A Population Growth Model for P. aeruginosa

Modifying the filamentous bacteriophage Pf4 could impact Pseudomonas aeruginosa's growth and other attributes due to their interdependent relationship. Pf4 promotes biofilm formation crucial for bacterial survival and pathogenicity. It also shields the bacteria from antibiotics, affecting treatment efficacy. Furthermore, Pf4 helps modulate host immune responses to favor infection, aiding the bacterium in evading host immune detection. Certain Pf4 variants reduce virulence-associated traits in Pseudomonas aeruginosa, indicating that Pf4 modifications can influence bacterial virulence and potentially growth. Pf4 is also integral for biofilm development, stress tolerance, and genetic variant formation in Pseudomonas aeruginosa, which are vital for its survival, growth, and pathogenicity under various conditions. Hence, any alteration to Pf4 could potentially affect these interactions and subsequently the growth and other phenotypic traits of Pseudomonas aeruginosa. [1], [2], [3].

Purpose of the Model: **We are interested in whether the engineered phage impact the growth of Pseudomonas aeruginosa.** In this study, we aim to explore the dynamics of growth curves in Pseudomonas aeruginosa and its phage interactions under varying parameter settings to better understand their relationships. Our investigative framework focuses on:

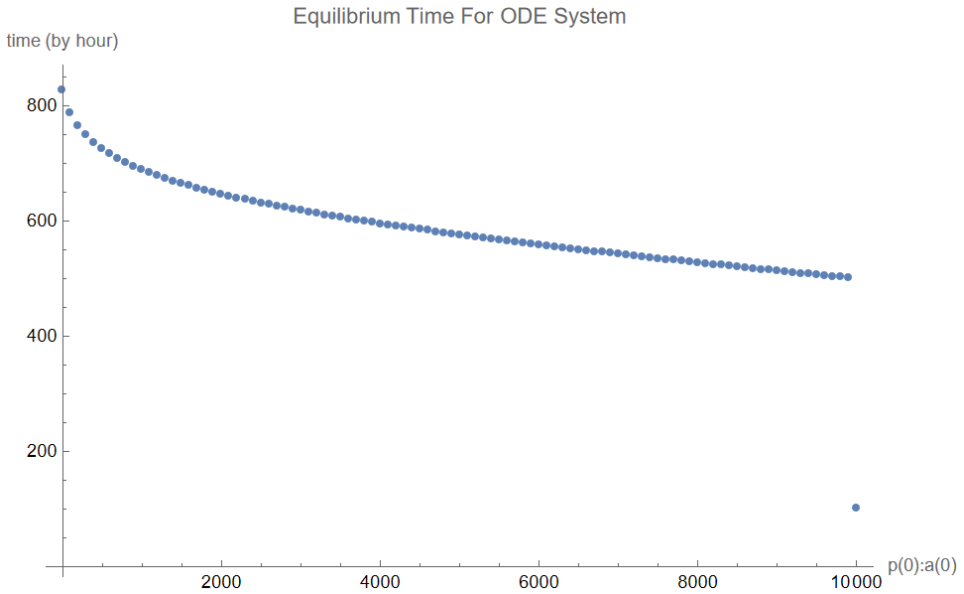
- Analyzing the effect of the initial ratio p(0):a(0) on the time required for growth curve stabilization.
- Examining the combined impact of parameter v and initial ratio p(0):a(0) on the time to reach a steady state, to uncover potential synergies or antagonisms affecting system stability.
- Extending the exploration to understand the intertwined influences of parameter alpha and the ratio p(0):a(0) on the time duration necessary for growth curve stabilization.
- Scrutinizing the influence of parameter u (probability of infection failure) and the initial ratio p(0):a(0) on the exact moment the growth curve attains a steady state.

Basic Assumptions: The phage, wild-type P. aeruginosa, and phage-infected P. aeruginosa are uniformly distributed in space.

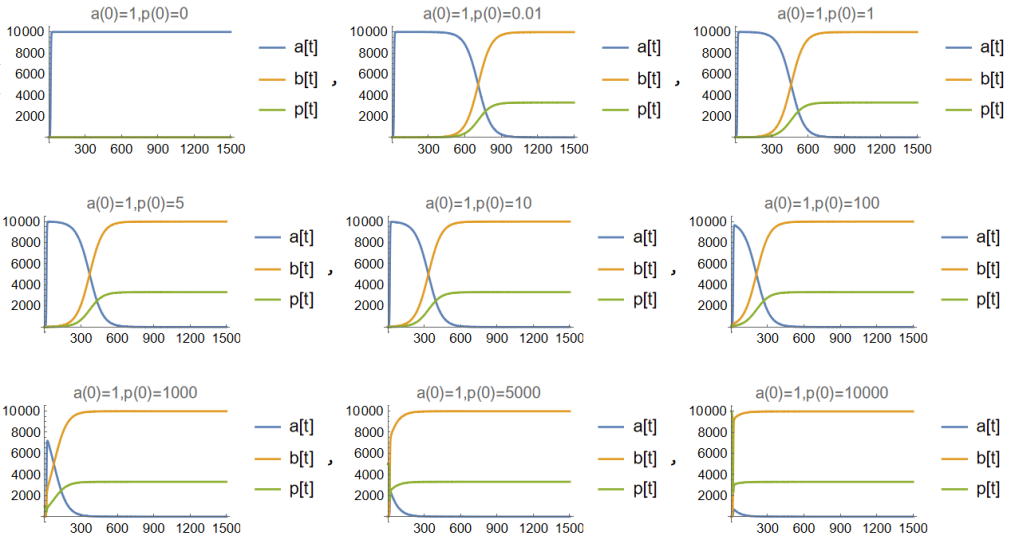
Task One: Explore the impact of the ratio p(0):a(0) on the time it takes for the growth curve to reach steady state.

Here, the concentration of wild-type P. aeruginosa changes with time as a(t), the concentration of phage-infected P. aeruginosa changes with time as b(t), the concentration of phage changes with time as p(t), and the environment accommodates is K, with a default value 10000.

Conclusions for task one: No matter how the ratio p(0):a(0) is selected, after a long enough time, the system will eventually reach its steady state, and a(t) tends to 0, p(t) tends to the environmental capacity K. This proves that temperate phage therapy is theoretically feasible to treat P. aeruginosa in condition of different initial ratio p(0):a(0). In addition, the relationship between the time needed for the system to reach the steady state and p(0):a(0) is as follows, which shows that the larger the ratio of p(0):a(0), the lower the time to reach the steady state.



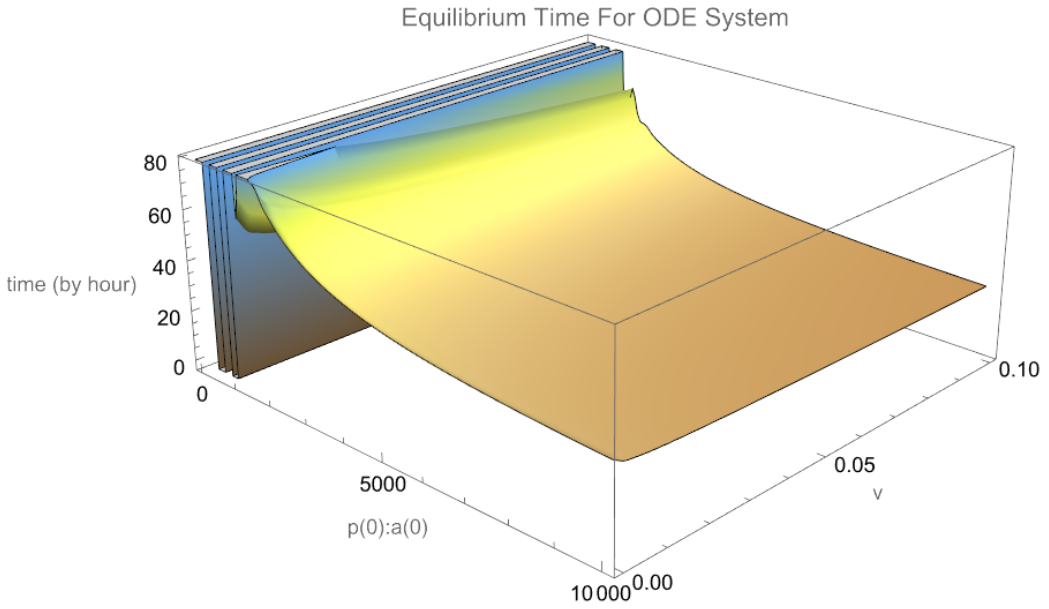
Here are also profiles for the growth curves when the ratio p(0):a(0) is equal to 0, 0.01, 1, 5, 10, 100, 1000, 5000, 10000, respectively to help better understand how the population changing over time:



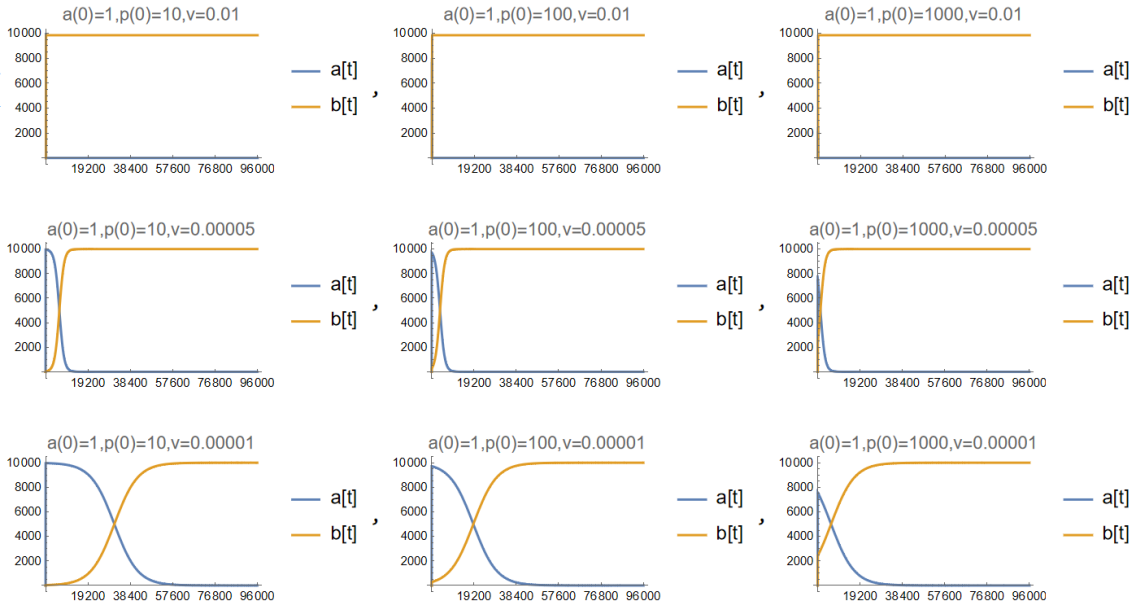
Task Two: Explore the impact of the selection of v and p(0):a(0) on the time needed for the growth curves reach steady state.

Recall that v is the lysis rate of lysogenic bacteria, the value range is $10^{-5} \sim 10^{-2}\ h^{-1}$.

Conclusions for task two: Within the reference range of v, after a long enough time, the system will eventually reach its steady state, where a(t) approaches 0 and p(t) converges to the environmental capacity denoted as K. This establishes a theoretical foundation for the viability of temperate phage therapy in treating P. aeruginosa, contingent upon v residing within its specified range. In addition, the relationship between the time needed for the system to reach the steady state and p(0):a(0), v is as shown in the following picture, which indicates that generally, when v remains unchanged, the larger the ratio of p(0):a(0), the lower the time to reach the steady state; when p(0):a(0) remains unchanged, the larger v induces the lower time to reach the steady state. Note that when p(0):a(0)<2000, there are tremendous vibrations within the computation, thus rendering the equilibrium time within this realm as dubious.



In addition, here are also profiles for the growth curves when the ratio p(0):a(0) is equal to 10, 100, 1000; v is equal to 0.01, 0.00005, 0.00001:



Task Three: Explore the impact of alpha and the ratio p(0):a(0) selection on the time it takes for the growth curve to reach steady state.

Here alpha is the phage produced by lysis of unit bacteria, and the default value is 248.

Conclusions for task three: Ideally, provided that the value of alpha is sufficiently large (exceeding 12.6), the variable a(t) will ultimately converge to 0, while b(t) will progressively approach the environmental capacity, K. Given a default alpha value of 248, which significantly surpasses the critical threshold of 12.6, the theoretical groundwork for employing temperate phage therapy to manage P. aeruginosa infections is established, contingent upon a regular alpha value. With a substantial alpha, an augmentation in its value will expedite the trajectory towards stability. Conversely, when alpha is proximal to its critical value (equated to 12.6), a stable coexistence between a(t) and b(t) is anticipated, reflecting a balanced interaction emblematic of a system in equilibrium, resonating with the biological implications inherent to the host-pathogen dynamics.

Task Four: Explore the impact of the selection of u and p(0):a(0) on the time when the growth curve reaches steady state.

Here u is the probability of infection failure, and the default value is 1-0.0769=0.9231

Conclusions for task four: Given the default parameters, the critical threshold for variable u is identified as 0.996. When u is either equal to or less than this critical threshold, the system will eventually transition into the desired state, where a(t) approaches zero and b(t) asymptotically reaches the environmental capacity, K. However, as u approaches the critical value from below, the time required to attain the desired state prolongs. Conversely, when u exceeds the critical thresh-

old, the system fails to achieve the desired state. Literature sources specify a reference value of u as 0.9231 (which is below 0.996), suggesting that employing temperate phage therapy to treat P. aeruginosa is theoretically viable under a regular u setting.

Overall Conclusions: temperate phage therapy is theoretically feasible to treat P. aeruginosa.

Reference:

[1]. Schwartzkopf CM, Robinson AJ, Ellenbecker M, et al. Tripartite interactions between filamentous Pf4 bacteriophage, *Pseudomonas aeruginosa*, and bacterivorous nematodes. *PLoS Pathog*. 2023;19(2):e1010925. Published 2023 Feb 17. doi:10.1371/journal.ppat.1010925

[2]. Ismail MH, Michie KA, Goh YF, et al. The Repressor C Protein, Pf4r, Controls Superinfection of Pseudomonas aeruginosa PAO1 by the Pf4 Filamentous Phage and Regulates Host Gene Expression. *Viruses*. 2021;13(8):1614. Published 2021 Aug 15. doi:10.3390/v13081614

[3]. Tortuel, D., Tahrioui, A., David, A., et al. Pf4 Phage Variant Infection Reduces Virulence-Associated Traits in *Pseudomonas aeruginosa*. *Microbiology Spectrum*, 2022;10(5), e01548-01522. https://doi.org/doi:10.1128/spectrum.01548-22

Coding Material:

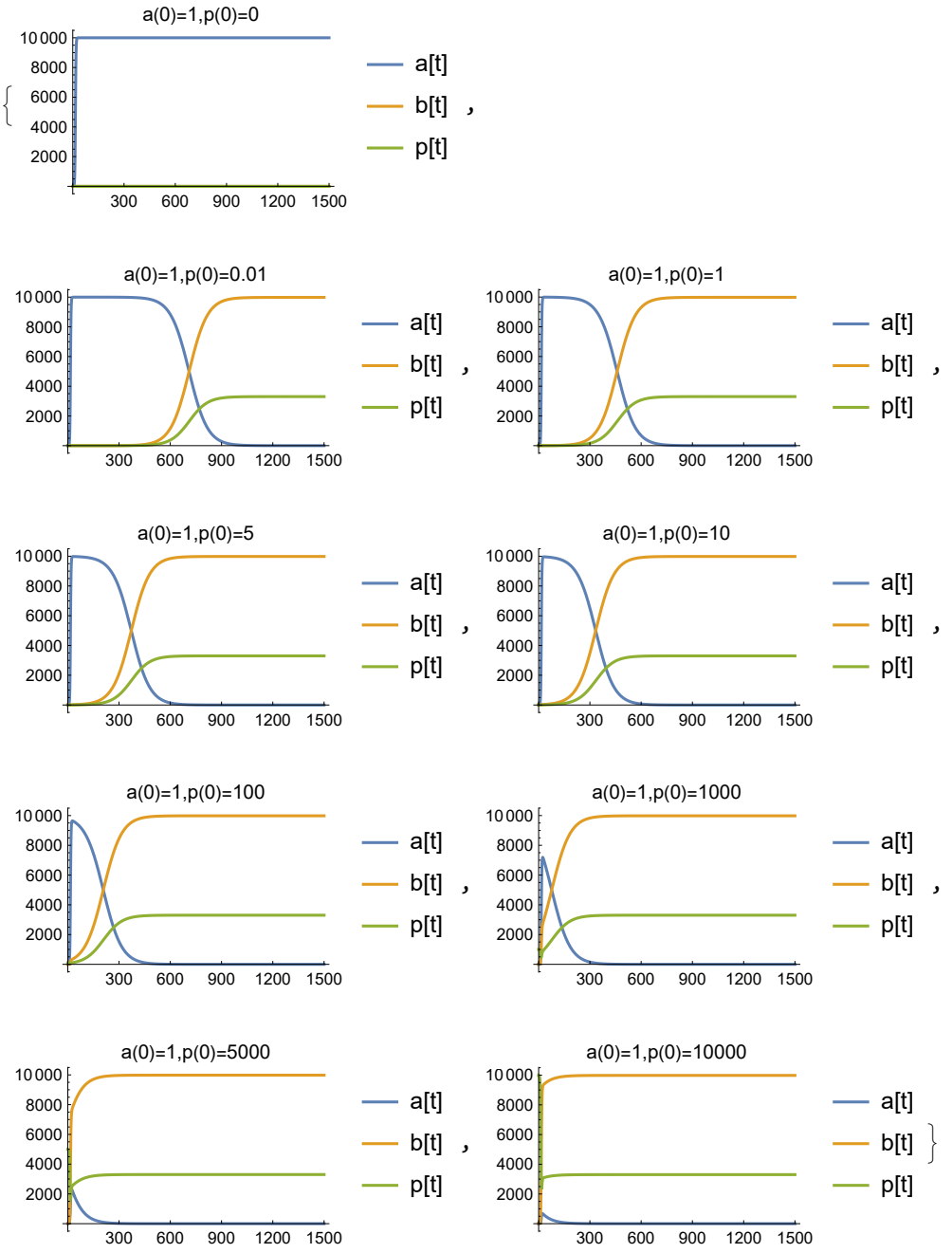
In[24]:=
```
(*Clear all to prevent chaotic in computation*)
ClearAll["Global`*"]
```

In[25]:=
```
(*paramerters setup*)
(*task1*)
end = 1500; (*end time of ode system*)
ra = 0.5861175448479;
rb = 0.5861175448479;
K = 10000;
u = 1 - 0.0769;
n = 1;
s = K / 3;
beta = 1;
alpha = 248;
v = 0.001;
```

In[35]:=
```
(*Plot the growth curves and compute the time required for the system to stable*)
solution[x_] := NDSolve[
  {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
   b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
   p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)), a[0] == 1,
   b[0] == 0, p[0] == x, WhenEvent[Abs[a[t]] < 0.1, Print[StringTemplate[
       "The equilibrium is `` for p(0)= ``"][t, x]]]}, {a, b, p}, {t, 0, end}
  ];
Table[Plot[Evaluate[{a[t], b[t], p[t]} /. solution[x]],
  {t, 0, end}, PlotLegends → {"a[t]", "b[t]", "p[t]"},
  Ticks → {{0, end / 5, end / 5 * 2, end / 5 * 3, end / 5 * 4, end}, Automatic},
  PlotLabel → StringTemplate["a(0)=1,p(0)=``"][x], PlotRange → All],
 {x, {0, 0.01, 1, 5, 10, 100, 1000, 5000, 10000}}]
```

```
The equilibrium is 1327.91 for p(0)= 0.01
The equilibrium is 1077.22 for p(0)= 1
The equilibrium is 989.576 for p(0)= 5
The equilibrium is 951.809 for p(0)= 10
The equilibrium is 825.846 for p(0)= 100
The equilibrium is 694.236 for p(0)= 1000
The equilibrium is 577.196 for p(0)= 5000
The equilibrium is 500.624 for p(0)= 10000
```

Out[36]=

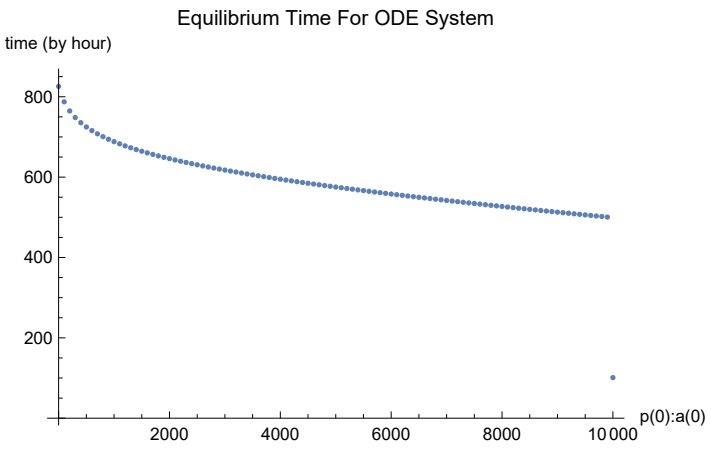


In[37]:=

```
equiTime = Array[# &, 101];
```

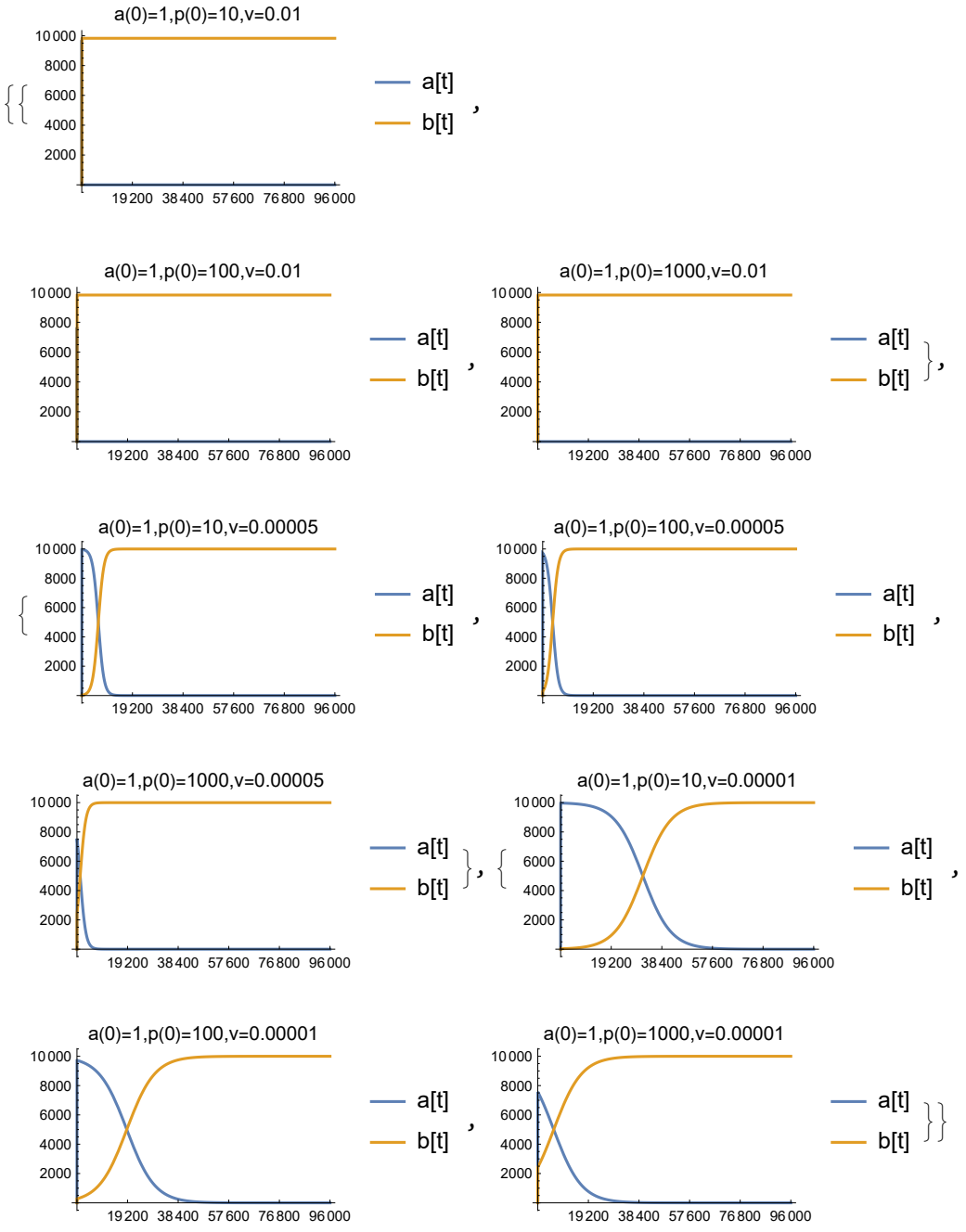
In[38]:=
```
Table[NDSolve[
  {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
   b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
   p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)), a[0] == 1, b[0] == 0,
   p[0] == x, WhenEvent[Abs[a[t]] < 0.1, equiTime[[x / 100]] = t]}, {a, b, p}, {t, 0, end}
  ], {x, Array[# &, 101, {0, 10000}]}];
ListPlot[Table[{x * 100 - 100, equiTime[[x]]}, {x, 1, 101}],
 PlotLabel → "Equilibrium Time For ODE System",
 AxesLabel → {"p(0):a(0)", "time (by hour)"}]
```

Out[39]=



In[40]:=
```
(*task2*)
end = 96000;
solution[v_, x_] := NDSolve[
  {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
   b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
   p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)),
   a[0] == 1, b[0] == 0, p[0] == x}, {a, b, p}, {t, 0, end}
  ];
Table[Plot[Evaluate[{a[t], b[t]} /. solution[v, x]],
  {t, 0, end}, PlotLegends → {"a[t]", "b[t]"},
  PlotLabel → StringTemplate["a(0)=1,p(0)=``,v=``"][x, v],
  AxesStyle → Directive[ FontSize → 8],
  Ticks → {{0, end / 5, end / 5 * 2, end / 5 * 3, end / 5 * 4, end}, Automatic}],
 {v, {0.01, 0.00005, 0.00001}}, {x, {10, 100, 1000}}]
```

Out[42]=

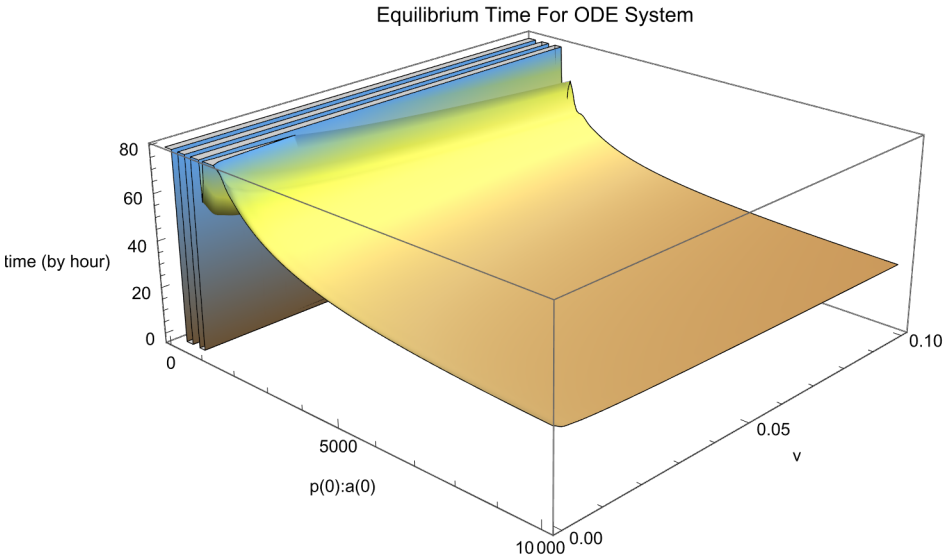

a(0)=1,p(0)=10,v=0.01
a(0)=1,p(0)=100,v=0.01
a(0)=1,p(0)=1000,v=0.01
a(0)=1,p(0)=10,v=0.00005
a(0)=1,p(0)=100,v=0.00005
a(0)=1,p(0)=1000,v=0.00005
a(0)=1,p(0)=10,v=0.00001
a(0)=1,p(0)=100,v=0.00001
a(0)=1,p(0)=1000,v=0.00001
a[t]
b[t]
10000
8000
6000
4000
2000
19200
38400
57600
76800
96000

In[43]:=
```
equiTime2 = Array[#1 &, {51, 51}];
apRatio = Array[# &, 51, {50, 10050}];
vArray = Array[# &, 51, {0.00001, 0.10001}];

Table[NDSolve[
  {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
   b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
   p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)),
   a[0] == 1, b[0] == 0, p[0] == x, WhenEvent[Abs[a[t]] < 0.1,
    equiTime2[[(x - 50) / 200 + 1]][[(v - 0.00001) / 0.002 + 1]] = t]}, {a, b, p}, {t, 0, end}
 ], {x, 50, 10050, 200}, {v, 0.00001, 0.10001, 0.002}];

data = Table[equiTime2[[(x - 50) / 200 + 1]][[(v - 0.00001) / 0.002 + 1]],
   {x, 50, 10050, 200}, {v, 0.00001, 0.10001, 0.002}];
S = ListPlot3D[data, PlotLabel → "Equilibrium Time For ODE System",
  AxesLabel → {"p(0):a(0)", "v", "time (by hour)"}, Mesh → None, InterpolationOrder → 3,
  ColorFunction → "SouthwestColors", DataRange → {{50, 10050}, {10^-5, 0.1}}]
```

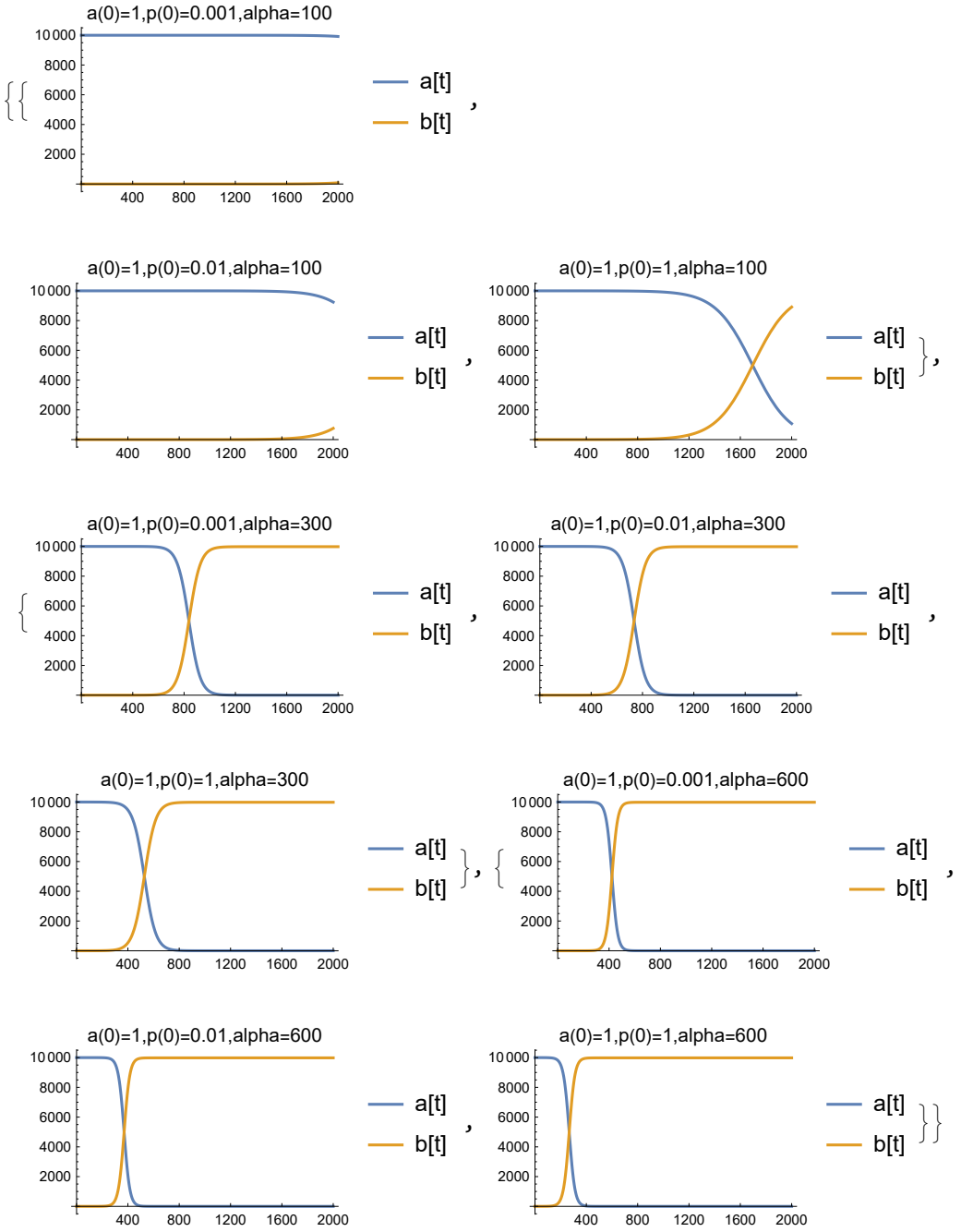
Out[48]=



In[49]:=
```
(*task3*)
end = 2000;
solution[alpha_, x_] := NDSolve[
    {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
     b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
     p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)),
     a[0] == K, b[0] == 0, p[0] == x}, {a, b, p}, {t, 0, end}
   ];
Print["When alpha is relatively large (>12.6):"]
Table[Plot[Evaluate[{a[t], b[t]} /. solution[alpha, x]],
  {t, 0, end}, PlotLegends → {"a[t]", "b[t]"},
  PlotLabel → StringTemplate["a(0)=1,p(0)=``,alpha=``"][x, alpha],
  AxesStyle → Directive[ FontSize → 8],
  Ticks → {{0, end / 5, end / 5 * 2, end / 5 * 3, end / 5 * 4, end}, Automatic}],
 {alpha, {100, 300, 600}}, {x, {0.001, 0.01, 1}}]
Print["When alpha is around critical value: "]
end = 100 000 000;
Table[Plot[Evaluate[{a[t], b[t]} /. solution[alpha, x]],
  {t, 0, end}, PlotLegends → {"a[t]", "b[t]"},
  PlotLabel → StringTemplate["a(0)=1,p(0)=``,alpha=``"][x, alpha],
  AxesStyle → Directive[ FontSize → 8], Ticks → {{0, end / 2, end}, Automatic}],
 {alpha, {12.4, 12.5, 12.6}}, {x, {0.001, 0.01, 1}}]
```

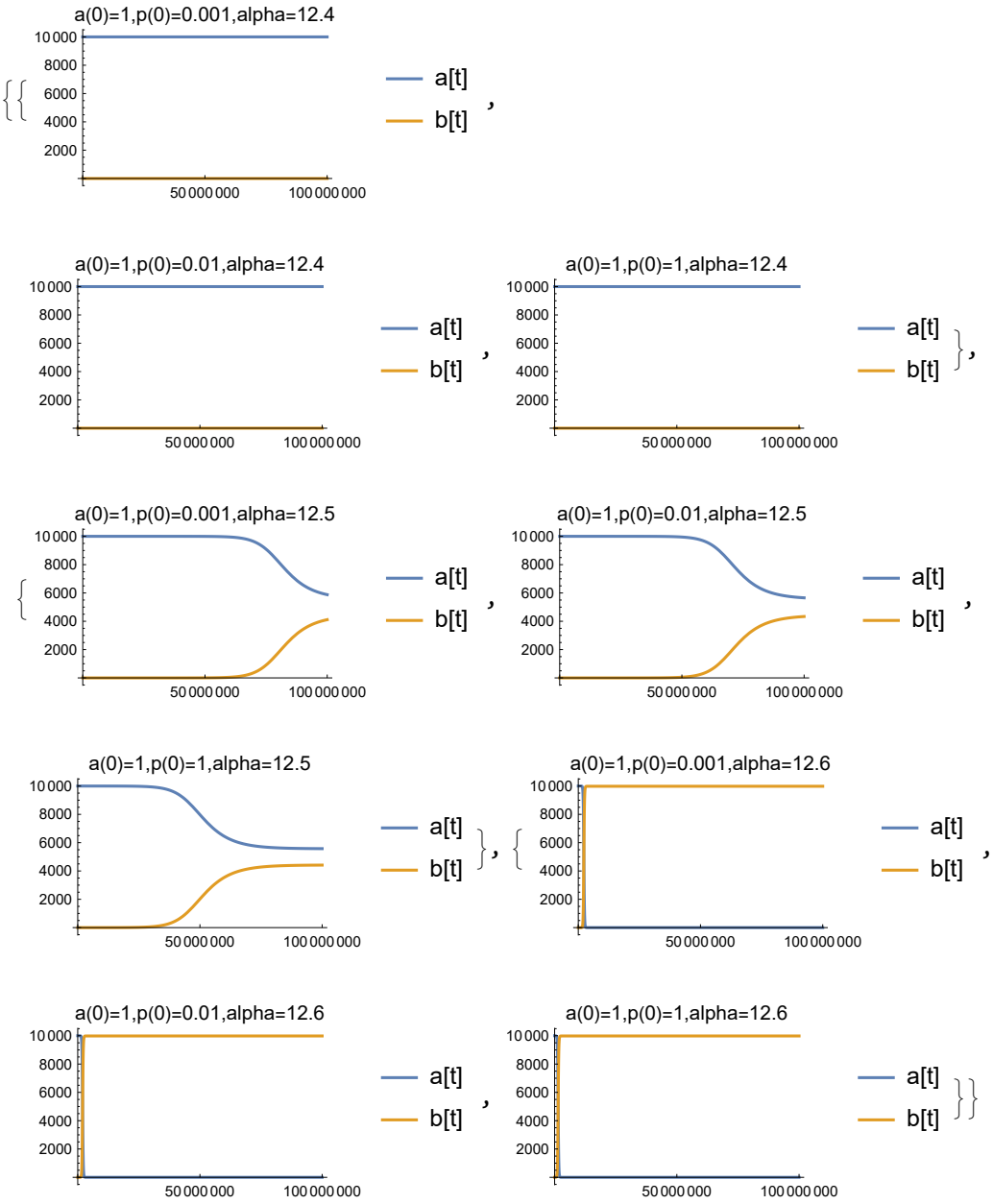
```
When alpha is relatively large (>12.6):
```

Out[52]=



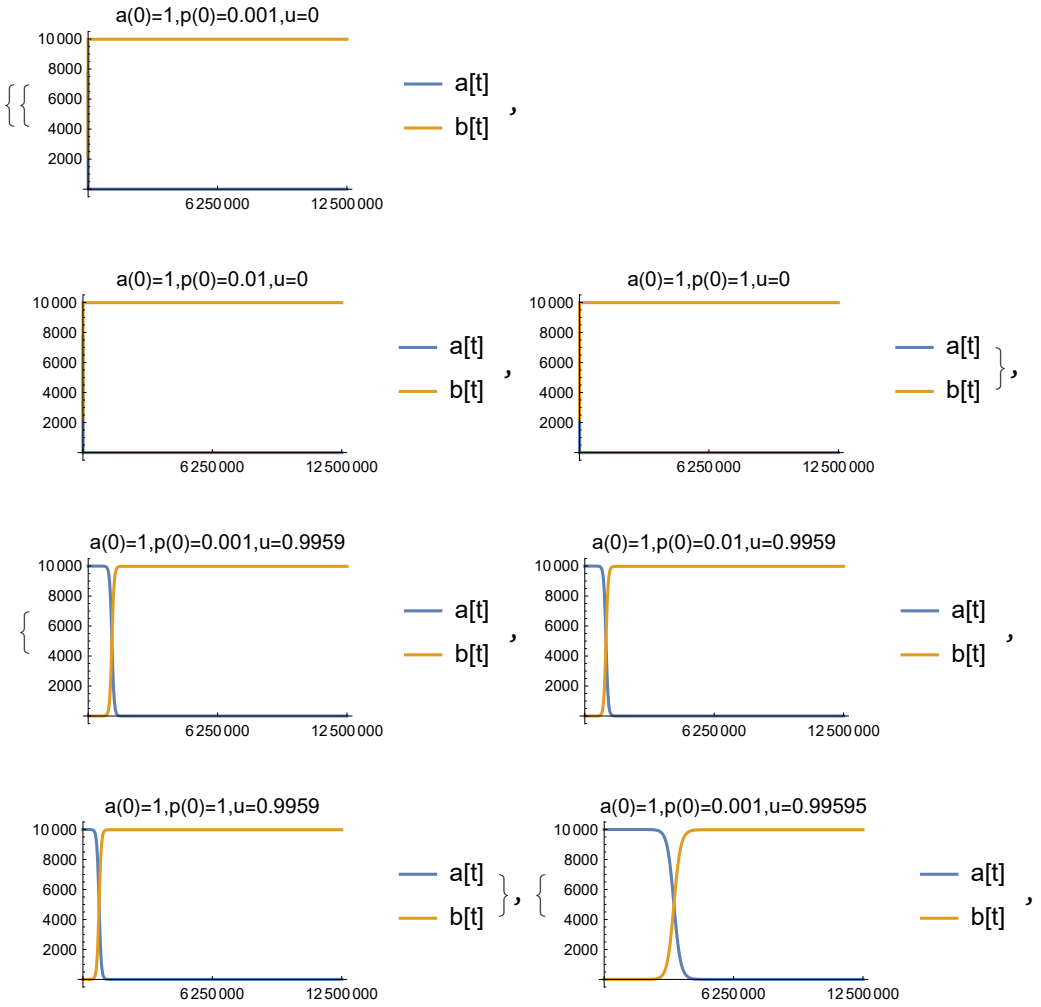
When alpha is around critical value:

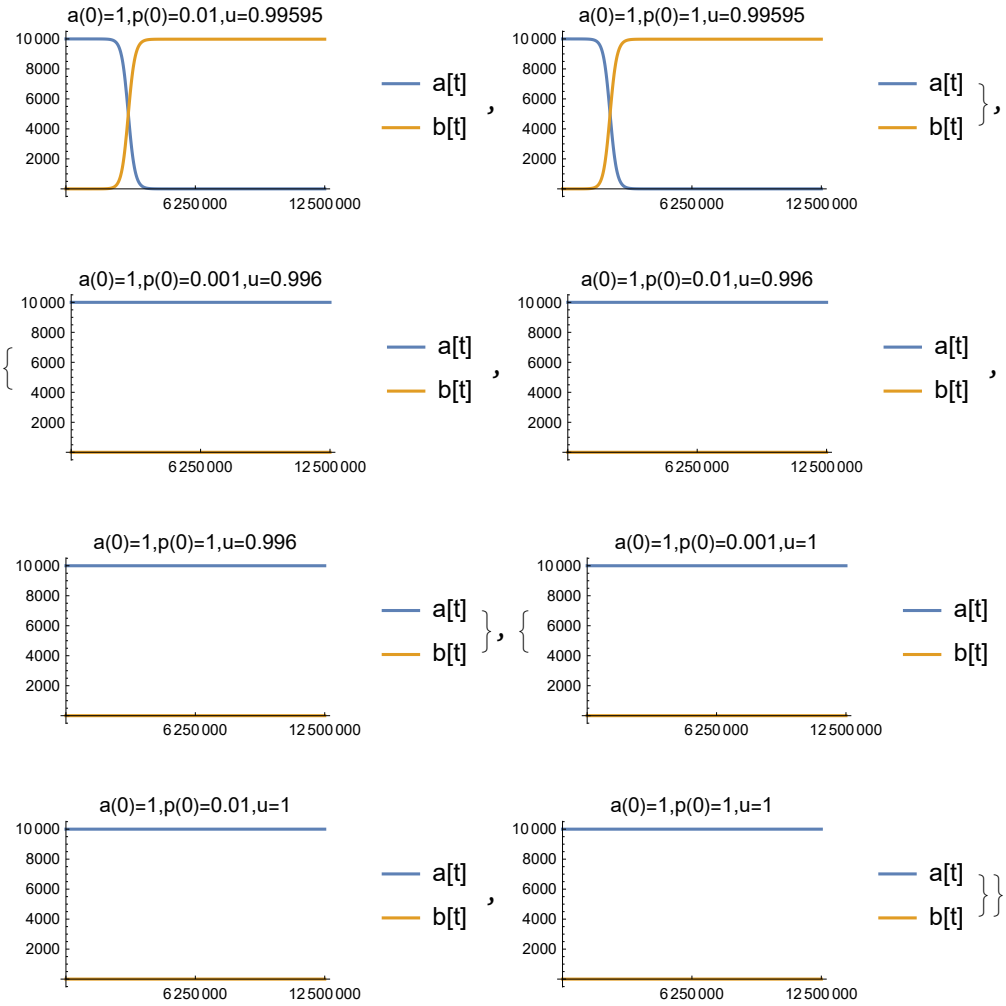
Out[55]=


a(0)=1,p(0)=0.001,alpha=12.4
a(0)=1,p(0)=0.01,alpha=12.4
a(0)=1,p(0)=1,alpha=12.4
a(0)=1,p(0)=0.001,alpha=12.5
a(0)=1,p(0)=0.01,alpha=12.5
a(0)=1,p(0)=1,alpha=12.5
a(0)=1,p(0)=0.001,alpha=12.6
a(0)=1,p(0)=0.01,alpha=12.6
a(0)=1,p(0)=1,alpha=12.6
10 000
8000
6000
4000
2000
50 000 000
100 000 000
a[t]
b[t]

In[56]:=
```
(*task4*)
end = 12 500 000;
solution[u_, x_] := NDSolve[
  {a'[t] == ra * a[t] * ((K - a[t] - b[t])/K) - a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))),
   b'[t] == rb * b[t] * ((K - a[t] - b[t])/K) + a[t] * (1 - u^(beta*p[t]*((a[t]+b[t])^(n-1)/(s^n+(a[t]+b[t])^n)))) - v * b[t],
   p'[t] == v * b[t] * alpha - beta * p[t] * ((a[t] + b[t])^n/(s^n + (a[t] + b[t])^n)),
   a[0] == K, b[0] == 0, p[0] == x}, {a, b, p}, {t, 0, end}
  ];
Table[Plot[Evaluate[{a[t], b[t]} /. solution[u, x]],
  {t, 0, end}, PlotLegends → {"a[t]", "b[t]"},
  PlotLabel → StringTemplate["a(0)=1,p(0)=``,u=``"][x, u],
  AxesStyle → Directive[FontSize → 8], Ticks → {{0, end / 2, end}, Automatic}],
 {u, {0, 0.9959, 0.99595, 0.996, 1}}, {x, {0.001, 0.01, 1}}]
```

Out[58]=




a(0)=1,p(0)=0.01,u=0.99595
a(0)=1,p(0)=1,u=0.99595
a(0)=1,p(0)=0.001,u=0.996
a(0)=1,p(0)=0.01,u=0.996
a(0)=1,p(0)=1,u=0.996
a(0)=1,p(0)=0.001,u=1
a(0)=1,p(0)=0.01,u=1
a(0)=1,p(0)=1,u=1
10000
8000
6000
4000
2000
6250000
12500000
a[t]
b[t]

Quorum Sensing Model

Reference: Dockery JD, Keener JP. A mathematical model for quorum sensing in Pseudomonas aeruginosa. Bull Math Biol. 2001 Jan;63(1):95-116. doi: 10.1006/bulm.2000.0205. PMID: 11146885.

Quorum sensing is a process by which bacteria communicate with each other to coordinate behavior based on their population density. They do this through the production, release, and sensing of signaling molecules called autoinducers. Now, let's break down the components in the given description related to modeling quorum sensing:

A (Autoinducer)	3 – oxo – C12 – HSL : a specific type of signaling molecule, that can be detected by other bacteria to help them gauge the population density.
R (LasR protein)	When the autoinducer (3 – oxo – C12 – HS) binds to LasR, it forms a complex that can then activate the expression of certain genes.
P (complex of A and R)	This complex is crucial for the transmission of the quorum sensing signal within the bacterial community.
ρ (cell density)	The ratio of cell volume to the space volume.

Formulas:

$$\frac{dR}{dt} = V_R \frac{P}{K_R + P} - k_R R + R_0, \tag{20}$$

$$\frac{dA}{dt} = V_A \frac{P}{K_A + P} + A_0 - d(\rho) A, \tag{21}$$

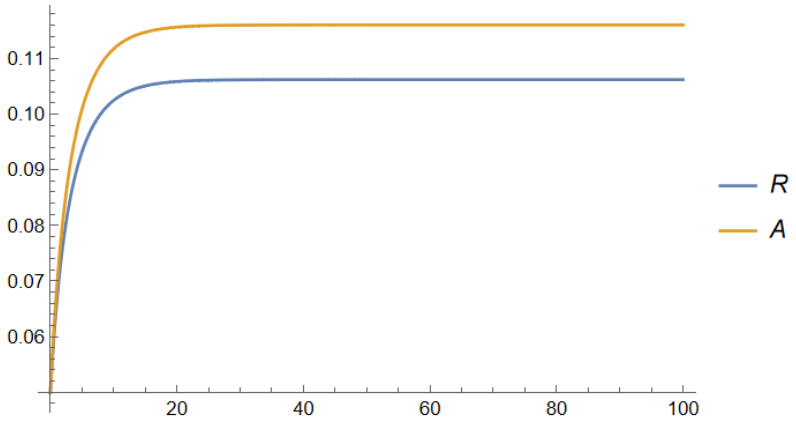
where $P = \frac{k_{RA} R A}{k_P}$ and $d(\rho) = k_A + \frac{\delta}{\rho}\left(\frac{k_E(1-\rho)}{\delta + k_E(1-\rho)}\right)$.

Originally, this system aims to regulate expression of the elastase LasB and was therefore named the las system. The two enzymes, LasB elastase and LasA elastase, are responsible for elastolytic activity which destroys elastin-containing human lung tissue and causes pulmonary hemorrhages associated with P. aeruginosa infections. The las system is composed of lasI, the autoinducer synthase gene responsible for synthesis of the autoinducer 3-oxo-C12-HSL, and the lasR gene that codes for transcriptional activator protein. The LasR/3-oxo-C12-HSL dimer, which is the activated form of LasR, activates a variety of genes, but preferentially promotes lasI activity. The las system is positively controlled by both GacA and Vfr, which are needed for transcription of lasR. The transcription of lasI is also repressed by the inhibitor RsaL. The upper equations describes the kinetics of this system.

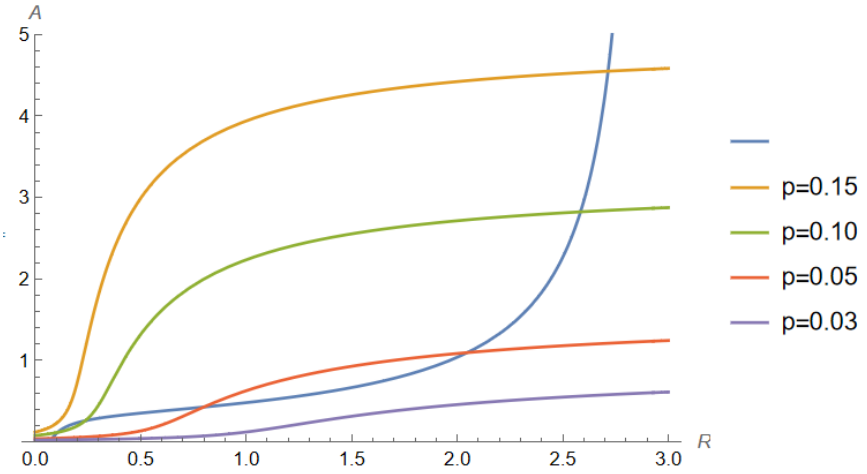
Next, use the default parameters to solve the above system of equations, which results in the

following picture:

parameters	defualt values
Vr	2
Va	2
Kr	1
Ka	1
kr	0.7
ke	0.1
ka	0.02
kra	1
kp	1
R0	0.05
A0	0.05
p	0.1
δ	0.2



We want to check how many equilibrium points are in the ode system. The picture below shows the stable line for this system, where the blue line indicates the stable line for equation (20), and the others indicate the stable line for equation (21) corresponds to different ρ. For example, the yellow line has only one common point with the blue one, which means that when ρ=0.15, there is only one equilibrium point in the ode system. There is only one equilibrium point when ρ is small. As ρ increases, the number of equilibrium points gradually becomes two, three, two, and eventually one.



Our purpose is to simulate the impact of the insertion of the two genes aiiA and ytnP on quorum sensing. The function of these two genes is to degrade AHLs, corresponds to the variable A mentioned above. Inserting these two genes is equivalent to increasing kA.

Purpose of our model: Taking kA as a variable, try to deduce the relationship between the ρ and kA to reach the critical condition of a single equilibrium point.

Model Construction:

With the equilibrium condition, we have $\frac{dR}{dt} = 0$, $\frac{dA}{dt} = 0$. Without loss of generality, assume that $\frac{k_{RA}}{k_P} = 1$. Then by (20), (21), we have

$$\frac{V_R R A}{K_R + R A} - k_R R + R_0 = 0;$$

$$\frac{V_A R A}{K_A + R A} - k_A A - \frac{\delta k_E (1-\rho)}{\rho\delta + \rho k_E (1-\rho)} A + A_0 = 0.$$

By elementary transformations, obtain:

$$-k_R A R^2 + (V_R A + R_0 A - K_R k_R) R + R_0 K_R = 0;$$

$$R = \frac{-A_0 K_A + K_A d(\rho) A}{V_A A + A A_0 - A^2 d(\rho)}. \text{ where } d(\rho) = k_A + \frac{\delta k_E (1-\rho)}{\rho\delta + \rho k_E (1-\rho)}.$$

Substitute the second into the first equation, we get a 3 order polynomial of A as follow:

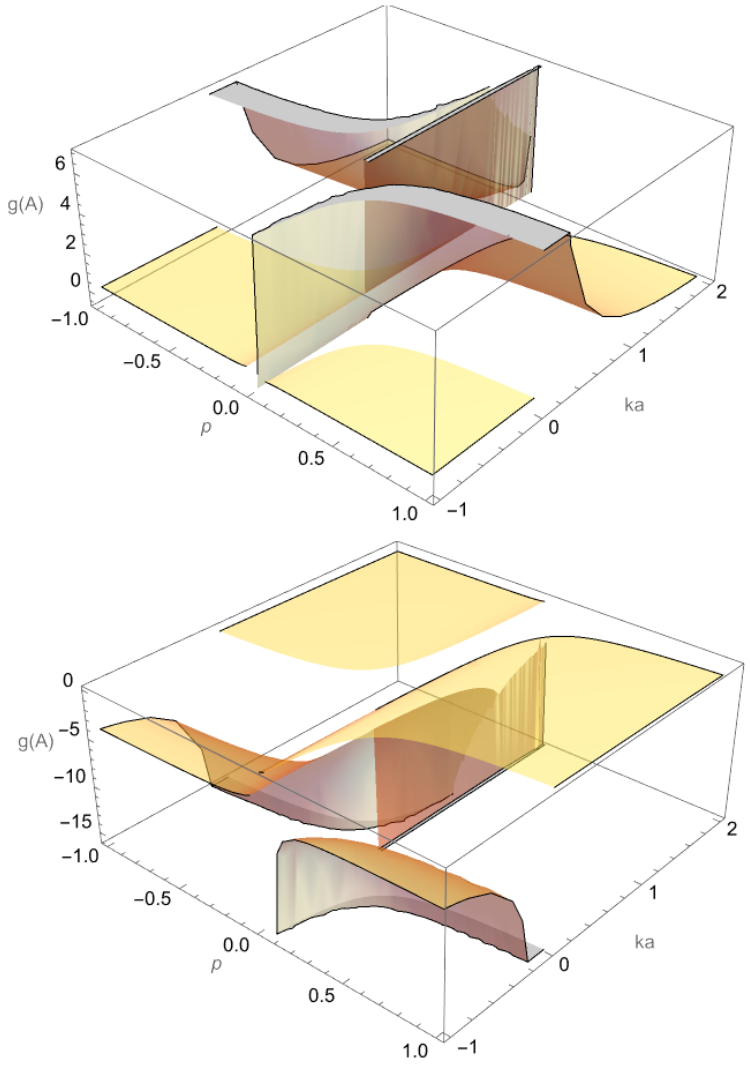
$$g(A) := \left(-d(\rho)^2 (V_R K_A + R_0 K_A + R_0 K_R)\right) A^3 + \left(d(\rho)^2 \left(-k_R K_A^2 + K_R K_A k_R\right) + d(\rho) (V_A V_R K_A + V_A R_0 K_A + A_0 V_R K_A + A_0 R_0 K_A + V_R K_A A_0 - 2 R_0 K_R V_A - 2 R_0 A_0 K_R)\right) A^2 + \left(d(\rho) \left(2 k_R A_0 K_A^2 - V_A K_R k_R - A_0 k_R K_R K_A - K_R k_R A_0 K_A\right) - V_A V_R K_A A_0 - V_R K_A A_0^2 + R_0 K_R V_A^2 + A_0^2 R_0 K_R + 2 V_A A_0 R_0 K_R\right) A + \left(-k_R A_0^2 K_A^2 + V_A K_R k_R A_0 K_A + k_R A_0^2 K_R K_A\right) = 0.$$

To find the relationship between the critical ρ and kA at a single equilibrium point, we are looking for the critical condition of only one real solution of g(A) greater than 0.

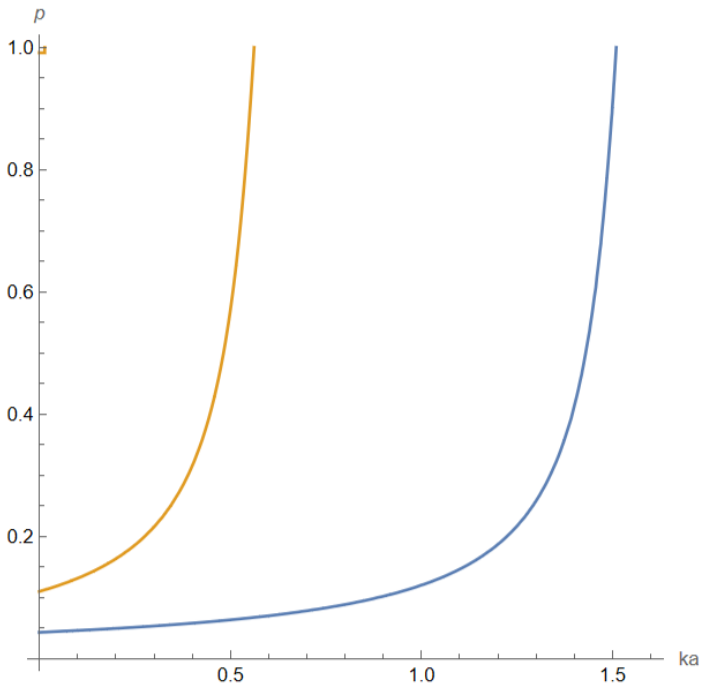
Solution:

As g(A) is a cubic polynomial of A, according to the characteristics of the cubic equation, it is a

function that is symmetric about the center of a certain point. Since the constant before the highest order is negative, this equation is negative infinity at positive infinity point and positive infinity at negative infinity point. What's more, g (0)=0.07>0. Also, the abscissa of the center point of the function is greater than zero. Let $g'(A) := a x^2 + b x + c$, then the abscissa of local maximum and minimum of g(a) are $\frac{-b+\left(b^2-4\,ac\right)^{0.5}}{2\,a}$, $\frac{-b-\left(b^2-4\,ac\right)^{0.5}}{2\,a}$; so in this way the critical condition is $g\left(\frac{-b+\left(b^2-4\,ac\right)^{0.5}}{2\,a}\right)=0$, or $g\left(\frac{-b-\left(b^2-4\,ac\right)^{0.5}}{2\,a}\right)=0$. To solve these two critical conditions, we plot $g\left(\frac{-b+\left(b^2-4\,ac\right)^{0.5}}{2\,a}\right)=0$ and $g\left(\frac{-b-\left(b^2-4\,ac\right)^{0.5}}{2\,a}\right)=0$ with respect to ρ and ka as following:



By computing the contour of 0 on the two surfaces, the relationship between the critical ρ and kA to reach a single equilibrium point is deduced as shown in the figure below:



Conclusion:

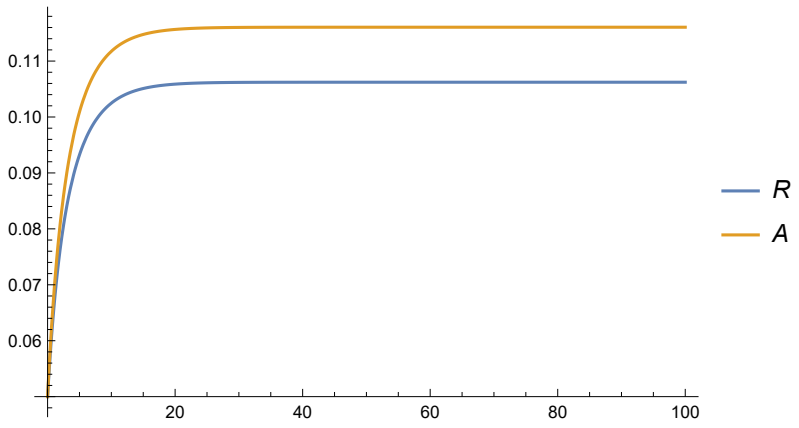
As kA increases, ρ increases significantly.

Therefore, after the insertion of aiiA and ytnP, as kA increases, ρ also increases significantly. P. aeruginosa requires higher densities to activate the quorum sensing mechanism. (The quorum sensing effect also disappears faster when the density of P. aeruginosa decreases)

Coding Material:

In[ ]:=
```
(*Set up parameters*)
Vr = 2;
Va = 2;
Kr = 1;
Ka = 1;
R0 = 0.05;
A0 = 0.05;
sigma = 0.2;
ke = 0.1;
kr = 0.7;
ka = 0.02;
kra = 1;
kp = 1;
end = 100;
p = 0.1;
(*Solve the ODE system, use the default parameter*)
solution = NDSolve[
    {R'[t] == Vr * (kra*R[t]*A[t]/kp) / (Kr + kra*R[t]*A[t]/kp) - kr * R[t] + R0,
     A'[t] == Va * (kra*R[t]*A[t]/kp) / (Ka + kra*R[t]*A[t]/kp) + A0 - A[t] * (ka + sigma/p * ke*(1 - p)/(sigma + ke*(1 - p))),
     R[0] == R0, A[0] == A0}, {A, R}, {t, 0, end}
  ];
Plot[Evaluate[{R[t], A[t]} /. solution],
 {t, 0, end}, PlotRange → All, PlotLegends → {R, A}]
```

Out[ ]=



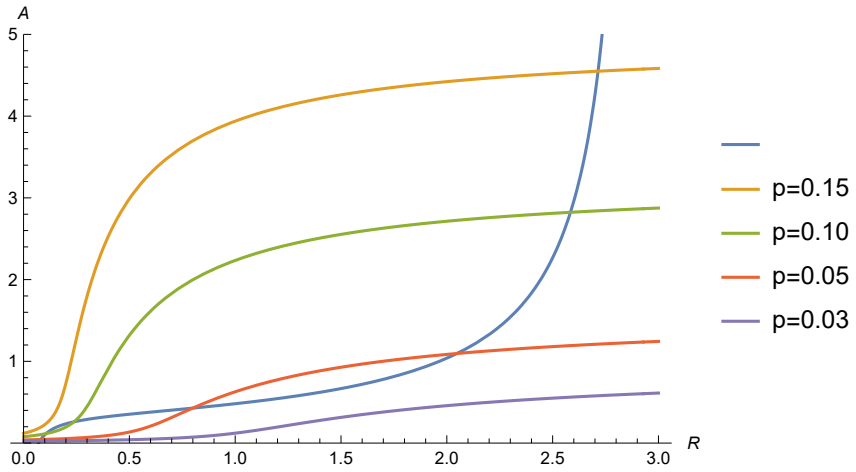
In[ ]:=
```
sol1 = Solve[
  Vr * (kra*R*A/kp) / (Kr + kra*R*A/kp) - kr * R + R0 == 0 , A
  ];

sol[p_] := Solve[Va * (kra*R*A/kp) / (Ka + kra*R*A/kp) + A0 - A * (ka + sigma/p * ke * (1 - p)/(sigma + ke * (1 - p))) == 0 , A];

Plot[{Evaluate[A /. sol1], Evaluate[A /. sol[0.15][[2]]], Evaluate[A /. sol[0.1][[2]]],
  Evaluate[A /. sol[0.05][[2]]], Evaluate[A /. sol[0.03][[2]]]},
 {R, 0, 3}, PlotRange → {0, 5}, AxesLabel → {R, A},
 PlotLegends → {"", "p=0.15", "p=0.10", "p=0.05", "p=0.03"}]
```

Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

Out[ ]=



In[ ]:=
```
(*Find the critical relation when there is
 only one real solution beyond zero in the equation*)
ClearAll["Global`*"]

(*Set up parameters*)
Vr = 2;
Va = 2;
Kr = 1;
Ka = 1;
ke = 0.1;
kr = 0.7;
delta = 0.2;
A0 = 0.05;
R0 = 0.05;
f[p_] := ka + delta ke (1 - p) / (p (delta + ke (1 - p)));
(*Obtain the equation*)
equation[A_, p_] = -f[p]^2 (Vr Ka + R0 Ka + R0 Kr) A^3 + (f[p]^2 (-kr Ka^2 + Kr Ka kr) +
    f[p] (Va Vr Ka + Va R0 Ka + A0 Vr Ka + A0 R0 Ka + Vr Ka A0 - 2 R0 Kr Va - 2 R0 Kr A0)) A^2 +
  (f[p] (2 kr A0 Ka^2 - Va Kr kr - A0 Kr Ka kr - Kr kr A0 Ka) - Va Vr Ka A0 - Vr Ka A0^2 +
    R0 Va^2 Kr + A0^2 R0 Kr + 2 Va A0 R0 Kr) A + (-kr A0^2 Ka^2 + Va Kr kr A0 Ka + kr A0^2 Kr Ka)
```

Out[ ]=
$$0.07 + A\left(0.005125 - 1.4\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right)\right) + A^2\left(0. + 4.0975\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right)\right) - 2.1\,A^3\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right)^2$$

In[ ]:=
```
∂_A equation[A, p]
```

Out[ ]=
$$0.005125 + 2A\left(0. + 4.0975\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right)\right) - 1.4\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right) - 6.3\,A^2\left(ka + \frac{0.02\,(1-p)}{(0.2+0.1\,(1-p))\,p}\right)^2$$

In[ ]:=
```
a = -6.3 (ka + 0.02 (1 - p) / ((0.2 + 0.1 (1 - p)) p))^2;
b = 2 (4.0975 (ka + 0.02 (1 - p) / ((0.2 + 0.1 (1 - p)) p)));
c = 0.005125 - 1.4 (ka + 0.02 (1 - p) / ((0.2 + 0.1 (1 - p)) p));
characteristic = b^2 - 4 a c;
centralLine = -b / (2 a)
Reduce[{centralLine < 0, ka > 0, 1 ≥ p ≥ 0}, p, Reals]
```

Out[ ]=

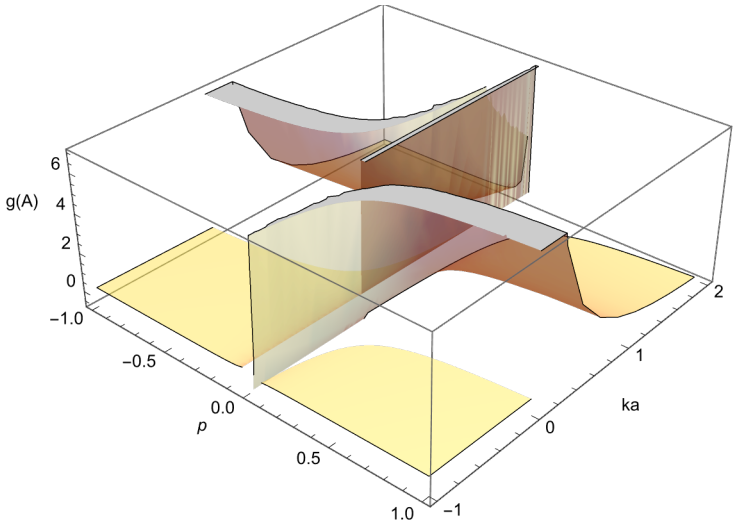
$$\frac{0.650397}{ka + \frac{0.02\ (1-p)}{(0.2+0.1\ (1-p))\ p}}$$

Reduce: Reduce was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

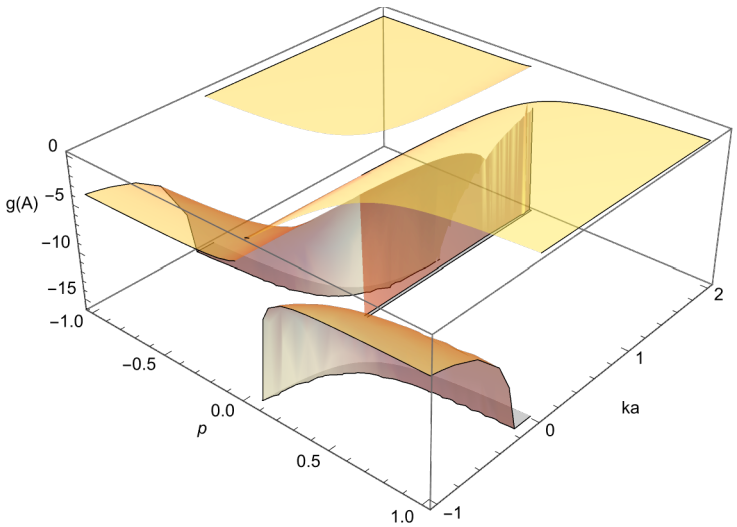
Out[ ]=
```
False
```

In[ ]:=
```
point1 = (-b - Sqrt[characteristic])/(2 a);
point2 = (-b + Sqrt[characteristic])/(2 a);
Plot3D[equation[point1, p], {p, -1, 1}, {ka, -1, 2}, AxesLabel → {p, ka, "g(A)"},
 PlotStyle → Directive[Opacity[.5], Orange, Specularity[White]], Mesh → None]
Plot3D[equation[point2, p], {p, -1, 1}, {ka, -1, 2}, AxesLabel → {p, ka, "g(A)"},
 PlotStyle → Directive[Opacity[.5], Orange, Specularity[White]], Mesh → None]
```

Out[ ]=



Out[ ]=



In[ ]:=
```
ContourPlot[{equation[point1, p] == 0, equation[point2, p] == 0},
 {ka, 0, 1.6}, {p, 0, 1}, Frame → False, Axes → True, AxesLabel → {ka, p}]
```

Out[ ]=