**Project 1**

**Tribble Population Modeling**

Name: Mohammed Al-Jaber, Mansour Alameer

Course: AAPM 2360

Date: 6/17/2025

**Introduction**

In this project, we will work with the population dynamics of tribbles by using differential equation models which encompass growth and competition and hunting effects. In other words, using a modeled logistic differential equation with a hunting term, we analyze equilibrium solutions, population stability, and the effects of parameter variations. Numerical simulations and direction field visualizations are employed to evaluate long-term population behavior. Seasonal hunting effects are also considered.

**Model Formulation**

In order to model the population dynamics of tribble, we use the logistic equation:

where y shows the units of hundreds of tribbles, H(y) represents the hunting term and it is equal to and it represents the Klingon’s effectiveness at hunting tribbles and in this problem, p = 1.5 (hundreds per day) and q = 1.25 (hundreds3) . Also, a and b represent the reproduction ability of tribbles where a is constant **a = 0.75 (day-1)** but **b** varies from 0.005 to 0.10 **(hundreds-1day-1)** for different species. Thus, the model can be shown as:

**Analysis & Conclusion**

In the mathematical model of population of tribbles, we have a **1st order non-linear autonomous ODE**. To interpret the elemnts of this differential equation, we can say that the term “ay” shows the natural growth with its unit being hundreds of tribbles per day. In addition, the term “-by2” shows the population limitation because of resource constraints. Finally, the term “-H(y)” shows the population reduction of tribbles due to Klingon hunting with its unit being hundreds of tribbles per day.

If we dive deeper into the mathematical term of reduction due to the hunting, , we can see that:

0

Which means as population gets very small and **y, hunting rate approaches 0** (H(y) = 0) which shows that when tribbles’s population gets very small, Klingons struggle to find and hunt them which is very logical since a low population reduces encounter rates. Also, from the 2nd equation, we see that as population gets very large and **y, hunting rate approaches p = 1.5 which shows that Klingon hunting reaches to its maximum rate when tribbles are abundant**. Note that this actually shows a limit to how many tribbles Klingons can hunt per day regardless of population size.

In the next step, we want to analyze the population dynamic equation deeper. Consider the case when this logistic equation is without hunting and H(y) = 0. So, our new equation transforms into:

If we solve this equation by separation of variables, we conclude that in this

case,

We can see that in this scenario, as t, the population converges to . So, **shows the carrying capacity of the population**.

Also, in this case, it specifically shows the maximum sustainable population in the absence of hunting. In the other case when , we reach to a complex integral due to hunting term which can’t be solved analytically.

If we consider the ODE as y’ = f(y) and set f(y) = 0, we can find the equilibrium points of our differential equation. These quilibrium solutions show steady-state populations. In other words, **equilibrium solution is a constant population size where the rate of change = 0 which means births, deaths and hunting balance out and population neither grows nor gets smaller over time.** To get these points, as mentioned before, we should solve the equation y’ = f(y) = 0 which in our case gives us:

Now, in order to determine the equilibrium points of our ODE and analyze their stability, we will plot f as a function of y for different values of b. So, we will find answers of equation f(y) = 0 using MATLAB.

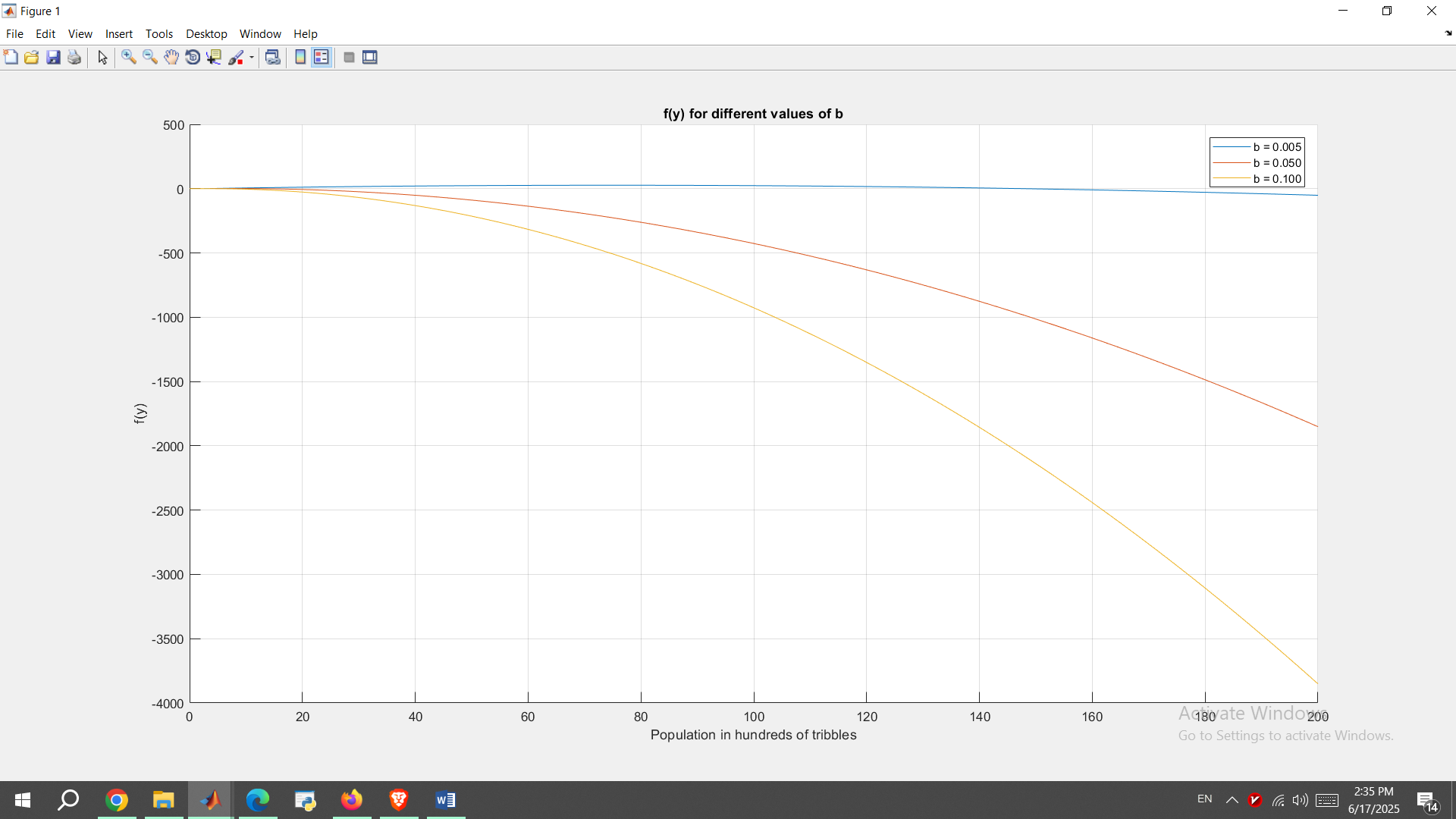


Figure 1 Population of tribbles for different values of b

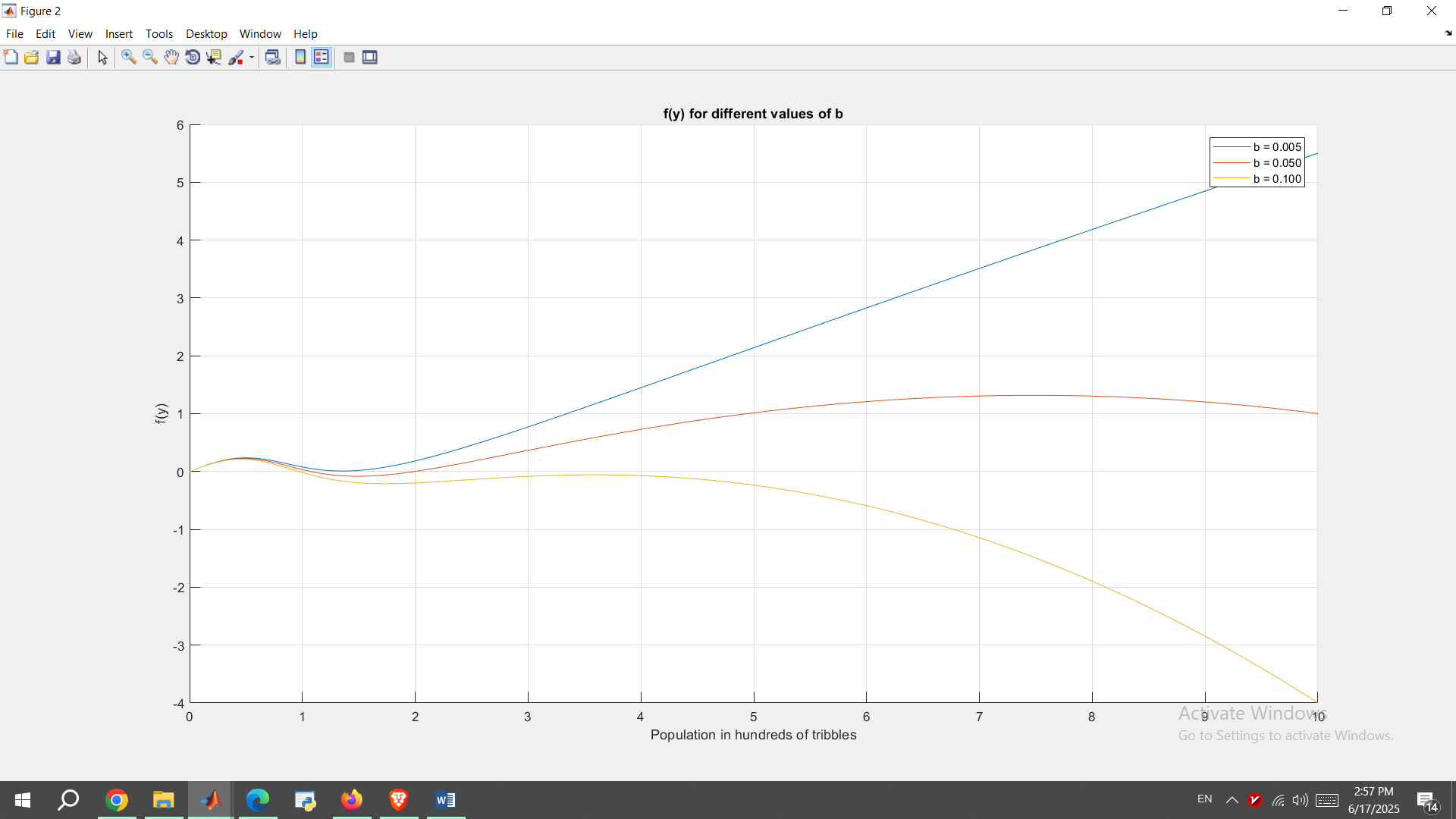


Figure 2 Population of tribbles (smaller interval)

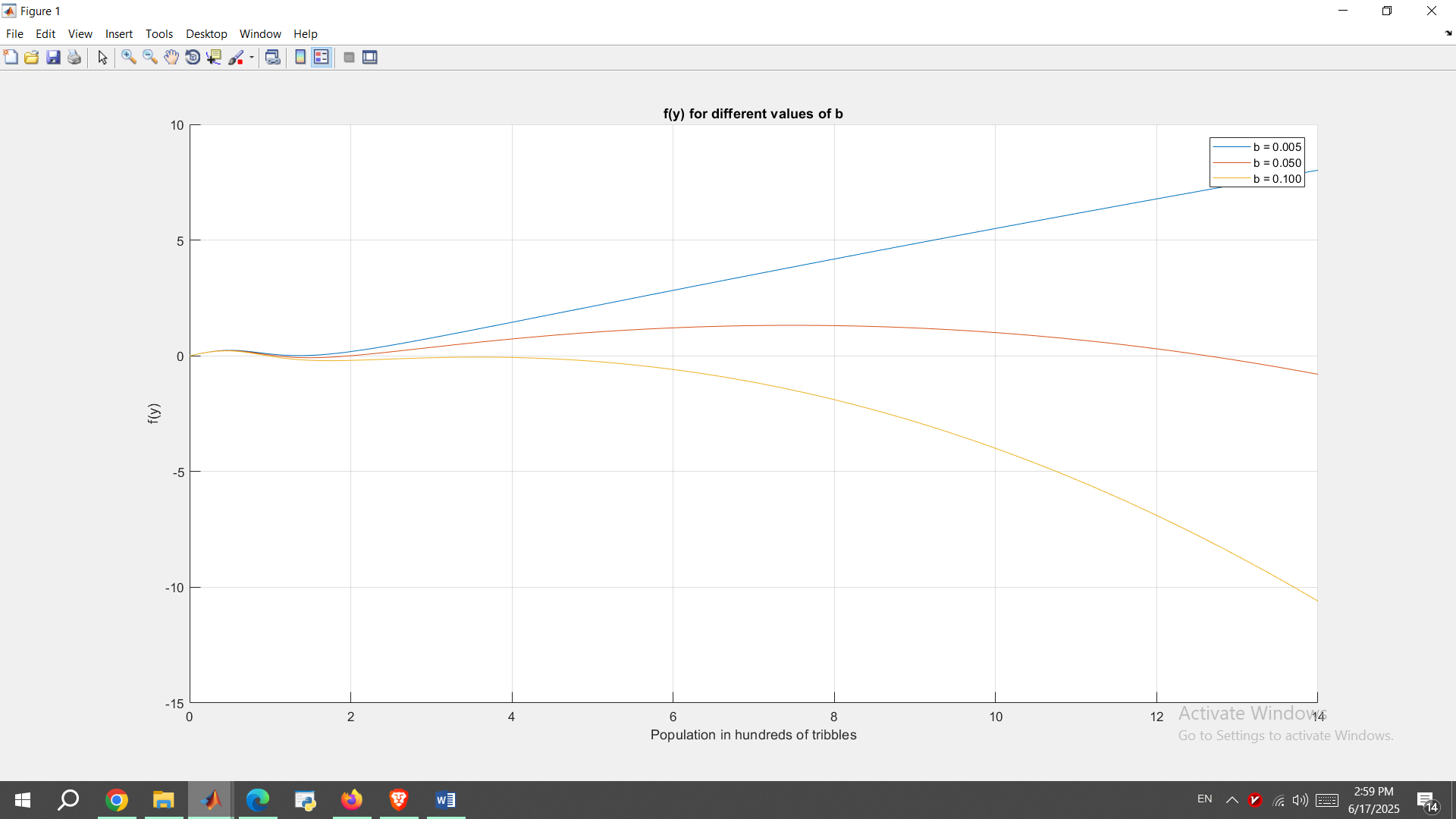


Figure 3 Population of tribbles (another smaller interval)

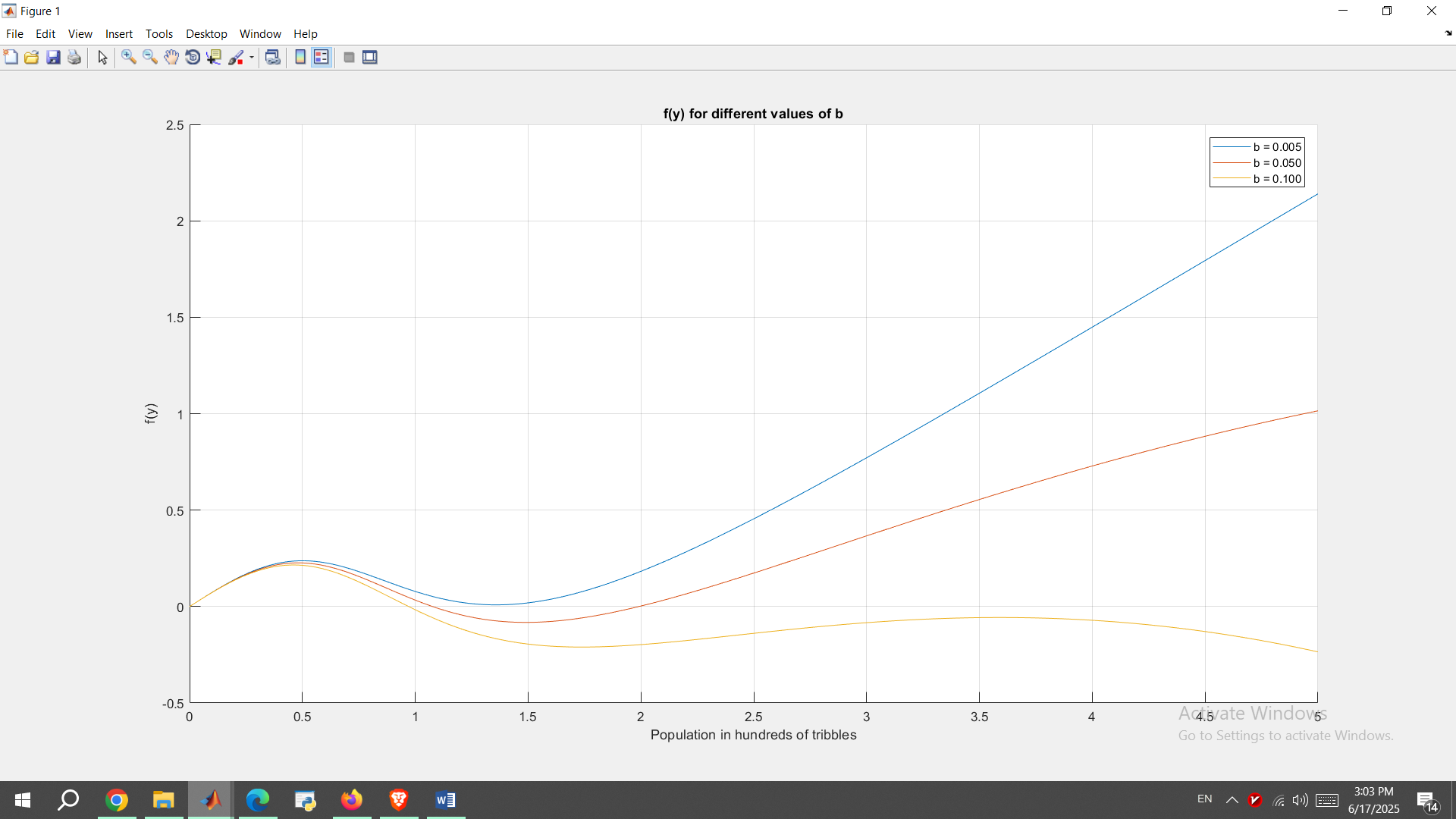


Figure 4 Population of tribbles (another smaller interval)

The equilibrium points are the points where f(y) touches the zero horizontal line. From these plots, we can see that:

**For b = 0.005: Equilibrium points = {0, 147.97}**

**For b = 0.05: Equilibrium points = {0, 1.08, 2, 12.6}**

**For b = 0.10: Equilibrium points = {0, 0.97}**

In the next step, we want to plot the direction fields of our ODE for different values of b.

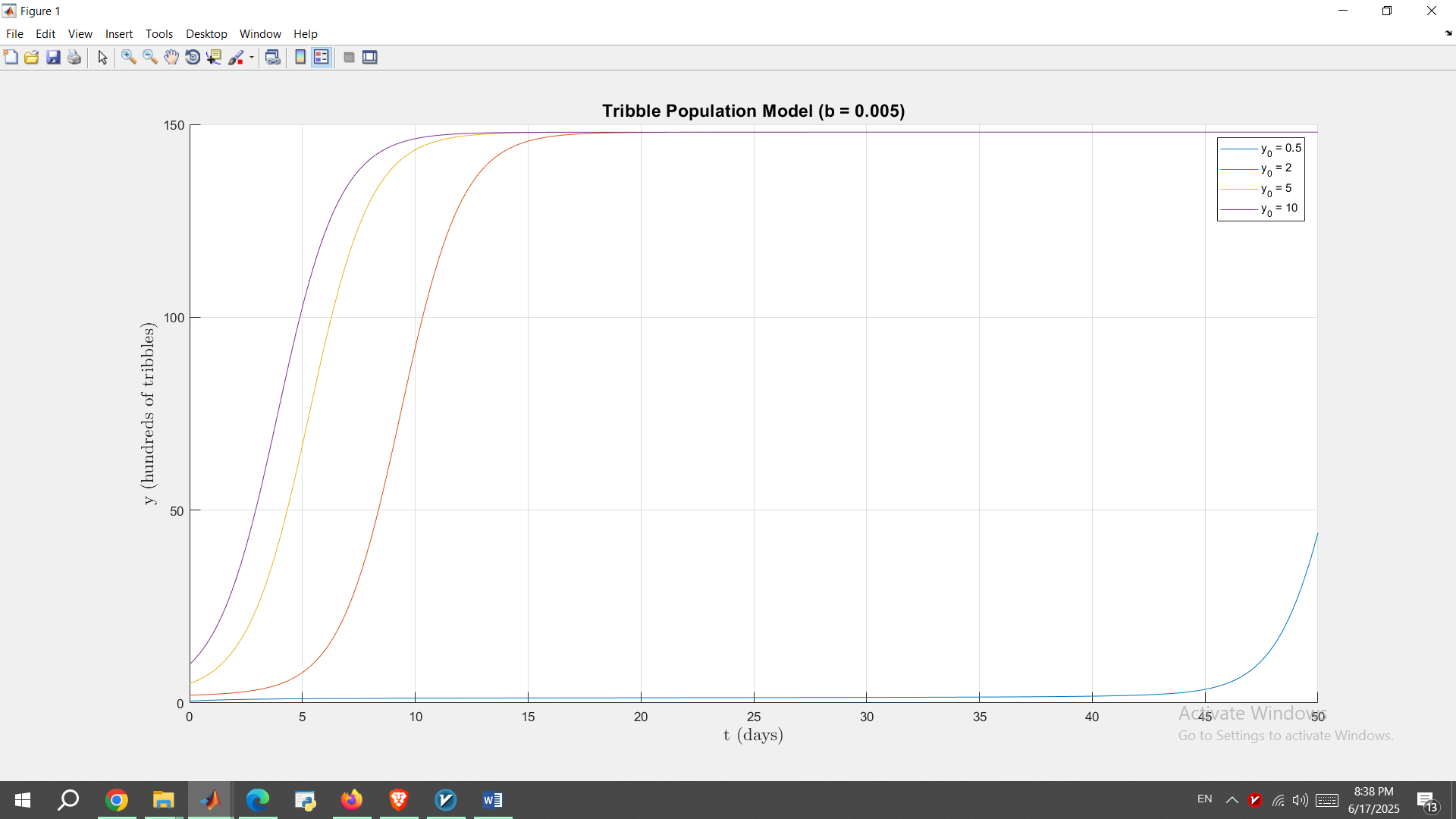


Figure 5 Direction filed by MATLAB for b = 0.005

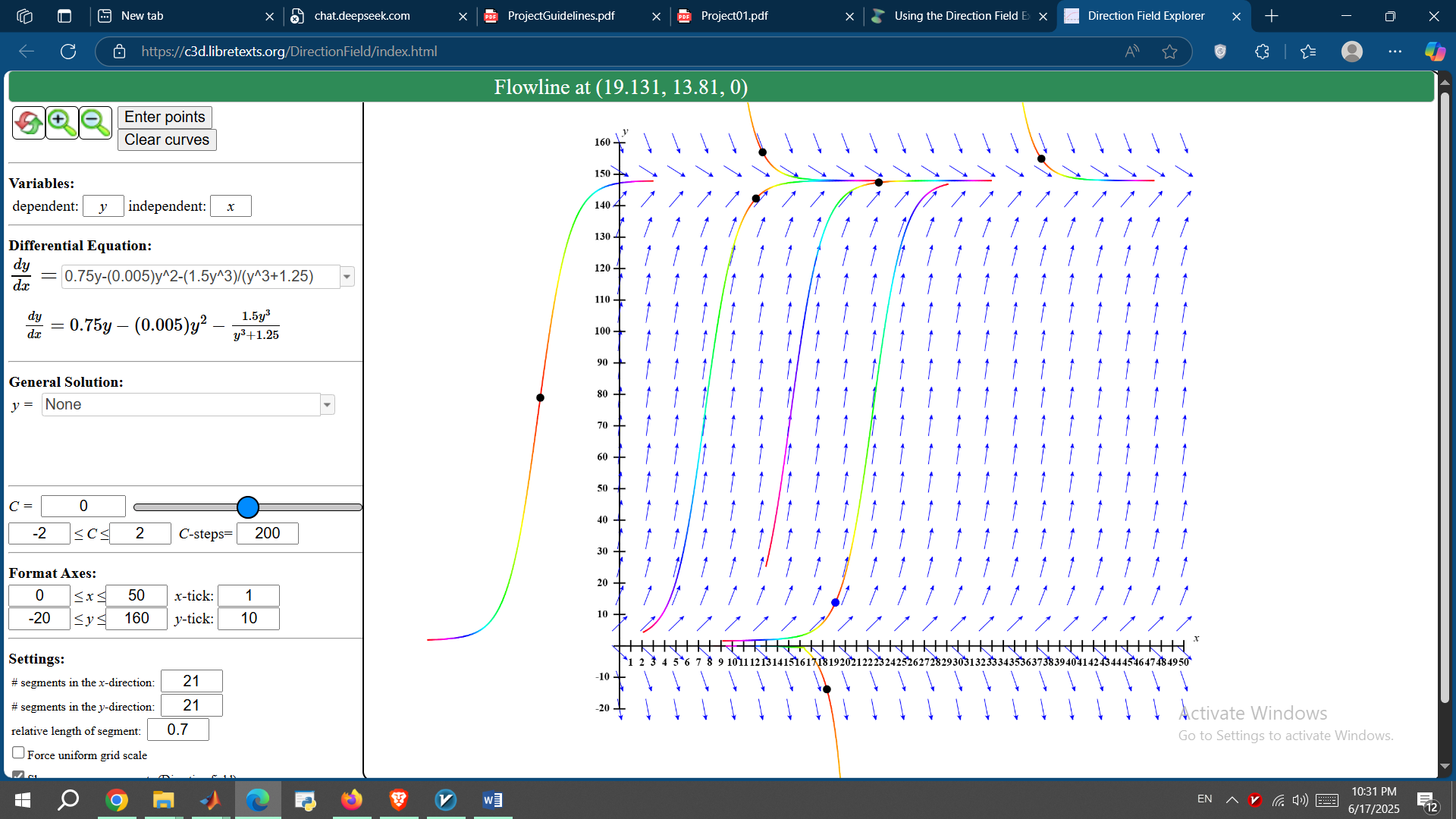


Figure 6 Direction filed by CalcPlot3D for b = 0.005

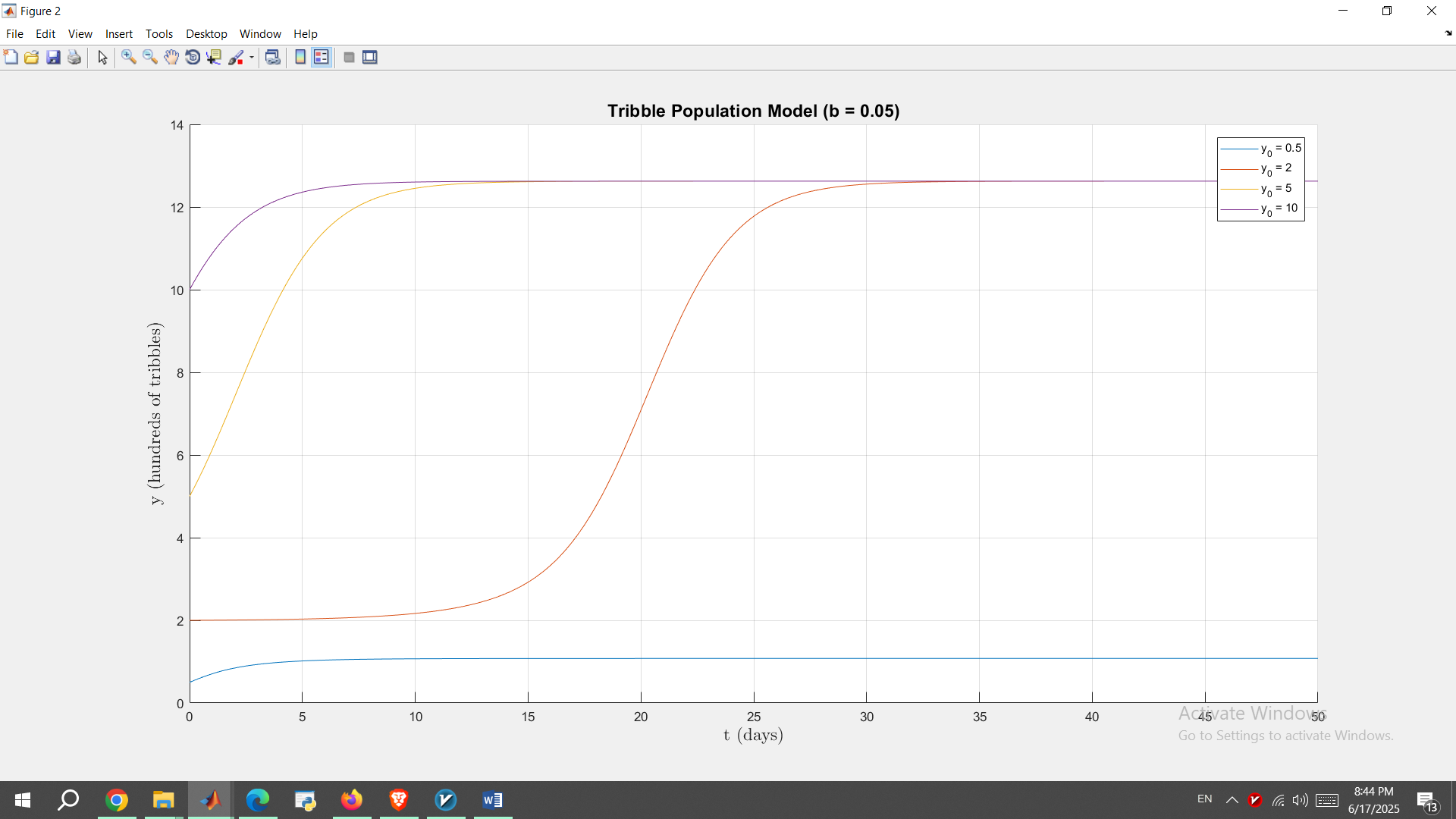


Figure 7 Direction filed by MATLAB for b = 0.05

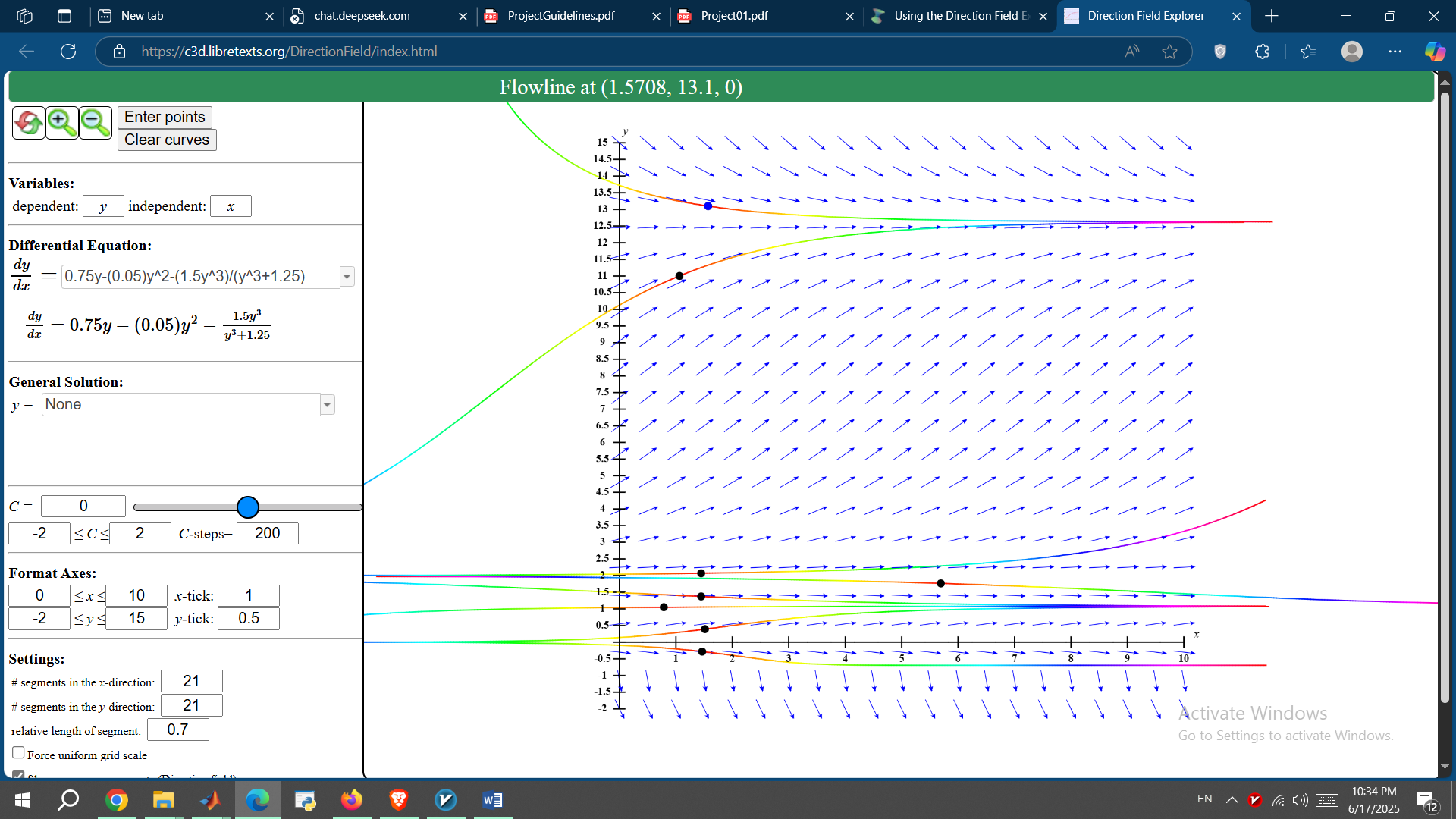


Figure 8 Direction filed by CalcPlot3D for b = 0.05

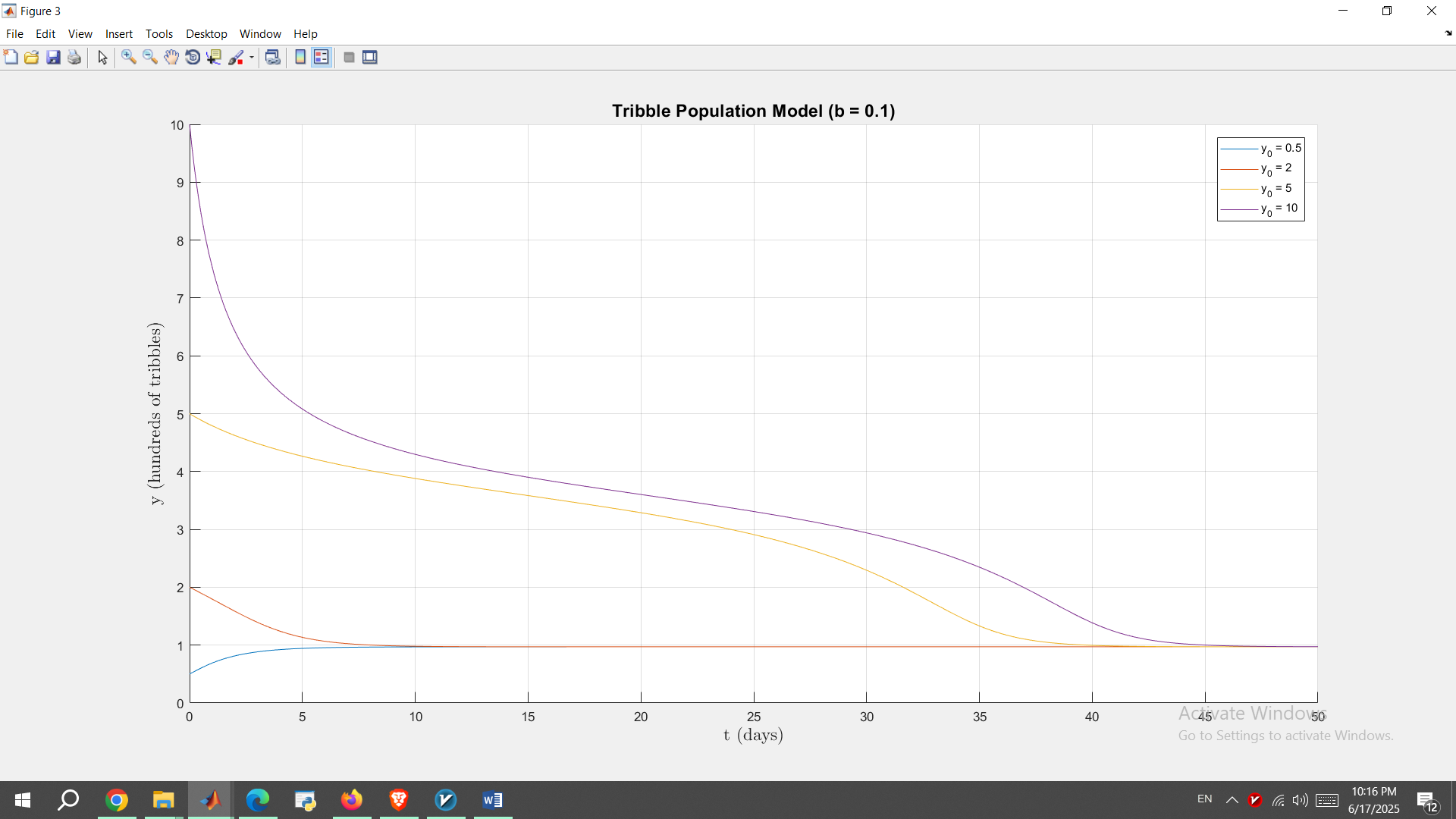


Figure 9 Direction filed by MATLAB for b = 0.10

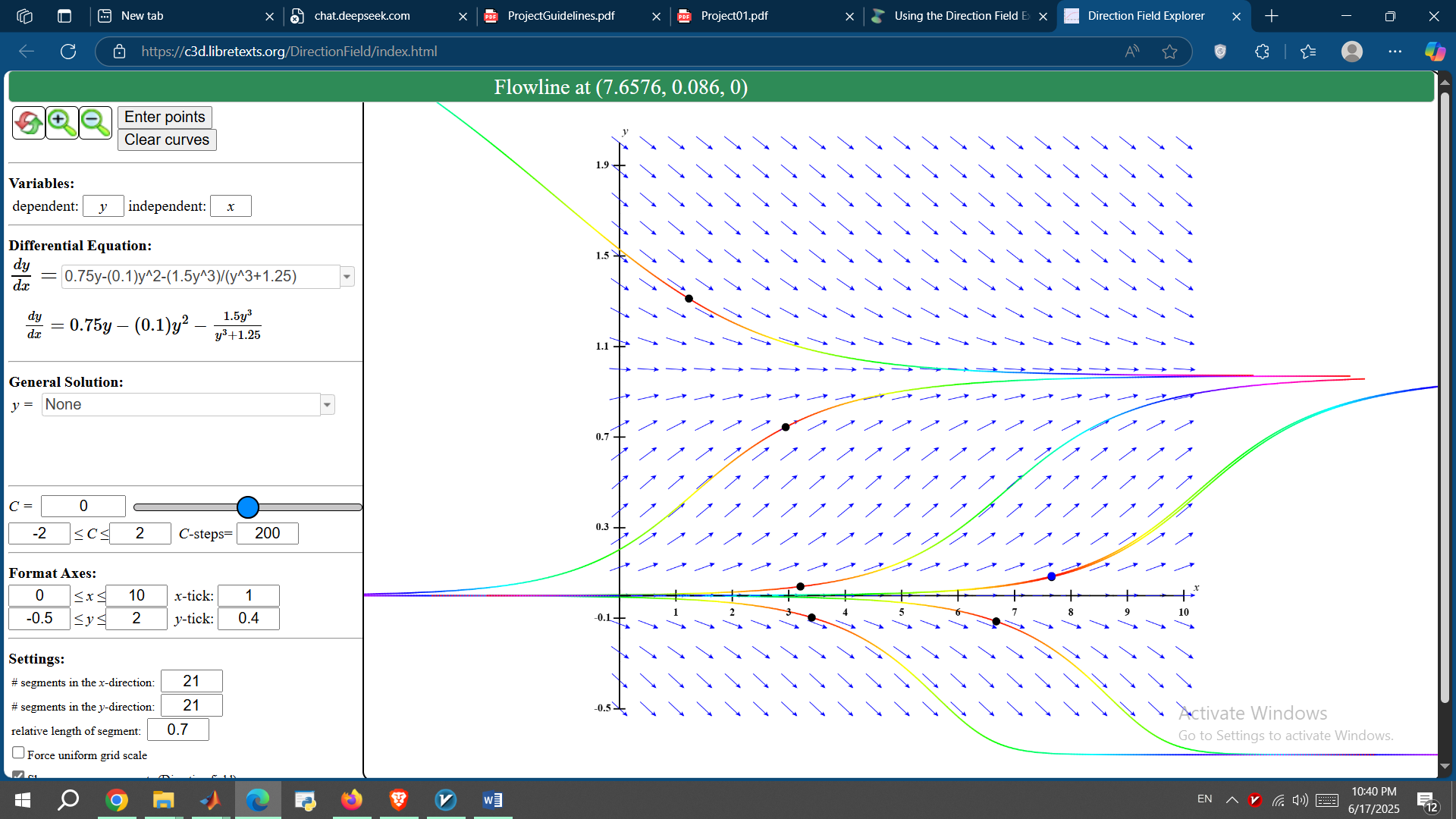


Figure 10 Direction filed by CalcPlot3D for b = 0.10

From these plots, we can conclude that for b = 0.005:

|  |  |
| --- | --- |
| **Equilibrium** | **Stability** |
| 0 | Unstable |
| 147.97 | Stable |

For b = 0.05:

|  |  |
| --- | --- |
| **Equilibrium** | **Stability** |
| 0 | Unstable |
| 1.08 | Stable |
| 2 | Unstable |
| 12.6 | Stable |

From these plots, we can conclude that for b = 0.10:

|  |  |
| --- | --- |
| **Equilibrium** | **Stability** |
| 0 | Unstable |
| 0.97 | Stable |

In this stage, we will determine the basins of attraction for the stable equilibrium solutions.

For b = 0.005, equilibrium point is y\* = 0 is unstable and the only stable equilibrium point is y\* = 147.97. Hence, all solutions with y0 > 0 approach to 147.97 which means:

**Basin of attraction: y0 > 0**

For b = 0.005, we have equilibria:

( 0 ) (unstable)

( 1.08 ) (stable)

( 2 ) (unstable)

( 12.6 ) (stable)

And The unstable equilibrium at y = 2 separates the basins.

So, for 0 < y0 < 2, solutions approach 1.08 since 0 is unstable and 2 repels solutions downward.

**Basin of attraction for y\* = 1.08: 0 < y0 < 2**

For y0 > 2, solutions approach 12.6 as no upper equilibria to attract elsewhere. Thus:

**Basin of attraction for y\* = 12.6: y0 > 2**

For b = 0.10, equilibrium point is y\* = 0 is unstable and the only stable equilibrium point is y\* = 0.97. Hence, all solutions with y0 > 0 approach to 0.97 which means:

**Basin of attraction y\* = 0.97: y0 > 0**

Over a long period of time, for b = 0.005, population grows and adjusts to y = 147.97 hundreds of tribbles and any positive initial population stabilizes at 14797 tribbles.

For b = 0.05, over a long period of time, if initial population is between 0 and 2 hundred tribbles, the population stabilizes at 1.08\*100 = 108 tribbles over time and if initial population is greater then 200 tribbles,

the population stabilizes at 12.62\*100 = 1262 tribbles.

Finally, for b = 0.10, population grows and adjusts to y = 0.97 hundreds of tribbles or 0.97 \* 100 = 97 tribbles over time and any positive initial population stabilizes at 97 tribbles.

Now, let’s put all findings above into real world practice. Assume we want to stock the station with a sustainable population of 3 different knds of tribbles with different b values where:

* brown tribbles have b = 0.005
* white have b = 0.05
* grey have b = 0.10

In this case, we should prioritize a **high, stable population and minimal risk of extinction to avoid low equilibria**.

From previous analyze results, we know that:

Brown with b = 0.005 is stable at 14797 tribbles for any y0 > 0

So: No threshold + robust growth for this kind.

White with b = 0.05 is stable at 108 tribbles if 0 < y0 < 2 or 1260 tribbles if y0 > 200 tribbles

So: Risk of low population if understocked for white kind.

Grey with b = 0.10 is stable at 97 tribbles for y0 > 0 which is the lowest stable population.

Taking all these into consideration, assuming that population growth follows the logistic model with mentioned parameters as before, our final choice is to stock the **Brown tribbles** for the highest stable population of 14797 tribbles and no threshold risk.

In this step, we suppose the hunting function, H, now depends on t and y, such that

if we plot several solutions (different initial conditions) for b = 0.005, 0.05 and b = 0.10, we get:

For first 10 days:

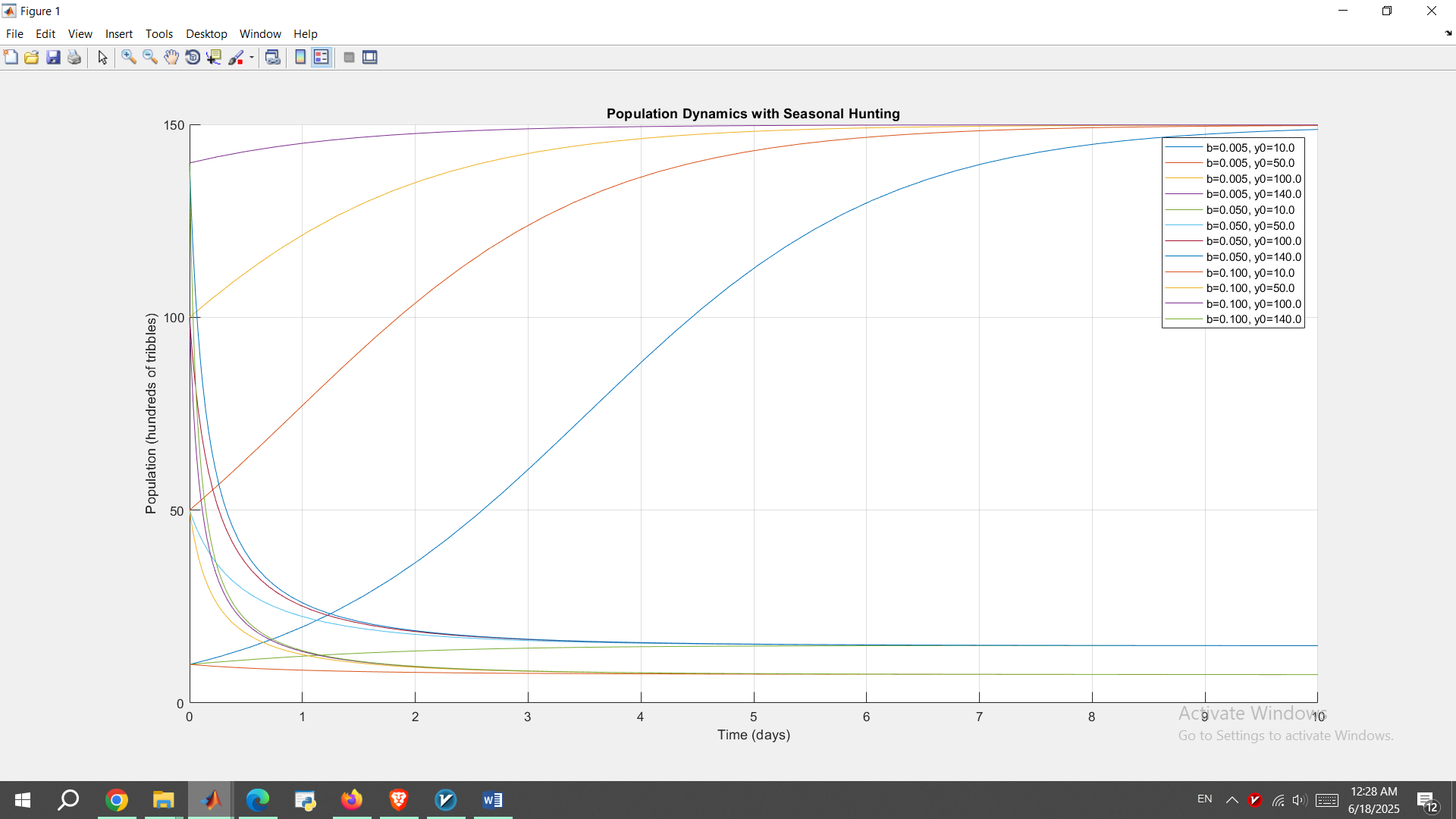


Figure 11 Population with harvest for 10 days

for 50 days, it will look like this:

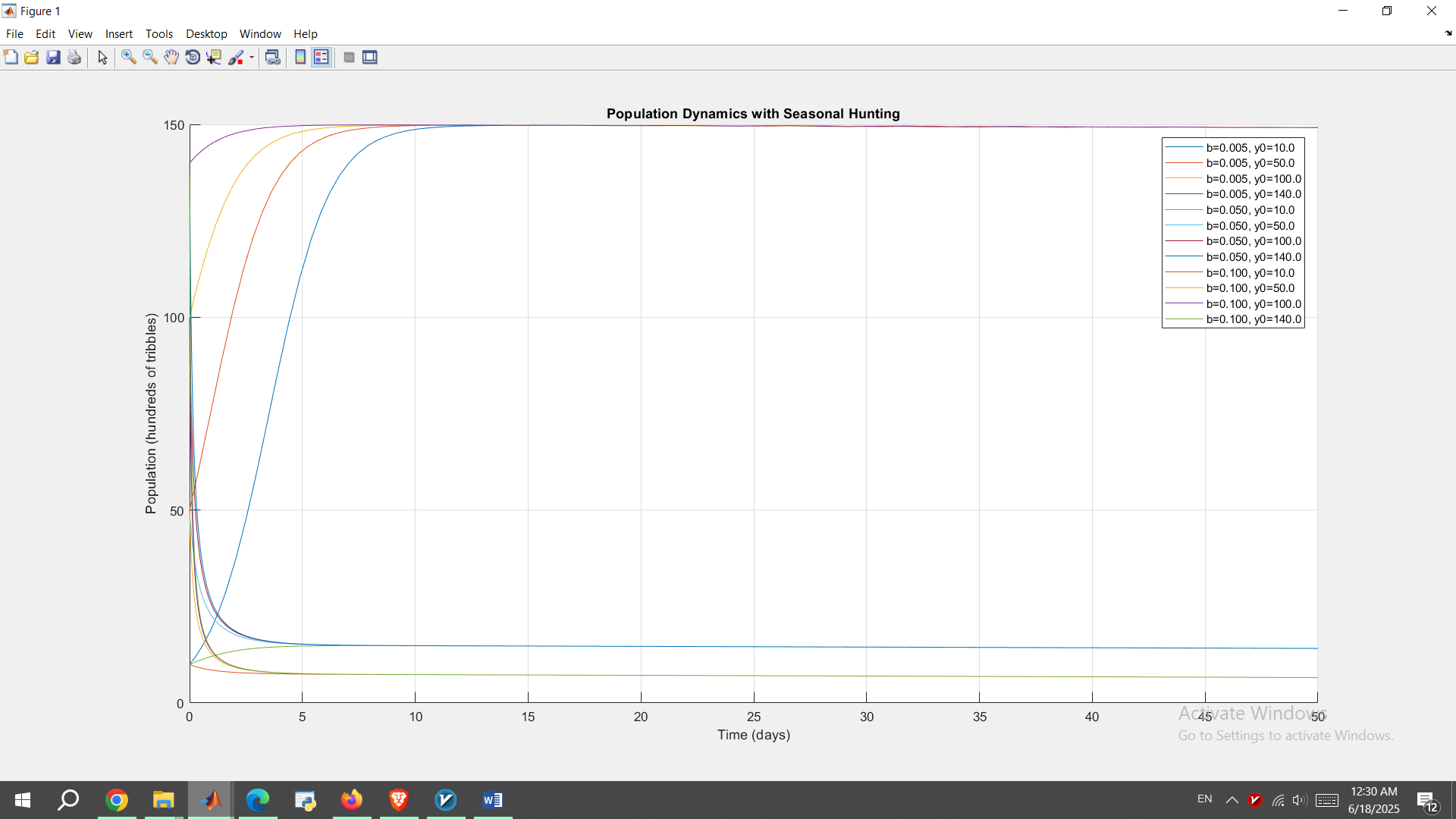


Figure 12 Population with harvest for 50 days

And for all 365 days of a year:

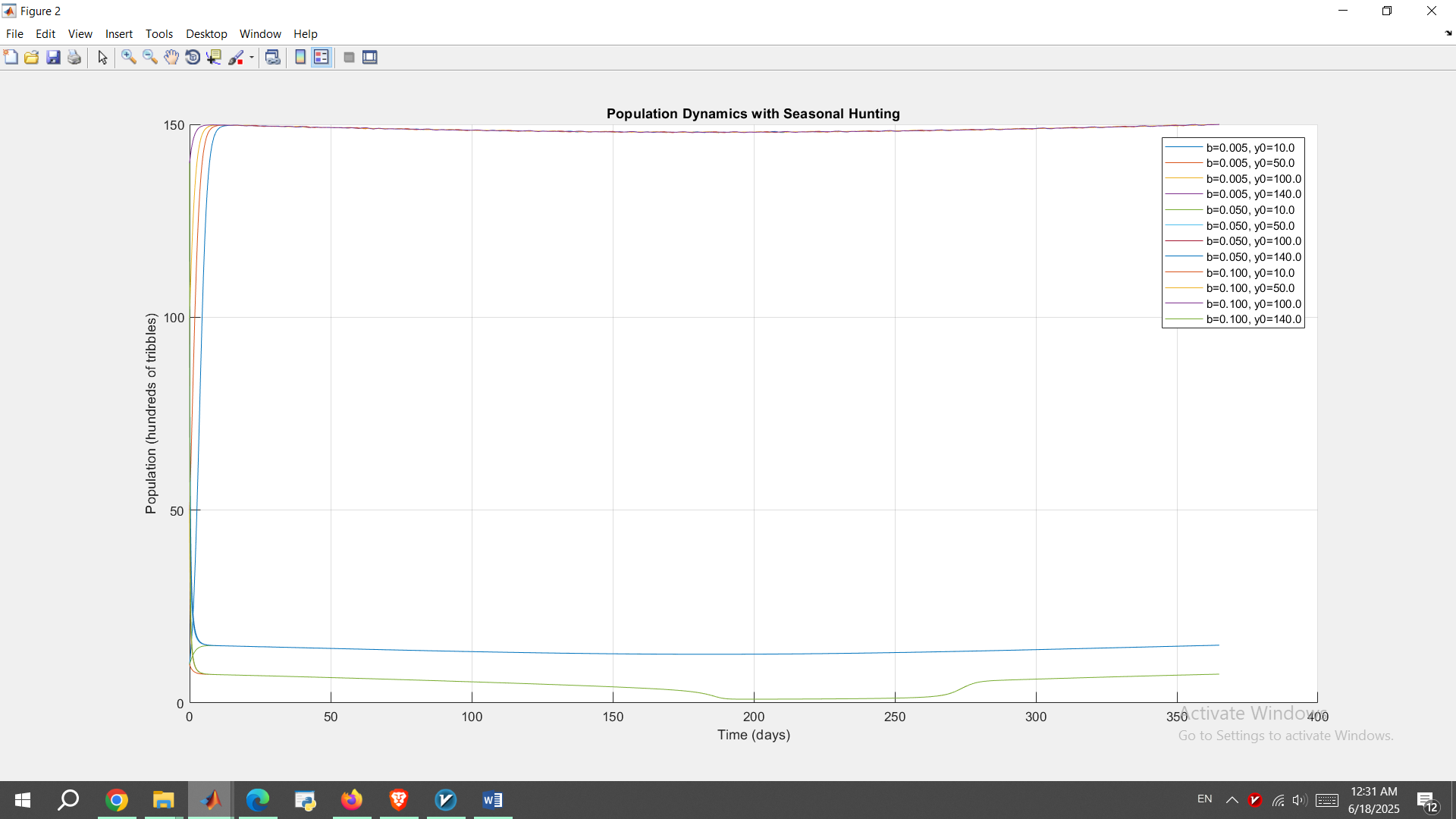


Figure 13 Population with harvest for a whole year

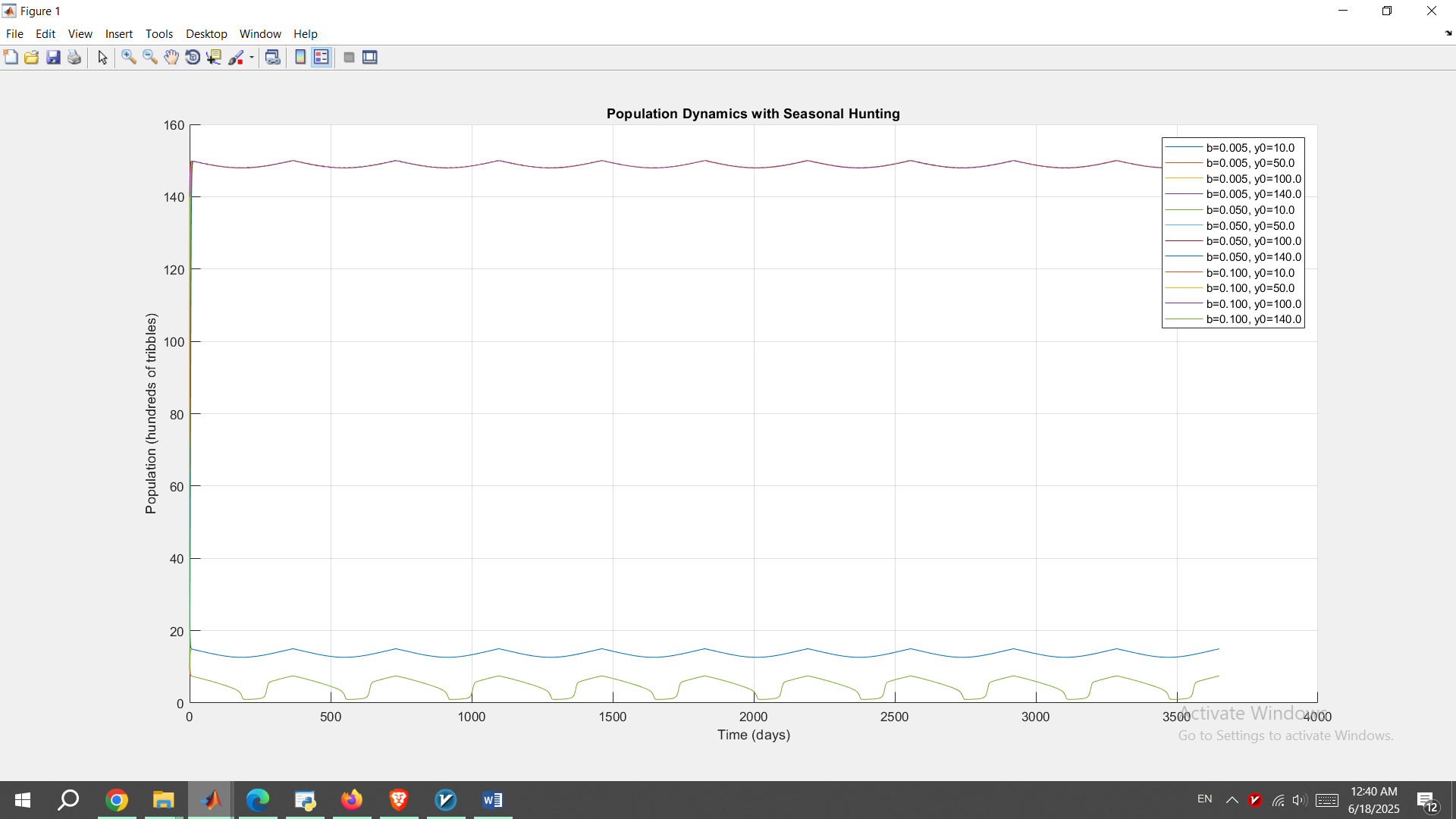


Figure 14 Population with harvest for 10 years

We can see that if we let the model run long enough (couple of years), we see the population shows **periodic oscillations** due to seasonal hunting.

More specifically, for small b like b =0.005, the population stabilizes near a high oscillatory equilibrium where for larger b values, the population stabilizes near lower oscillatory values.

This is due to the fact that in this new differential equation with this new harvest term, the **ODE is non-autonomous and equilibria no longer exist because of time dependence** and therefore solutions **oscillate with period 365 days**.

As for its analytical solution, we must say it is not possible since it is nonlinear and time-dependent which means it has **no closed-form solution**.

We have **no equilibria** in this new population dynamic model and only have periodic solutions.

The proposed model which we analyzed in this project has some major weak nesses. For instance, it ignores random events affecting small populations. It also has simplified hunting term, H(y), and this may not reflect practical, real-world Klingon behavior. Finally, it ignores the age structure and assumes identical tribbles.

In order to improve the model, we can refine the harvest term, H(t, y), with actual, better modeled hunting patterns which has resulted from real-world actual data. Also, we can add spatial diffusion or age classes.

**APPENDIX**

dirfiled.m:

function dirfield(f,IndVarVals,DepVarVals,SegmentLength)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plot direction field for the first order ODE y' = f(t,y)

% IndVarVals - vector of grid points of the independent variable

% DepVarVals - vector of grid points of the dependent variable

% f is an anonymous (@) function or name of an m-file in single quotes

% SegmentLength - length of tick marks

%

% Usage example: y' = -y^2 + t

% show direction field for t in [-1,3], y in [-2,2]

% using a spacing of .2 for both t and y:

%

% f = @(t,y) -y^2+t

% dirfield(f, -1:.2:3, -2:.2:2, 0.35)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set up the grid on which the direction field is plotted

% and compute the spacing between grid points.

[tg,yg] = meshgrid(IndVarVals,DepVarVals);

DeltaInd = IndVarVals(2) - IndVarVals(1);

DeltaDep = DepVarVals(2) - DepVarVals(1);

% Fix up the function if necessary.

if isa(f,'function\_handle')

f = eval(vectorize(f));

end

% Evaluate the function on the grid and compute the scaling factor, s,

% which helps keep the line segments' lengths relatively constant, and

% create the direction field on the grid.

z = feval(f,tg,yg);

s = 1./max(1/DeltaInd,abs(z)./DeltaDep)\*SegmentLength;

quiver(tg,yg,s,s.\*z,0,'.r'); hold on;

quiver(tg,yg,-s,-s.\*z,0,'.r');

% Fix up the axes.

axis([IndVarVals(1)-DeltaInd/2,IndVarVals(end)+DeltaInd/2, ...

DepVarVals(1)-DeltaDep/2,DepVarVals(end)+DeltaDep/2])

end

dirfield\_plot.m:

a = 0.75;

p = 1.5;

q = 1.25;

b\_values = [0.005, 0.05, 0.1];

y\_range = linspace(0, 5, 1000);

figure;

hold on;

for b = b\_values

H = @(y) (p \* y.^3) ./ (y.^3 + q);

f = @(y) a \* y - b \* y.^2 - H(y);

plot(y\_range, f(y\_range), 'DisplayName', sprintf('b = %.3f', b));

end

xlabel('Population in hundreds of tribbles');

ylabel('f(y)');

title('f(y) for different values of b');

legend('show');

grid on;

single\_ODE.m:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Solving the Tribble Population Model with ode45

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Housekeeping

clc; clear; close all;

% ODE45 options

options = odeset('AbsTol',1e-10,'RelTol',1e-7);

a = 0;

b = 50;

h = 0.1;

tt = a:h:b;

a\_param = 0.75;

p = 1.5;

q = 1.25;

b\_values = [0.005, 0.05, 0.10];

y0\_vals = [0.5, 2, 5, 10];

for b\_param = b\_values

figure;

hold on;

for y0 = y0\_vals

[t, y] = ode45(@(t, y) rhs(t, y, a\_param, b\_param, p, q), tt, y0, options);

plot(t, y, 'DisplayName', ['y\_0 = ', num2str(y0)]);

end

xlabel('t (days)', 'Interpreter', 'latex', 'FontSize', 14);

ylabel('y (hundreds of tribbles)', 'Interpreter', 'latex', 'FontSize', 14);

title(['Tribble Population Model (b = ', num2str(b\_param), ')'], 'FontSize', 14);

legend('show');

grid on;

end

function dydt = rhs(t, y, a, b, p, q)

dydt = a\*y - b\*y.^2 - (p\*y.^3)./(y.^3 + q);

end

H\_y\_t.m:

a = 0.75; p = 1.5; q = 1.25;

b\_values = [0.005, 0.05, 0.1];

T = 10\*365;

t\_span = [0, T];

H\_seasonal = @(t, y) (p \* y.^3) ./ (q + y.^3) .\* abs(sin(pi \* t / 365));

figure;

hold on;

for i = 1:length(b\_values)

b = b\_values(i);

f\_seasonal = @(t, y) a \* y - b \* y.^2 - H\_seasonal(t, y);

for y0 = [10, 50, 100, 140]

[t, y] = ode45(f\_seasonal, t\_span, y0);

plot(t, y, 'DisplayName', sprintf('b=%.3f, y0=%.1f', b, y0));

end

end

xlabel('Time (days)');

ylabel('Population (hundreds of tribbles)');

title('Population Dynamics with Seasonal Hunting');

legend('show');

grid on;

Manual calculations for analytical solution of tribble population in absence of harvest term:

