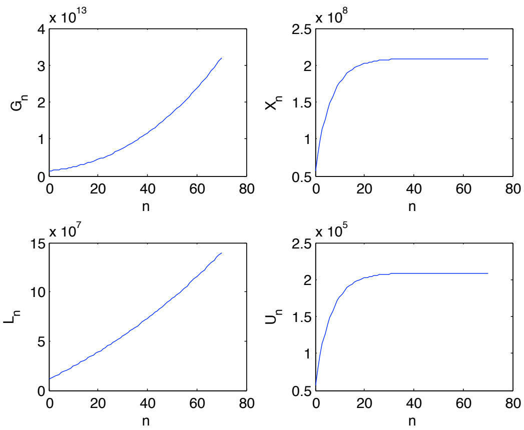

Figure 14: Model 1, $k = 1000$

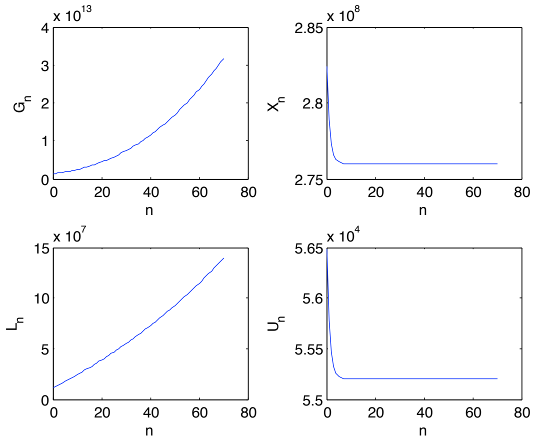

Figure 15: Model 1, $k = 5000$

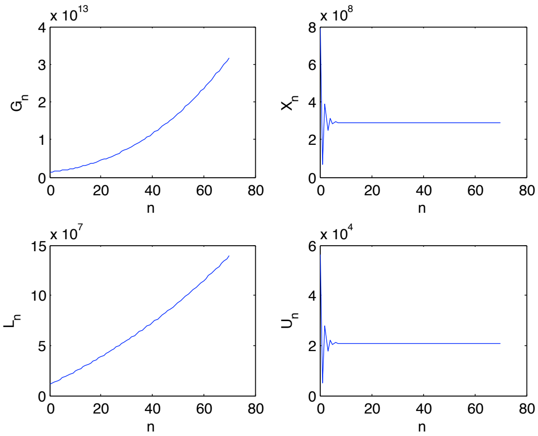

Figure 16: Model 1, $k = 10000$

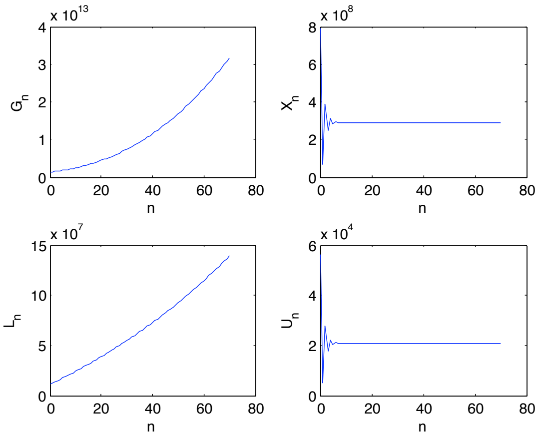

Figure 17: Model 1, $k = 14000$

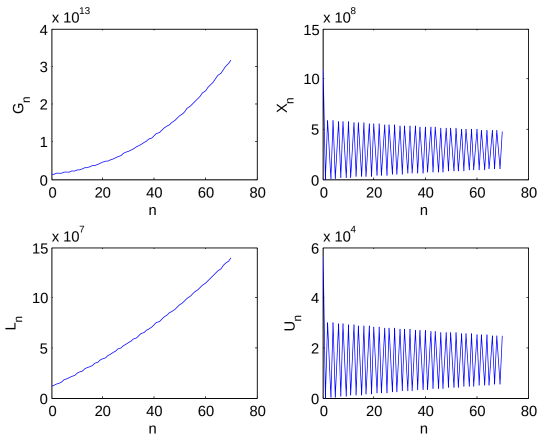

Figure 18: Model 1, $k = 19500$

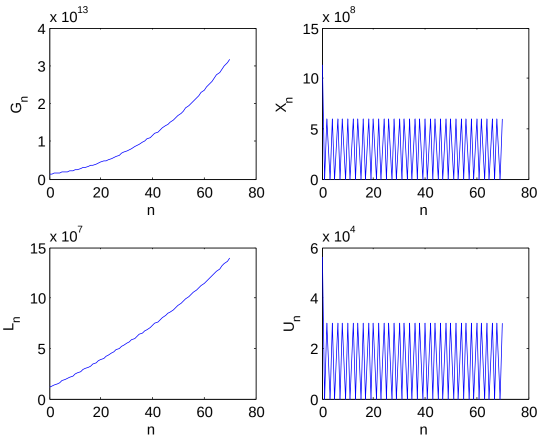

Figure 19: Model 1, $k = 20000$

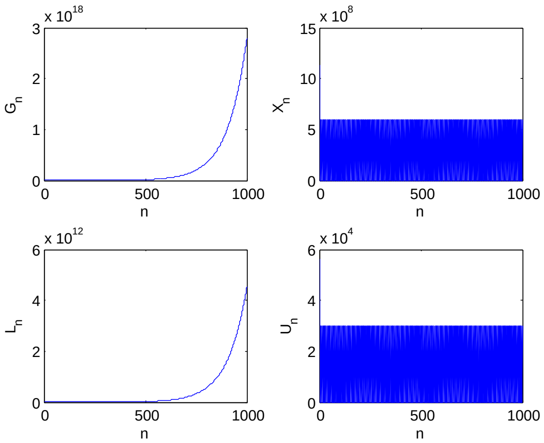

Figure 20: Model 1, $k = 20000$, $0 \leq n \leq 100$

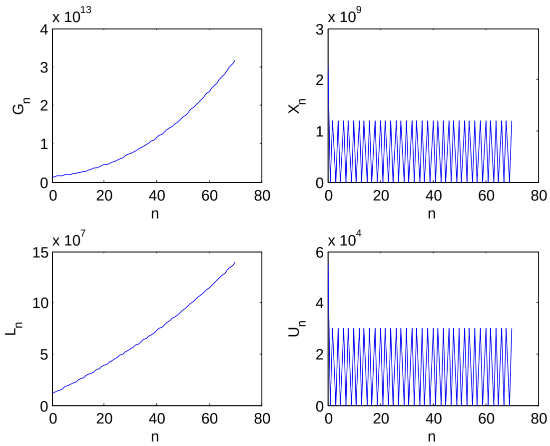

Figure 21: Model 1, $k = 40000$

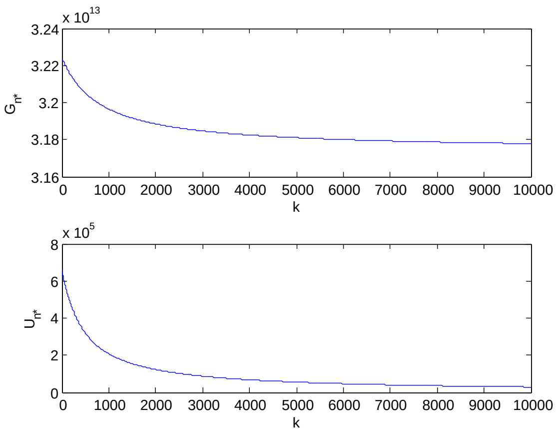

Figure 22: G_{n*}, U_{n*} for $0 \leq k \leq 10000$

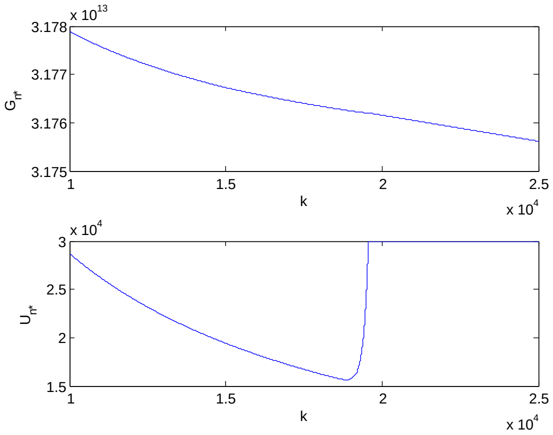

Figure 23: G_{n*}, U_{n*} for $10000 \leq k \leq 25000$

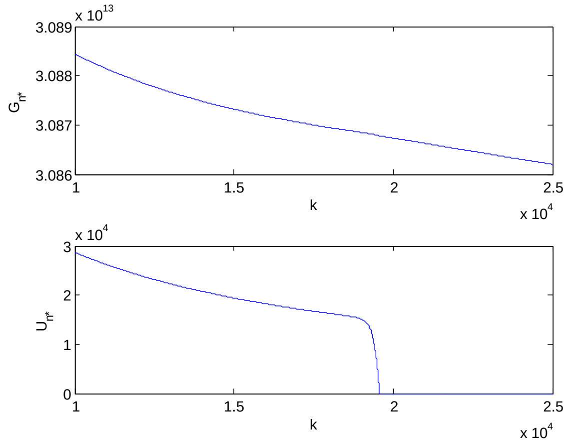

Figure 24: G_{n*-1}, U_{n*-1} for $10000 \leq k \leq 25000$

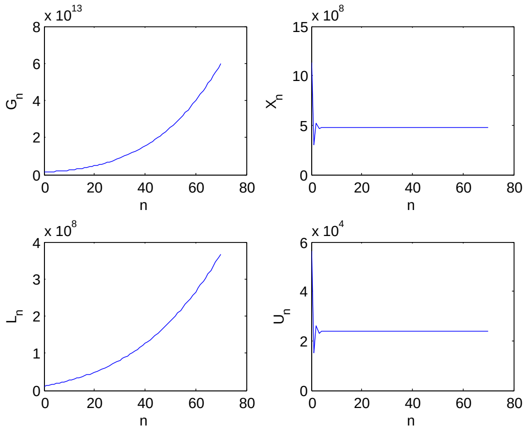

Figure 25: Model 2, $k = 20000, v = .05$

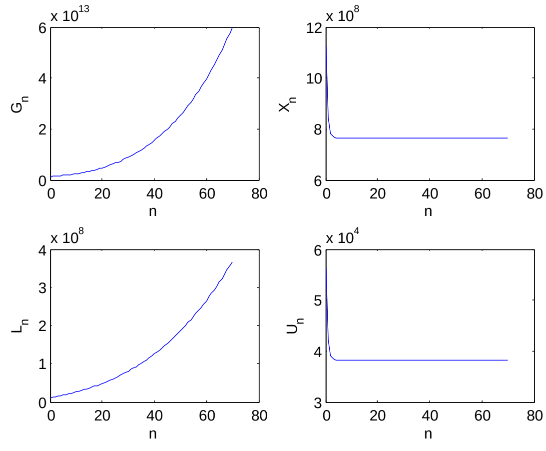

Figure 26: Model 2, $k = 20000, v = .10$

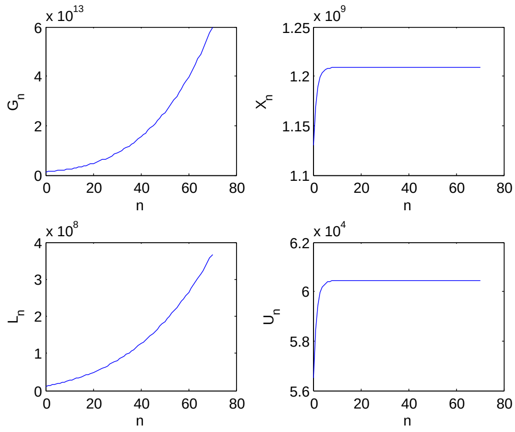

Figure 27: Model 2, $k = 20000, v = .15$

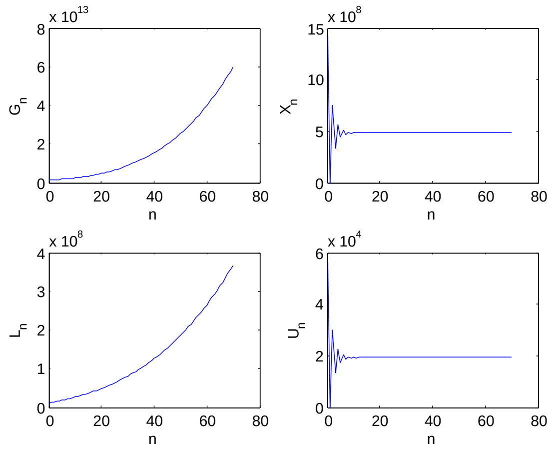

Figure 28: Model 2, $k = 25050, v = .05$

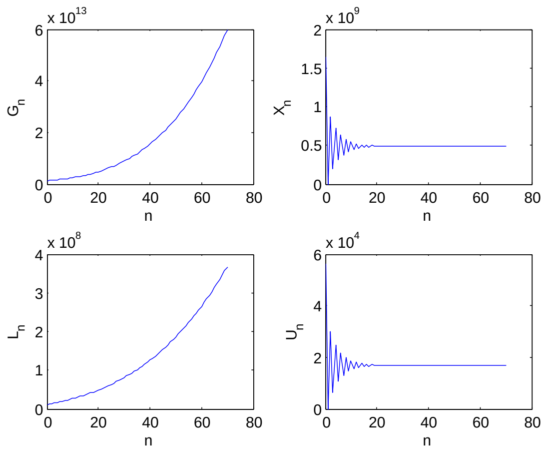

Figure 29: Model 2, $k = 29050, v = .05$

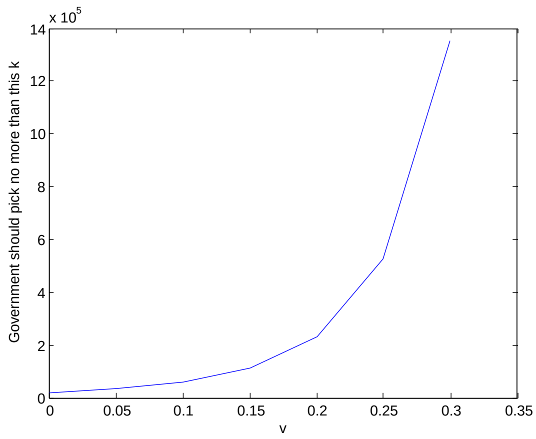

Figure 30: Minimum k required for oscillation of U_n versus v

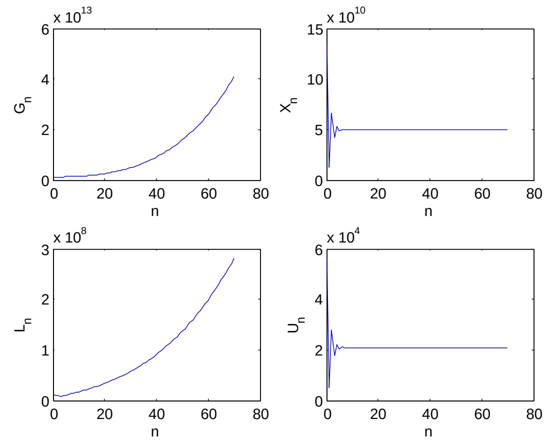

Figure 31: Model 2, $k = 2381770, v = .35$

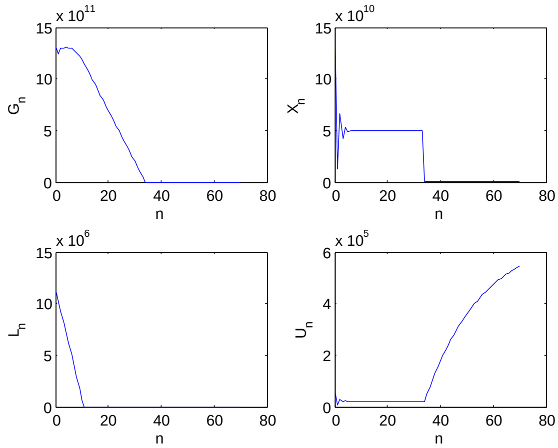

Figure 32: Model 2, $k = 2381870, v = .35$

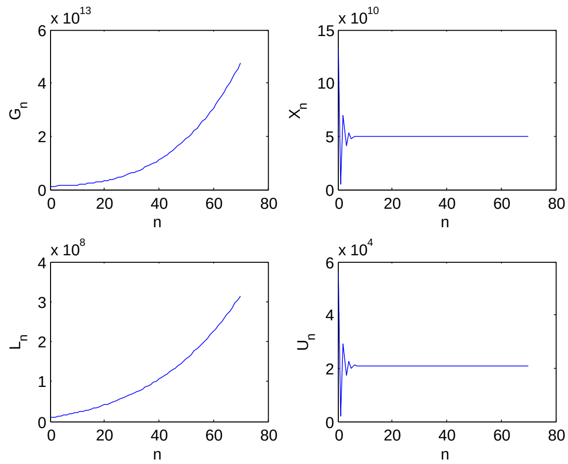

Figure 33: Model 3, $k = 2381870, v = .35$

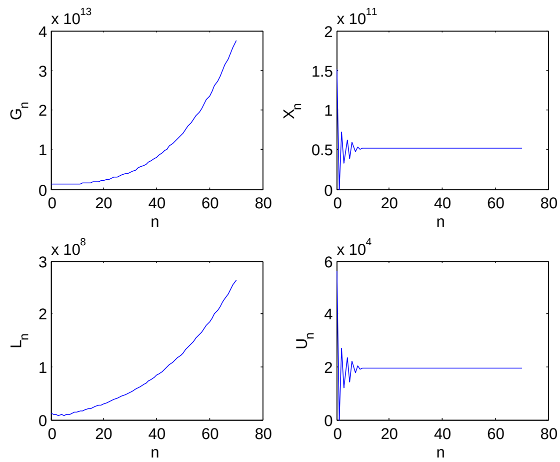

Figure 34: Model 3, $k = 2664170, v = .35$

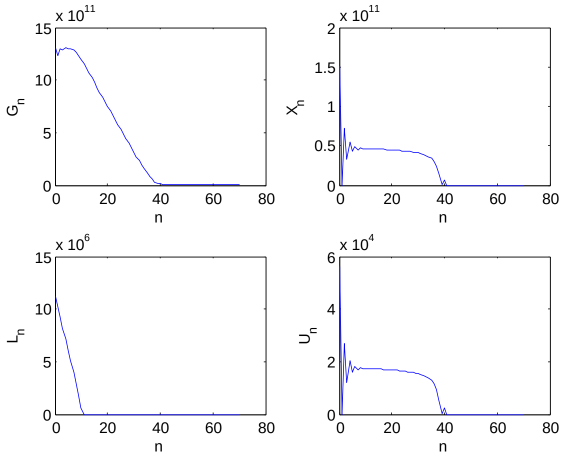

Figure 35: Model 3, $k = 2664270, v = .35$

Appendix C - Matlab Code

```
% quanprojectmath164_part1(k).m
% This takes k as an input; k represents the amount of money
% the government wishes to spend per undocumented worker at the
% start of every year. This is for the original model.

% Written by Eugene Quan

function data = quanprojectmath164_part1(k)

mL = 6815;  % Amount of money each legal resident contributes
mU = 1976;  % Amount of money each undocumented worker contributes
bU = .0286; % Birth proportion for undocumented workers
bL = .07;   % Birth proportion for legal residents
dL = .03938;% Death proportion for legal residents
gU = .0235; % Legalization proportio for undocumented workers
dU = .02;   % Death proportion for undocumented workers
lL = .02;   % Leave proportion for legal residents
nL = 1.12*10^(6); % Number of legal residents who enter per year
nU = 3*10^(4);    % Number of undocumented workers who enter per year
e = 1*10^(-4);    % Efficacy constant

% Initial conditions for amount of money, number of legal residents,
% and number of undocumented workers.
initialG = 1.308608*10^(12); %G_0
initialL = 1.1271743*10^(7); %L_0
initialU = 5.6488*10^(4);    %U_0

G = initialG;
L = initialL;
U = initialU;

% We start at t = 0.
initialT = 0;

% Used to store the values of the variables at the start of every year.
Gvector = [initialG];
Lvector = [initialL];
Uvector = [initialU];
Tvector = [initialT];
Xvector = [];

% We iterate for every year.
for t=(initialT + 1):70
% How much the government plans to spend...
newX = k*U;
% and how that will affect the economy.
newG = G + mL*(1-dL)*L + mU*(1-dU)*U - newX;
```

```
if(newG < 0)    % If planned spending will produce a deficit
    X = G + mL*(1-dL)*L + mU*(1-dU)*U; %Spend maximum amount of money allowed
    G = 0; % Economy is now zero
else
    X = newX; % Otherwise, spend planned amount
    G = newG; % And gets G_n accordingly.
end

% Potential L_n. If it's negative, then force it to 0.
newL = (bL - dL - lL)*L + L + nL +(bU + gU)*U;
if(newL <=0)
    L = 0;
else
    L = newL;
end

% Potential U_n. If it's negative, then force it to 0.
newU = U -(gU + dU)*U + nU  - e*X; %gets I(t)
if(newU <= 0)
    U = 0;
else
    U = newU;
end

% Saves the value of the variables into the vector for this particular
% year.
Gvector = [Gvector G];
Lvector = [Lvector L];
Uvector = [Uvector U];
Xvector = [Xvector X];
Tvector = [Tvector t];
end

% This to get the final value of X, since X is behind by one
% in indexing.
finalX = k*U;
finalG = G + mL*(1-dL)*L + mU*(1-dU)*U - finalX;
if(finalG <= 0)
    X = G + mL*(1-dL)*L + mU*(1-dU)*U
else
    X = finalX;
end
Xvector = [Xvector X];

% Returns the final set of values for every year.
data = [Tvector' Gvector' Xvector' Lvector' Uvector'];

% Plots everything.
```

```
figure(1)
clf
subplot(2,2,1)
plot(Tvector,Gvector)
xlabel('n')
ylabel('G_n')

subplot(2,2,2)
plot(Tvector,Xvector)
xlabel('n')
ylabel('X_n')

subplot(2,2,3)
plot(Tvector,Lvector)
xlabel('n')
ylabel('L_n')

subplot(2,2,4)
plot(Tvector,Uvector)
xlabel('n')
ylabel('U_n')
```

```
% quanprojectmath164_part2(k,v).m
% This takes k  and v as inputs; k represents the amount of money
% the government wishes to spend per undocumented worker at the
% start of every year. v is used in the efficacy function.
% This is for the second model.

% Written by Eugene Quan

function data = quanprojectmath164_part2(k,v)

mL = 6815;  % Amount of money each legal resident contributes
mU = 1976;  % Amount of money each undocumented worker contributes
bU = .0286; % Birth proportion for undocumented workers
bL = .07;   % Birth proportion for legal residents
dL = .03938;% Death proportion for legal residents
gU = .0235; % Legalization proportio for undocumented workers
dU = .02;   % Death proportion for undocumented workers
lL = .02;   % Leave proportion for legal residents
nL = 1.12*10^(6); % Number of legal residents who enter per year
nU = 3*10^(4);    % Number of undocumented workers who enter per year
c = 1*10^(-4);    % Constant used in calculating efficacy.

% Initial conditions for amount of money, number of legal residents,
% and number of undocumented workers.
initialG = 1.308608*10^(12); %G_0
initialL = 1.1271743*10^(7); %L_0
initialU = 5.6488*10^(4);    %U_0

G = initialG;
L = initialL;
U = initialU;

% We start at t = 0.
initialT = 0;

% Used to store the values of the variables at the start of every year.
Gvector = [initialG];
Lvector = [initialL];
Uvector = [initialU];
Tvector = [initialT];
Xvector = [];
% We iterate for every year.
for t=(initialT + 1):70
% How much the government plans to spend...
newX = k*U;
% and how that will affect the economy.
newG = G + mL*(1-dL)*L + mU*(1-dU)*U - newX;
if(newG < 0)    % If planned spending will produce a deficit
    X = G + mL*(1-dL)*L + mU*(1-dU)*U; %Spend maximum amount of money allowed
```

```
    G = 0; % Economy is now zero
else
    X = newX; % Otherwise, spend planned amount
    G = newG; % And gets G_n accordingly.
end

% This proportion is used for the legal residents entering/leaving
% based on the status of the economy.
if(G < initialG)
    if(G < 10^(-2))
        changingProportion = -sqrt(initialG*10^(2));
    else
        changingProportion = -sqrt(initialG/G);
    end
else
    changingProportion = sqrt(G/initialG);
end

% Potential L_n. If it's negative, then force it to 0.
newL = (bL - dL - lL)*L + L + nL*changingProportion +(bU + gU)*U;
if(newL <=0)
    L = 0;
else
    L = newL;
end

% Efficacy function.
e = c*k^(-v);
% Potential U_n. If it's negative, then force it to 0.
newU = U -(gU + dU)*U + nU  - e*X; %gets I(t)
if(newU <= 0)
    U = 0;
else
    U = newU;
end

% Saves the value of the variables into the vector for this particular
% year.
Gvector = [Gvector G];
Lvector = [Lvector L];
Uvector = [Uvector U];
Xvector = [Xvector X];
Tvector = [Tvector t];
end

% This to get the final value of X, since X is behind by one
% in indexing.
finalX = k*U;
finalG = G + mL*(1-dL)*L + mU*(1-dU)*U - finalX;
```

```
if(finalG <= 0)
    X = G + mL*(1-dL)*L + mU*(1-dU)*U
else
    X = finalX;
end
Xvector = [Xvector X];

% Returns the final set of values for every year.
data = [Tvector' Gvector' Xvector' Lvector' Uvector'];

% Plots everything.
figure(1)
clf
subplot(2,2,1)
plot(Tvector,Gvector)
xlabel('n')
ylabel('G_n')

subplot(2,2,2)
plot(Tvector,Xvector)
xlabel('n')
ylabel('X_n')

subplot(2,2,3)
plot(Tvector,Lvector)
xlabel('n')
ylabel('L_n')

subplot(2,2,4)
plot(Tvector,Uvector)
xlabel('n')
ylabel('U_n')
```

```
% quanprojectmath164_part3(k,v).m
% This takes k  and v as inputs; k represents the amount of money
% the government wishes to spend per undocumented worker at the
% start of every year. v is used in the efficacy function.
% This is for the final model.

% Written by Eugene Quan

function data = quanprojectmath164_part1(k,v)

mL = 6815;  % Amount of money each legal resident contributes
mU = 1976;  % Amount of money each undocumented worker contributes
bU = .0286; % Birth proportion for undocumented workers
bL = .07;   % Birth proportion for legal residents
dL = .03938;% Death proportion for legal residents
gU = .0235; % Legalization proportio for undocumented workers
dU = .02;   % Death proportion for undocumented workers
lL = .02;   % Leave proportion for legal residents
nL = 1.12*10^(6); % Number of legal residents who enter per year
nU = 3*10^(4);    % Number of undocumented workers who enter per year
c = 1*10^(-4);    % Constant used in calculating efficacy.
eU = 3*10^(3);    % Number used in proportion of undocumented workers
                  % who leave due to poor economy.

% Initial conditions for amount of money, number of legal residents,
% and number of undocumented workers.
initialG = 1.308608*10^(12); %G_0
initialL = 1.1271743*10^(7); %L_0
initialU = 5.6488*10^(4);    %U_0

G = initialG;
L = initialL;
U = initialU;

% We start at t = 0.
initialT = 0;

% Used to store the values of the variables at the start of every year.
Gvector = [initialG];
Lvector = [initialL];
Uvector = [initialU];
Tvector = [initialT];
Xvector = [];
% We iterate for every year.
for t=(initialT + 1):70
% How much the government plans to spend...
newX = k*U;
% and how that will affect the economy.
newG = G + mL*(1-dL)*L + mU*(1-dU)*U - newX;
```

```
if(newG < 0)     % If planned spending will produce a deficit
    X = G + mL*(1-dL)*L + mU*(1-dU)*U; %Spend maximum amount of money allowed
    G = 0;  % Economy is now zero
else
    X = newX; % Otherwise, spend planned amount
    G = newG; % And gets G_n accordingly.
end

% This proportion is used for the legal residents entering/leaving
% based on the status of the economy.
if(G < initialG)
    if(G < 10^(-2))
        changingProportion = -sqrt(initialG*10^(2));
    else
        changingProportion = -sqrt(initialG/G);
    end
else
    changingProportion = sqrt(G/initialG);
end

% Potential L_n. If it's negative, then force it to 0.
newL = (bL - dL - lL)*L + L + nL*changingProportion +(bU + gU)*U;
if(newL <=0)
    L = 0;
else
    L = newL;
end

% Efficacy function.
e = c*k^(-v);
% Potential U_n.
newU = U -(gU + dU)*U + nU  - e*X;
% If the economy is deproving, then there are some undocumented workers
% who will chose to leave, which edits the potential U_n.
if(changingProportion < 0)
    newU = newU + eU*changingProportion;
end

%If U_n < 0, then force U_n = 0.
if(newU <= 0)
    U = 0;
else
    U = newU;
end

% Saves the value of the variables into the vector for this particular
% year.
Gvector = [Gvector G];
```

```
Lvector = [Lvector L];
Uvector = [Uvector U];
Xvector = [Xvector X];
Tvector = [Tvector t];
end

% This to get the final value of X, since X is behind by one
% in indexing.
finalX = k*U;
finalG = G + mL*(1-dL)*L + mU*(1-dU)*U - finalX;
if(finalG <= 0)
    X = G + mL*(1-dL)*L + mU*(1-dU)*U
else
    X = finalX;
end
Xvector = [Xvector X];

% Returns the final set of values for every year.
data = [Tvector' Gvector' Xvector' Lvector' Uvector'];

% Plots everything.
figure(1)
clf
subplot(2,2,1)
plot(Tvector,Gvector)
xlabel('n')
ylabel('G_n')

subplot(2,2,2)
plot(Tvector,Xvector)
xlabel('n')
ylabel('X_n')

subplot(2,2,3)
plot(Tvector,Lvector)
xlabel('n')
ylabel('L_n')

subplot(2,2,4)
plot(Tvector,Uvector)
xlabel('n')
ylabel('U_n')
```

```
% plot_k_G_U_part1(lowerk, upperk).m
% This takes in a lower and upper limit for k, and for all k in this range,
% it gets G_n* and U_n*. It then plots G_n* and U_n* versus k.
function plot_k_G_U_part1(lowerk, upperk)

% Written by Eugene Quan

% Used to store the data.
kvector = [];
Gvector = [];
Uvector = [];

for k=lowerk:upperk
    data = quanprojectmath164_part1(k); %Gets the data for the system
    [m,n] = size(data);
    finalG = data(m,2);  %Gets G_n* - change m to m-1 to get G_(n*-1)
    finalU = data(m,5);  %Gets U_n* - change m to mel to get U_(n*-1)
    kvector = [kvector k];
    Gvector = [Gvector finalG];
    Uvector = [Uvector finalU];
end

% Makes the plots.
figure(1)
clf
subplot(2,1,1)
plot(kvector,Gvector)
xlabel('k')
ylabel('G_n_*')

subplot(2,1,2)
plot(kvector,Uvector)
xlabel('k')
ylabel('U_n_*')
```

```
% plot_k_v.m
% No inputs. For v = 0, .05, .1, .15, .2, .25, and .3, this finds the
% minimum k that causes the infinite oscillation for U_n, for the
% second model.

% Written by Eugene Quan

% Used to store the data.
kvector = [];
vvector = [];

% j is v*10^(2). Used for precision.
j = 0;
v = j*10^(-2);

% Starting guess for v = 0. Starting guesses are provided to speed up
% runtime.
k = 19500;
% While v<=.3.
while(j <= 30)
k = k + 1;
    data = quanprojectmath164_part2(k,v);
    [m,n] = size(data);
    % If the last and third from last values of U_n, or
    % second from last and fourth from last values of U_n are 0,
    % we store that value of k as the minimum k.
    A = (data(m,5) == 0) && (data(m-2,5) == 0);
    B = (data(m-1, 5) == 0) && (data(m-3,5) == 0);
    if(A) || (B)
        kvector = [kvector k];
        vvector = [vvector v];

        % This increments v by .05. Again, starting guesses are provided
        % to speed up runtime.
       switch j
           case{0}
               k = 32800;
           case{5}
               k = 58600;
           case{10}
               k = 111800;
           case{15}
               k = 231300;
           case{20}
               k = 526500;
           case{25}
               k = 1351000;
       end
          j = j + 5;
```

```
          v = j*10^(-2);

    end
end

% Makes the plot.
figure(1)
clf
plot(vvector,kvector)
xlabel('v')
ylabel('Government should pick no more than this k')
```