

DRAFT COPY

Printed April 5, 2018

DYNAMICS OF AN SIS-LIKE AUDIENCE APPLAUSE
MODEL

Antonio Miguel V. Cruz

acruz@nip.upd.edu.ph

AN UNDERGRADUATE THESIS SUBMITTED TO
NATIONAL INSTITUTE OF PHYSICS
COLLEGE OF SCIENCE
UNIVERSITY OF THE PHILIPPINES
DILIMAN, QUEZON CITY

In Partial Fulfillment of the Requirements
for the Degree of
BACHELOR OF SCIENCE IN PHYSICS
April 2018

Table of Contents

List of Figures	iv
1 Applause Dynamics	2
1.1 Nature of Applause	2
1.2 Social Dynamics	3
1.3 Background	3
1.3.1 Networks	3
1.3.2 Complex Systems	3
1.4 The audience as a complex system	4
1.5 Known Studies and Models	4
1.6 Problem Statement	5
2 Monte Carlo Compartmental Method	6
2.1 The applause as a contagion	6
2.2 Compartmental Model	6
2.2.1 States	6
2.2.2 Parameters	7
2.2.3 Functions	7
2.3 Simulation Alogrithm	9
3 Steady-state dynamics of the audience applause	16
3.1 Steady-state equation for different cases	16
3.1.1 Critical Points	18
3.2 Simulation experiments	19
3.2.1 Unstable points	21
4 Incorporating Spatial Effects	23
4.1 Real-life applause and the standing ovation effect	23
4.2 Different configurations for field of vision	23

5 Conclusions and Recommendations	24
Bibliography	25

List of Figures

2.1	The compartmental model of audience applause based on the SIS epidemic model. Shown is how the transitional probabilities and functions control how each agent switches between states.	7
2.2	Sample simulations given a fixed population $N = 100$. Simulation (a) is a typical, intuitive audience applause that has an end. Simulation (b) is non-trivial and counter-intuitive in that it seems to have no end.	15
3.1	The phase space plot of the steady-state value n_c versus \bar{a} for various β values and $b = 0.5$. Included is the trivial steady-state solution $n_c = 0$	18
3.2	Sample simulations that compare the analytical steady-state of the given parameters and the mean n_c value of the system.	19
3.3	The simulated steady-state values plotted against the analytic steady-state curves. The error bars provide the extent of deviation in simulations. The vertex is represented with the cross.	20
3.4	The vector graph of the phase space plot. The origin of the vector represents the initial n_c and corresponding \bar{a} value. The direction it points reveals whether it settles to the trivial or non-trivial solution. The color represents the probability of settling towards the directed steady-state.	22

Abstract

asdf asdf

Chapter 1

Applause Dynamics

1.1 Nature of Applause

The audience applause is one of the most ubiquitous and timeless social phenomena observed in human culture. People would normally applaud to express their approval over a social event, and may even jeer, shout, snap, and boo, on top of clapping. Such behavior is also exhibited by animals such as gorillas and chimpanzees. People collectively seem to know when to start clapping in any occasion, whether it is during or after a speech, performance, sport event, etc. People also unconsciously know whether to continue clapping or to stop. Audience members extend their applause to musicians playing an outro or to being addressed, and stop immediately when the musician proceeds with the next song. Lastly, it is historically known that people who have been hired to willingly and consciously applaud are inserted in an audience in hopes to extend the applause for a social event. These observations of an audience applause being self-organized and somehow connected give rise to the question of what the nature of applause is and how people decide when and how long to clap.

1.2 Social Dynamics

Social economics is an interdisciplinary study of the interrelationship between group and individual behavior. Individual decisions made interactively with others have been modelled formally to be able to understand the dynamics. The underlying assumption is that individuals are influenced by the choices of others. From this assumption, it can be said that feedback loops exist since the past decisions of a certain individuals may influence future decisions of others. Examples of such social phenomena are crime, teenage pregnancy, and high school dropout rates.[1][2]

1.3 Background

1.3.1 Networks

A network is a collection of points, called vertices or nodes, connected by lines, called edges. Networks are normally used to represent systems that contain individual parts that are somehow connected, such as the internet, big data, or social interactions. These are used to study the nature of individual components and their dynamics. A few properties of networks are topology and homogeneity. Network topology refers to how the network is connected. The different topologies are named after the physical shape of the network, such as a ring or a line. A key topology is a fully-connected network wherein all nodes are completely connected to each other. Network homogeneity refers to the nodes of the network. A homogeneous network has individual nodes of the same characteristics and properties while a heterogeneous network contains different nodes.

1.3.2 Complex Systems

By investigating the mechanisms that determine the topology of the networks of aforementioned systems (internet, big data, etc.) properties previously not observed when studying the components individually emerge. This has lead to a new field of

study that focuses on how relationships between components give rise to its collective behaviors and how the system interacts and forms relationships with its environment, called complex systems. Intrinsic to complex systems is that they are hard to model due to the complexity of the interacting components. Over-simplifying the complexity may lead to the failure of the model. Included in as a property of complexity is the inclination of a large system to mutate after reaching a critical point or state. Methods on studying complex systems have been established and broken down to three parts, data acquisition, modelling, and measuring complexity. Data acquisition generally tends towards statistical learning and data mining. Modelling turns to mainly two methods, cellular automata and agent-based modelling. The cellular automata is a simple mathematical model used to investigate self-organization in statistical mechanics. This model represents spatial dynamics and highlights local interactions, spatial heterogeneity, and large-scale aggregate patterns. Measuring complexity has demanded an inclusion of new metrics. These metrics include average path length, clustering coefficient, degree distribution, and spectral properties.

1.4 The audience as a complex system

Human behavior is usually studied qualitatively (under psychology or sociology) and is notoriously hard to quantify due to all the possible parameters and the difficulty in creating a controlled environment. The audience applause is an example of human behavior that is a collective of interacting agents with underlying dynamics. This allows us to treat the audience as a complex system in order to study its complexity and dynamics.

1.5 Known Studies and Models

There have been very little studies made on applause, down to the sound, its rhythm and its dynamics. One particular study treats the applause as a contagion that

propagates through the audience, allowing it to be modelled using an SIR-model. The SIR-model allows each unit in the model (for this case, each person henceforth referred to as an agent) to have 3 states, susceptible, infected, and recovered. Each agent is initially silent before the applause (susceptible). After which they start to applaud (infected). Finally stop clapping (recovered). Such a model would be appropriate if the agents no longer clap again after stopping, but cases exist where the agent may stop clapping prematurely, and then feel obliged to clap again due to the fact that the rest still continue to do so. With that, an SIS-like compartmental model with only two states (susceptible or infected) is adapted to properly account for such cases.

1.6 Problem Statement

The dynamics of the applause of an audience with N agents using an SIS epidemic model is presented and analyzed. *The same is done with an agent-based model with spatial effects.* The agents are connected via two dimensional lattice network and are initially fully connected. Later on, the agents will observe a modified, extended Moore neighborhood in order to incorporate spatial effects. The network is assumed to be homogeneous for simplicity; all agents share the same parameters for a given simulation. Agents are assumed to clap immediately after a performance, and may continue to do so depending on their given parameters. The research seeks only to find a correlation between the applause duration and the audience size, ultimately showing interdependence among the agents of the system. Why people applaud and to what are not investigated due to its unquantifiable nature.

Chapter 2

Monte Carlo Compartmental Method

2.1 The applause as a contagion

Modelling epidemics has been done in order to study the mechanisms by which diseases spread. This helps in predicting how fast and far the disease can spread in order to control and prevent future outbreaks. By treating the applause as a diseases that spreads in the audience, the same tools to model epidemics can be used to model the applause. (i think this should be in the intro)

2.2 Compartmental Model

2.2.1 States

The proposed compartmental model separates the agents into two states, silent (S) and clapping (C). The silent state replaces the susceptible state while clapping state replaces the infected state. Intuitively, agents in state S are audience members who are not clapping while agents in state C are audience members who are clapping. The number of agents in state S, given by n_s , and the number of agents in state C, given

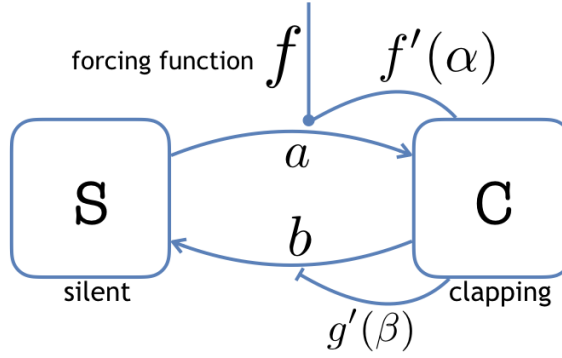


Figure 2.1: The compartmental model of audience applause based on the SIS epidemic model. Shown is how the transitional probabilities and functions control how each agent switches between states.

by n_c dictate the state of the system, given by \vec{n} where $\vec{n} \equiv (n_c, n_s)$. It is assumed that the total number of agents N is fixed, where $N = n_c + n_s$ and \vec{n} is fully specified by n_c alone.

2.2.2 Parameters

The parameters a and b are the transition parameters. They range from 0 to 1 to represent transitional probabilities. a controls the transition probability from S to C and b controls the transition probability from C to S. The respective probability transitions are given by R_1 and R_2 where

$$R_1 : S \xrightarrow{a} C \quad (2.1)$$

$$R_2 : C \xrightarrow{b} S. \quad (2.2)$$

2.2.3 Functions

An audience is initially silent until given a social cue to start clapping, such as the end of a performance. This corresponds in the model to all agents starting in state S, and then forcing the agents to undergo R_1 . The function f forces the transition R_1 for an indicated time interval. Once the f expires, the system behaves freely and its agents

may undergo various R_1 and/or R_2 transitions depending on the given parameters.

Aside from social cues, audience members may start clapping simply because others are. The peer influence may cause agents unaffected by social cues to start clapping, or those who have already stopped clapping to clap again. This creates a feedback mechanism that initiates more people to clap if majority of the agents are clapping. This corresponds in the model to the function f' . The function f' incorporates this feedback mechanism and is parametrized by α :

$$f'(\alpha) = \alpha \frac{n_c}{N-1}, \quad (2.3)$$

where α ranges from 0 to 1 for probabilistic interpretations. The probability for a spontaneous R_1 transition is directly proportional to the fraction of the population in state C and α . The denominator is set to $N-1$ because an agent cannot spontaneously influence itself; it is only influenced by the rest of the population. The function is more effective when there are more agents in state C.

Finally, audience members may contain their own bias towards a social event and may applaud longer or shorter depending on the bias. Also, the continuous applause of the majority can inhibit those who are clapping to stop clapping. This corresponds in the model to the function g' . The factor g' incorporates the inhibition as a modulation function and is parametrized by β :

$$g'(\beta) = \frac{1}{1 + \beta n_c / (N-1)} \quad (2.4)$$

where $\beta \geq 0$, representing the bias. Higher β translates to agents less likely to undergo transition R_2 . The equation is taken from the Michaelis-Menten equation, which aims to model enzyme kinetics[?].

This completes the differential equations for the reactions (2.1) and (2.2) as follows.

$$\frac{d}{dt}n_c = a(f + f' - f'f)n_s - bg'n_c \quad (2.5)$$

$$\frac{d}{dt}n_s = bg'n_c - a(f + f' - f'f)n_s \quad (2.6)$$

These equations are consistent with the assumption that the total audience size is fixed, that is $dn_c/dt = -dn_s/dt$.

2.3 Simulation Alogrithm

The compartmental SIS-like model confirmed by simulating the R_1 and R_2 process via an agent-based Monte Carlo method. For each iteration, every agent is assigned a random number that is compared to one of the transition probabilities:

$$P(R_1) = a(f + f' - f'f) \quad (2.7)$$

$$P(R_2) = bg' \quad (2.8)$$

Agents in state S undergo transition R_1 with probability shown in 2.7. Agents in state C undergo transition R_2 with probability shown in 2.8. A uniform random number, u , where $u \in [0, 1]$, is drawn from a random number generator and then compared to the corresponding probability P . If $u \leq P$, the appropriate transition is allowed to occur. Below is the code developed to simulate the audience applause.

```
from random import random
from numpy import zeros, sqrt, sum, nan_to_num
from math import isnan

#function that forces agents to go to state C
def force_func(t, end):
    if t < end:
        return 1
    else:
        return 0
```

```

#f'(alpha); feedback based on the number of ppl already in
    state C; the more ppl in state C, the more ppl will clap
def feedback_alpha(alpha, nC, population):
    return alpha * nC / (population - 1)

#g'(beta)
def feedback_beta(beta, nC, population):
    return 1 / (1 + beta * nC / (population - 1))

#spatial-dependent feedback, you can control the 'radius' of
    the reference agent
#taper refers to how many to add at the ends; radius = 0 taper
    = 1 is '90deg'
def feedback_space(alpha, system, system_row, system_column, N,
    M, radius, taper):
    applause_state = []
    for i in range(system_row):
        radius_mech = 0
        applause_state.append(system[i, system_column])
        while radius_mech != radius + taper*(system_row - i):
            radius_mech += 1
            if system_column + radius_mech < M:
                applause_state.append(system[i, system_column
                    + radius_mech])
            if system_column - radius_mech > -1:
                applause_state.append(system[i, system_column
                    - radius_mech])
    return alpha * nan_to_num(sum(applause_state) / len(
        applause_state))

```

```
def feedback_180deg(alpha, system, system_row, M): #reverted to
    simpler feedback space functions since runs took too long
    applause_state = [0]
    for i in range(system_row):
        applause_state.append(sum(system[i]))
    return alpha * nan_to_num(sum(applause_state) / (system_row * M
    ))
```

#quadratic equation

```
def quad_eq(x, y, z, sign):
    return (-y + sign * (sqrt((y ** 2) - 4 * x * z))) / 2 * x
```

#do i still need this??!?!?

```
def frange(start, stop, step):
    i = start
    while i < stop:
        yield i
        i += step
```

#creates 'network' of audience

```
def audience(N, M, C):
    agents = zeros((N, M))

    for i in range(N):
        for j in range(M):
            if sum(agents) == C:
                break
        else:
```

```

        agents[i][j]=1

    return agents

#the actual simulator
def app_sim(aStoC, bCtoS, alpha, beta, N, M, C, t, t_1):
    population = N * M
    AGENT = audience(N, M, C)
    graph = []

    for k in range(t):
        nC = sum(AGENT) #number of people clapping
        graph.append(nC)
        for i in range(N):
            for j in range(M):
                if AGENT[i,j] == 0:
                    if random() <= aStoC * (1 - (1-force_func
                        (k, t_1)) * (1 - feedback_alpha(alpha,
                            nC, population))):
                        AGENT[i,j] += 1
                else:
                    if random() <= bCtoS * feedback_beta(beta
                        , nC, population):
                        AGENT[i,j] -= 1

    return graph

#sim with spatial dependence specific to 180 deg
def sim_space(aStoC, bCtoS, alpha, beta, N, M, C, t, t_1):
    population = N * M

```



```

AGENT = audience(N, M, C)
graph = []
zeroCount = 0

for k in range(t):
    nC = sum(AGENT) #number of people clapping
    if nC == 0:
        zeroCount += 1
    graph.append(nC)
    for i in range(N):
        for j in range(M):
            if AGENT[i, j] == 0:
                if random() <= aStoC * (1 - (1-force_func
                    (k, t_1)) * (1 - feedback_180deg(alpha
                        ,AGENT,i , M))):
                    AGENT[i, j] += 1
            else:
                if random() <= bCtoS * feedback_beta(beta
                    , nC, population):
                    AGENT[i, j] -= 1
    if zeroCount == 6:
        break
    return graph
return graph

#graphs theoretical steady_state based on parameters
def steady_nC(aStoC, bCtoS, alpha, beta, population, sign):
    if beta == 0:
        if alpha == 0:

```

```

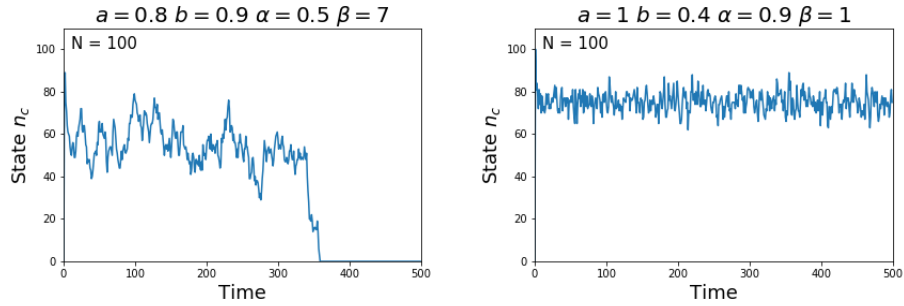
        return (aStoC * population) / (aStoC+bCtoS)
    else:
        if sign == 1:
            return population - (bCtoS * (population - 1)
                                ) / (aStoC * alpha)
        else:
            return 0
    else:
        if alpha == 0:
            return 0
        else:
            if sign == 0:
                return 0
            else:
                if isnan(quad_eq(1, ((population*aStoC*alpha*
                    beta)-(population*aStoC*alpha)+(aStoC*
                    alpha))/(-aStoC*alpha*beta), (((population
                    **2)*aStoC*alpha) - (population*aStoC*
                    alpha) - ((population**2)*bCtoS) + (2*
                    population*bCtoS) - (bCtoS))/(-aStoC*alpha
                    *beta), sign)) == True:
                    return 0
            else:

```

```

return quad_eq(1, ((population*aStoC*
alpha*beta)-(population*aStoC*alpha)+(
aStoC*alpha))/(-aStoC*alpha*beta), (((
population**2)*aStoC*alpha) - (
population*aStoC*alpha) - ((population
**2)*bCtoS) + (2*population*bCtoS) - (
bCtoS))/(-aStoC*alpha*beta), sign)

```



(a) A simulation with finite applause time. (b) A simulation with a seemingly infinite and steady applause time.

Figure 2.2: Sample simulations given a fixed population $N = 100$. Simulation (a) is a typical, intuitive audience applause that has an end. Simulation (b) is non-trivial and counter-intuitive in that it seems to have no end.

The simulator outputs a graph of n_c versus time, where time is defined to be each iteration. Examples of such simulations are shown in 2.2. An emergent property of the system is that a certain set of parameters will simulate an audience applause that does not end. Even if such cases do not exist in real life, further investigation of the steady-state applause may unravel the complexity of the system.

Chapter 3

Steady-state dynamics of the audience applause

A system is in steady-state when its behavior no longer changes in time. In the context of a real audience applause, the system rapidly spikes up in the number of people applauding, fluctuates, then slowly dies down till everyone stops clapping. This setup has the trivial steady-state of 0. However, simulations have shown that there are parameters where the steady-state is not zero, corresponding to n_c people clapping indefinitely. The value can be calculated by equating either (2.5) or (2.6) to zero, plugging in the given parameters a, b, α, β , and then solving for n_c .

3.1 Steady-state equation for different cases

The steady-state conditions are when the state \vec{n} is fixed and when the forcing function expires. This is when $d\vec{n}/dt = 0$ and when $f = 0$.

The derivation is as follows

$$\frac{d}{dt}n_c = 0 \quad (3.1)$$

$$a(f + f' - f'f)n_s - bg'n_c = 0 \quad (3.2)$$

$$a(f + f' - f'f)n_s = bg'n_c \quad (3.3)$$

$$(3.4)$$

Note: $N = n_c + n_s$, $n_s = N - n_c$, and $f' = 0$.

$$af'(N - n_c) = bg'n_c \quad (3.5)$$

Plugging in (2.3) and (2.4) to (3.5):

$$a(\alpha \frac{n_c}{N-1})(N - n_c) = b(\frac{1}{1 + \frac{\beta n_c}{N-1}})(n_c) \quad (3.6)$$

$$\frac{a\alpha(N - n_c)}{N-1}n_c = b\frac{N-1}{N-1 + \beta n_c}n_c \quad (3.7)$$

$$a\alpha(N - n_c) = b\frac{(N-1)^2}{N-1 + \beta n_c} \quad (3.8)$$

$$a\alpha(N - n_c)(N - 1 + \beta n_c) = b(N - 1)^2 \quad (3.9)$$

Once these conditions are achieved, (2.5) and (2.6) are simplified to the steady-state equation for n_c :

$$\bar{a}(N - n_c)(N - 1 + \beta n_c) = b(N - 1)^2 \quad (3.10)$$

where $\bar{a} \equiv a\alpha$. The values of steady state n_c may be solved for any given \bar{a} , b , and β . Parametrizing (3.10) to time gives the phases space plot for the given parameters b and β , shown in 3.1. Though the trivial steady-state solution is not included in (3.10), it has been included in the phase space. Values below 0 are extraneous since n_c refers to the number of people clapping, which cannot be negative.

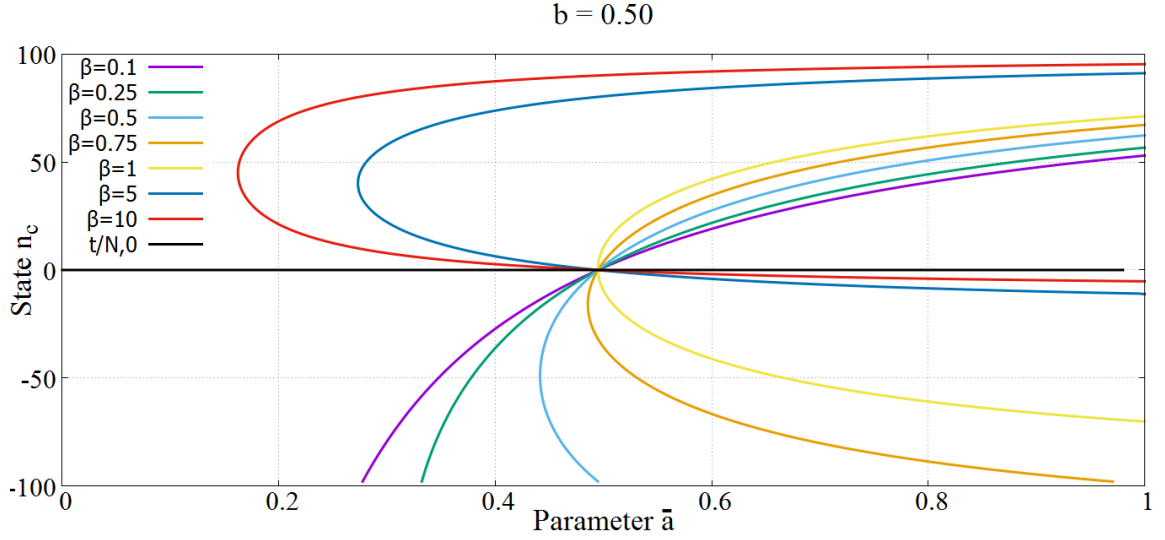


Figure 3.1: The phase space plot of the steady-state value n_c versus \bar{a} for various β values and $b = 0.5$. Included is the trivial steady-state solution $n_c = 0$

3.1.1 Critical Points

From observing all phase space plots of various b values, all β curves intersect, along with the trivial solution, at a specific point. Setting $n_c = 0$ in (3.10) gives us the critical point of intersection, $\bar{a} = \bar{a}_1$:

$$\bar{a}_1 = \frac{b(N-1)}{N} \quad (3.11)$$

which for $N \gg 1$ results to the expected $\bar{a}_1 \approx b$.

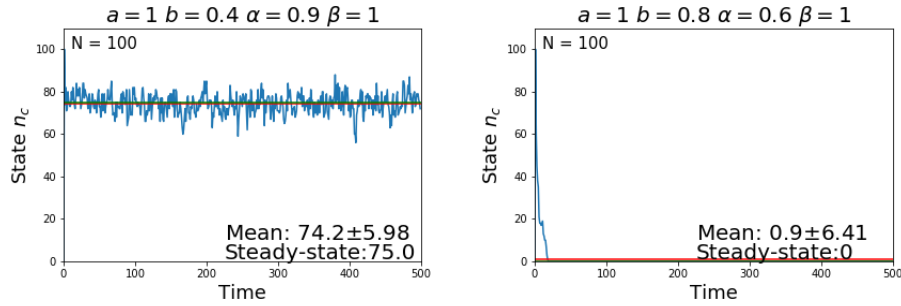
The non-trivial solution (3.10) is quadratic, which does not include the trivial solution. This results to two non-trivial n_c solutions for a given \bar{a} . For $\beta \leq 1$, the non-trivial steady-state solutions appear uniquely for $\bar{a} > \bar{a}_1$. The second n_c value is negative and therefore, extraneous. However, when $\beta > 1$, two non-trivial solutions appear between a new critical value of $\bar{a} = \bar{a}_2$, corresponding to the vertex of (3.10), and \bar{a}_1 . For $\bar{a} > \bar{a}_1$, the non-trivial steady-state solution appears uniquely once again, similar to $\beta \leq 1$. The \bar{a}_2 is given by

$$\bar{a}_2 = \frac{b(N-1)^2}{(N-n_c^*)(N-1+\beta n_c^*)} \quad (3.12)$$

where $n_c^* \equiv [1 + (\beta - 1)N]/2\beta$. This non-trivial solution however, is unstable upon substitution with \ddot{n} resulting to $\ddot{n} < 0$. Thus, the middle branch in the range $\bar{a} \in (\bar{a}_2, \bar{a}_1)$ is an unstable steady-state.

3.2 Simulation experiments

Analytical results are confirmed by simulation. The number of iterations, t , is set to be much greater than the duration of the forcing function, τ .



(a) A simulation with a non-trivial steady state of 75. The average n_c is steady state of 75. The average n_c close to the analytic value. (b) A simulation with a trivial steady state of 0. The average n_c value is approximately 0.

Figure 3.2: Sample simulations that compare the analytical steady-state of the given parameters and the mean n_c value of the system.

Figure 3.4 shows sample simulations along with the analytical steady-state and the average of the system. The analytical steady-state is plotted in green while the mean is plotted in red. As observed, the simulation closely resembles the analytical value. The system settles to its steady-state very early in the simulation. It can be safely assumed that the n_c values at $t \gg \tau$ are equivalent to n_c for $t \rightarrow \infty$. 10 sample runs were performed with different initial random number generator states and the final n_c value for each iteration were recorded. The mean and standard deviation of the simulated steady-state values are then computed for each parameter space (\bar{a}, b, β) .

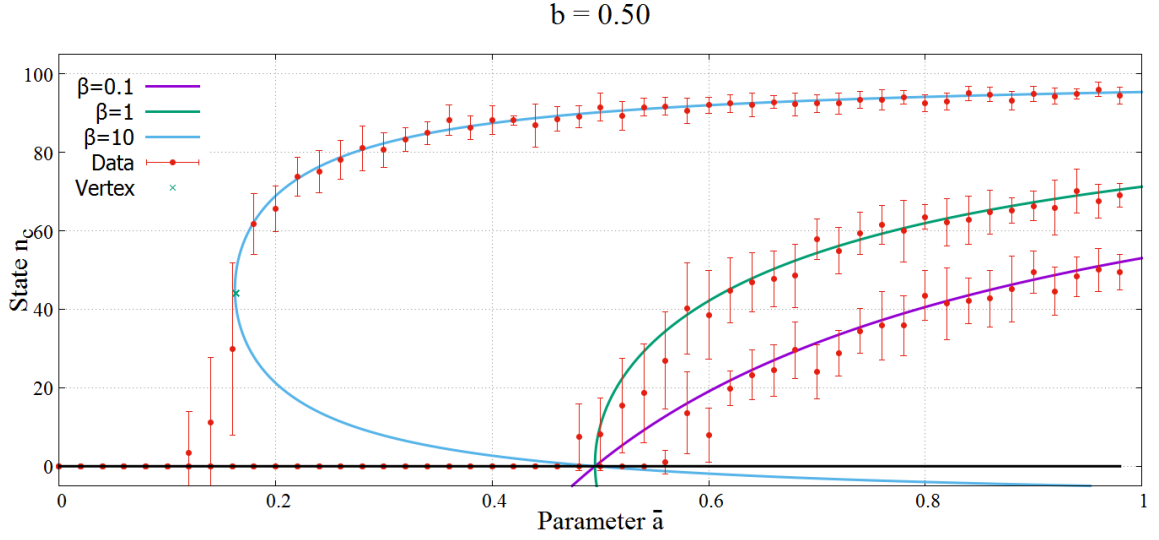


Figure 3.3: The simulated steady-state values plotted against the analytic steady-state curves. The error bars provide the extent of deviation in simulations. The vertex is represented with the cross.

The data points were then plotted with the corresponding parametrized differential curve shown in Fig. ??(c).

For $\beta \leq 1$, the simulation is consistent with the trivial steady state solution until it reaches the critical point $\bar{a} = b$, after which follows non-trivial, non-extraneous solution consistent with (3.10). For $\beta > 1$, the simulation follows the trivial steady-state and then breaks away and approaches the vertex of (3.10) at \bar{a}_2 , after which, continues to follow the upper branch of (3.10). The analytical solutions fail to predict the bifurcation from the trivial solution to the vertex of the curve. Also, the significance of the lower branch for $\beta > 1$ curves is unknown and warrants further investigation. The vertex is said to be unstable due to the fact that it unreliably and unpredictably settles to either trivial or non-trivial solution. The lower branch may share this instability.

3.2.1 Unstable points

For the previous simulation experiments, all agents start at state S and then forced to transition to state C. To investigate the properties of the lower branch on the non-trivial solution, we vary the number of agents that start the simulation in state C and remove the forcing function. This allows us to determine where the system will settle given the specific starting n_c value, more particularly for n_c values near the lower branch of the non-trivial solution.

Shown in figure ?? are the new phase space graphs that include arrows that point to where that nc value for the given \bar{a} settles. The color of the arrows represents the probability of going towards that direction. Intuitively for $\beta \leq 1$, all nc values point towards the trivial solution for all \bar{a} values less than \bar{a}_1 . All nc with \bar{a} values greater than \bar{a}_1 point towards the non-trivial solution. For $\beta > 1$, what differs is the behavior at the vertex and the lower branch of the non-trivial solution. nc values above the vertex have an estimated 50% to either settle at the vertex or zero, while nc values below the vertex absolutely settle to zero. Points slightly above the lower branch tend more towards settling to the upper branch but still have a chance to settle at 0. Likewise for points below the lower branch. Points within the lower branch have a 50% of settling to either trivial or non trivial solution.

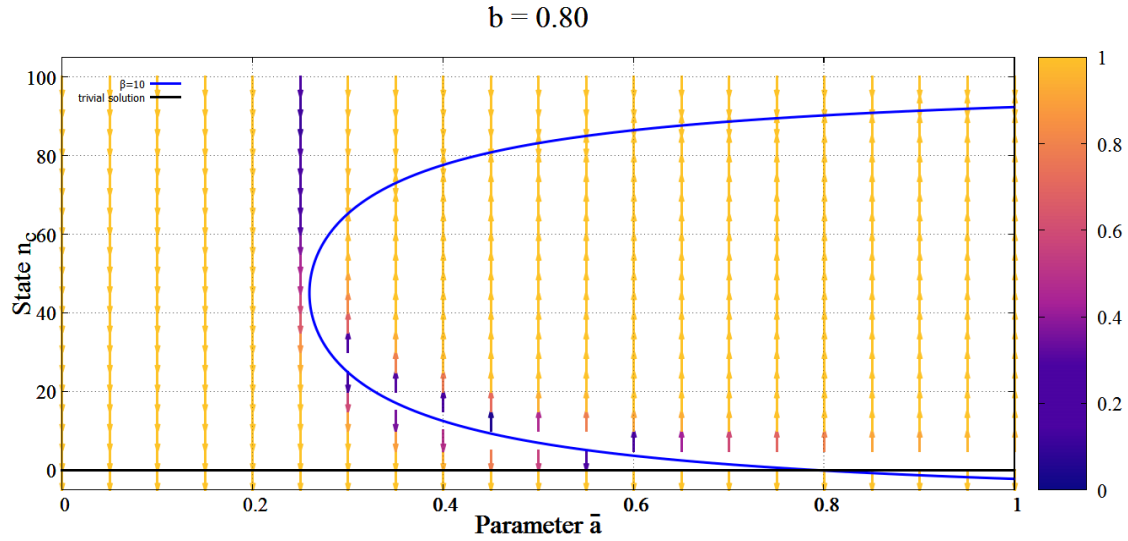
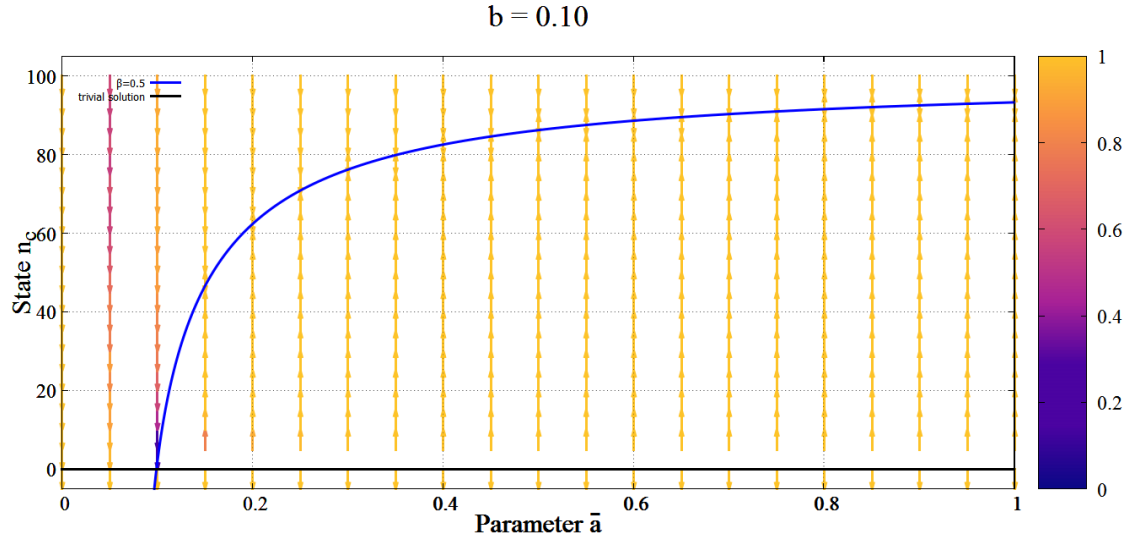


Figure 3.4: The vector graph of the phase space plot. The origin of the vector represents the initial n_c and corresponding \bar{a} value. The direction it points reveals whether it settles to the trivial or non-trivial solution. The color represents the probability of settling towards the directed steady-state.

Chapter 4

Incorporating Spatial Effects

4.1 Real-life applause and the standing ovation effect

asdfasdfasdf

4.2 Different configurations for field of vision

asdfasdfasdf

Chapter 5

Conclusions and Recommendations

In this thesis, we have reviewed the definition of the revival time associated with the wave packet revival phenomena. We have obtained approximate analytical expressions for the classical time scale T_{cl} and quantum revival time scale T_{rev} for the single- and double- δ -perturbed infinite square well system.

Bibliography

- [1] H. Peyton Young S.N. Durlauf. *Social Dynamics (Economic Learning and Social Evolution)*, volume 1. Academic Press, 2001.
- [2] B.V. Krauth. Simulation-based estimation of peer effects. *Journal of Econometrics*, 133(0304-4076):243–271, 2006.
- [3] J. Callaway. *Quantum theory of the solid state. Student edition*, volume 1 of *Quantum Theory of the Solid State. Student Edition*. Academic Press, 1976.
- [4] A.K. Das G.A. Vugalter and V.A. Sorokin. Revivals in an infinite square well in the presence of a δ well. *Phys. Rev. A*, 66(012104):1–7, 2002.
- [5] D.J. Griffiths. *Introduction to quantum mechanics*. Pearson Prentice Hall, second edition, 2005.
- [6] W. Heitler and F. London. Wechselwirkung neutralere atome und homopolare bindung nach der quantenmechanik. *ZP*, 44:455–472, 1927.
- [7] M.P. Marder. *Condensed matter physics*. Wiley-Interscience. John Wiley, 2000.
- [8] R.W. Robinett. Quantum wave packet revivals. *Physics Reports*, 392:1–119, 2004.
- [9] R. Schinke S.Yu. Grebenshchikov, C. Beck and S.C. Farantos. Three-dimensional molecular wave packets: Calculation of revival times from periodic orbits. *Physics Letters A*, (243):208–214, 1998.

- [10] D.F. Styer. Quantum revivals versus classical periodicity in the infinite square well. *Am. J. Phys.*, 69:56, January 2001.
- [11] T.K. Timberlake and S. Camp. Decay of wave packet revivals in the asymmetric infinite square well. *Am. J. Phys.*, 79:607–614, 2011.