

**MODELLING WILDEBEEST FORAGING PROCESSES AND THEIR
INTERACTION WITH ZEBRA AND LION IN THE SERENGETI
ECOSYSTEM**

Linus Nyarusanda Kisoma

**A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical and Computer Sciences and Engineering of the
Nelson Mandela African Institution of Science and Technology**

Arusha, Tanzania

June, 2022

ABSTRACT

Animal movements and foraging processes for the migrating species, especially wildebeests and zebras, and the prey-predator interactions of these prey species with lions are ambiguous biological characteristics in the Serengeti ecosystem. These complex dynamics help animals to adapt and survive. To understand such dynamics, investigating factors that determine foraging efficiency and the prey-predator interaction is worth it. This dissertation presents deterministic mathematical models to examine wildebeest foraging processes and the prey-predator interaction of wildebeest, zebra, and lion populations. The first model studies the foraging processes of migrating wildebeests using the concepts of random walk and diffusive processes. The model was equipped with data collected from the Serengeti ecosystem from 18 GPS collared wildebeests and analysed in two spatial dimensions. The qualitative analysis of the model was performed, and the parameters that regulate foraging efficiency were calculated for both dry and wet seasons. Numerical simulations were performed, and the results show that directed movements can explain the great migration of wildebeests to different habitats. Wildebeests spread across different habitats to utilize the resources through diffusive trends. The mutual association between wildebeests and zebras was studied by developing the Lotka-Volterra reaction-diffusion systems. This model was further modified to form the third model that includes the predation pressure from lions. The qualitative analyses of the models were carried out in two dimensions to determine points of equilibrium and the conditions for the stability and instability of the systems. The explicit Euler method was used to discretize the models and perform numerical simulations. The stability analyses of the models showed that wildebeests and zebras population growth approached their respective carrying capacities, and the absence of one prey species does not affect the existence of the other. The advection and diffusion parameters in the model produce Turing instabilities. Furthermore, the results show that both prey species are strongly affected by drought and predation pressure, especially from lions. Therefore, advection and diffusion of wildebeests and zebras are motivated by the search for better forage availability and avoidance of predators, while the predator's movement is motivated by capturing prey.

DECLARATION

I, Linus Nyarusanda Kisoma, hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this dissertation is my original work and that it has neither been submitted nor is concurrently submitted for a degree award in any other institution.

Linus Nyarusanda Kisoma



27/07/2022

Name and Signature of the Candidate

Date

The above declaration is confirmed by:

Prof. Dmitry Kuznetsov

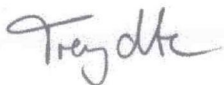


29/07/2022

Name and signature of Supervisor 1

Date

Prof. Anna Treydte



29/07/2022

Name and Signature of Supervisor

Dr. Colin Torney



29/07/2022

Name and Signature of Supervisor

Date

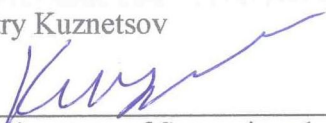
COPYRIGHT

This dissertation is copyright material protected under the Berne Convention, the Copyright Act of 1999 and other international and national enactments, on behalf, of intellectual property. It must not be copied by any means, in full or in part, except for short extracts in fair dealing; for a researcher's private study, critical scholarly review or discourse with an acknowledgement, without written permission of the Deputy Vice-Chancellor for Academic, Research and Innovation, on behalf of both the author and the Nelson Mandela The African Institute of Science and Technology.

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the Nelson Mandela African Institution of Science and Technology a dissertation titled: "*Modelling wildebeest foraging processes and their interaction with zebra and lion in the Serengeti ecosystem*" in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Mathematical and Computer Sciences and Engineering of the Nelson Mandela African Institution of Science and Technology.

Prof. Dmitry Kuznetsov

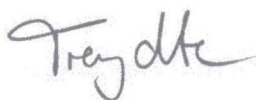


29.07.2022

Name and signature of Supervisor 1

Date

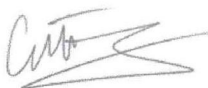
Prof. Anna Treydte



29/07/2022

Name and Signature of Supervisor

Dr. Colin Torney



29/07/2022

Name and Signature of Supervisor

Date

ACKNOWLEDGEMENTS

Through God Almighty, I received this opportunity to pursue my PhD studies at NM-AIST. It was a tiring journey with many turbulences and trails, but God stood by me all the time. Throughout the preparations of this dissertation, I received tremendous support and assistance.

My heartfelt gratitude and thanks go to my supervisors: Prof. Dmitry Kuznetsov, Prof. Anna Treydte, and Dr Colin Torney, for their tireless support, trust, guidance and counselling throughout this research study. My special thanks go to Dr. Colin Tourney and Dr. Grant C. J. Hopcraft for assisting me with some data and allowing me to visit them at the University of Glasgow, UK. May God Bless You all.

I would like to thank the NM-AIST for awarding me the scholarship through DAAD grants, and this study would not have been possible without their support. Similarly, I would like to thank the National Geographic Society (NGS) through its early career grant for supporting my research activities and the visitation to the University of Glasgow, UK.

I extend my gratitude to all the CoCSE staff for their constant support, patience, and coordination throughout my studies.

Thanks to my fellow PhD classmates, Mr. Anold Mwapinga and the late Mr. Musa Antidius Mjankwi, for their friendship, support and cordial working environment.

In addition, I would like to thank my family especially my beloved wife, Ms. Joyce Joseph, for her wise counsel and sympathetic ear. Finally, my special thanks go to my mother, Ms. Penina and my Children Kisoma, Cletus Kisoma, Clare Kisoma, Penina Kisoma and Anna Kisoma who had patience and endurance during my study.

DEDICATION

To the memory of my late Father Mr. Joseph Kisoma

TABLE OF CONTENTS

ABSTRACT	i
DECLARATION	ii
COPYRIGHT	iii
CERTIFICATION	iv
ACKNOWLEDGEMENTS	v
DEDICATION	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES.....	xi
LIST OF APPENDICES	xiii
LIST OF ABBREVIATIONS AND SYMBOLS	xiv
CHAPTER ONE	1
INTRODUCTION.....	1
1.1 Background of the Problem.....	1
1.1. 1 Selection of Animals for this Study	2
1.2 Statement of the Problem	5
1.3 Rationale of the Study	6
1.4 Objectives of the Study	6
1.4. 1 General Objective.....	6
1.4. 2 Specific Objectives.....	7
1.5 Research Questions	7
1.6 Significance of the Study	7
1.6.1 Wildlife Management and Conservationists	8
1.6.2 Contributions of the Study	8
1.6.3 Future Researchers	9

1.7	Delineation of the Study	9
CHAPTER TWO.....		11
LITERATURE REVIEW.....		11
2.1	Introduction	11
2.2	Related Work.....	11
2.3	Random Walk and Advection-Diffusion Equation	12
2.4	Diffusive Prey Predator Mathematical Models	14
2.5	Chapter Summary	16
CHAPTER THREE.....		17
MATERIALS AND METHODS		17
3.1	Introduction	17
3.2	The Study Site	17
3.3	Data Sources	18
3.4	The Concept of Random Walk and Diffusive Processes	18
3.4.1	The Fokker-Planck Equation (2D Random Walk).....	18
3.5	Mutualism Reaction-Diffusion Models for Wildebeests and Zebras	21
3.5.1	Assumptions of the Model	21
3.5.2	The Mutualism Model with Diffusion and Migration.....	22
3.5.3	Reaction-Diffusion System	22
3.5.4	The Reaction	23
3.5.5	Local Stability Analysis	24
3.5.6	Local Stability Analysis	25
3.5.7	Local Stability of the Equilibrium Points.....	27
3.5.8	Stability Analysis of the Model with Diffusion and Migration	29
3.5.9	Numerical Methods	33
3.6	Diffusive Prey-predator Relations between Wildebeest, Zebra and Lion	34

3.6.1	The Mathematical Model	34
3.6.2	Local Stability Analysis	36
3.6.3	Existence of Equilibrium Points in the System	37
3.6.4	Local Stability of the Equilibrium Points.....	41
3.6.5	Global Stability of the Steady States.....	49
3.6.6	Dispersion Relation	56
3.6.7	Numerical Methods	61
3.7	Chapter Summary	62
CHAPTER FOUR.....		63
RESULTS AND DISCUSSION		63
4.1	Results	63
4.1.1	Foraging Processes of Wildebeest.....	63
4.1.2	Mutualism between Wildebeests and Zebras	70
4.1.3	Wildebeest, Zebra and Lion Prey-Predator Interactions	75
4.2	Discussion	80
4.2.1	Foraging Processes of Wildebeest.....	80
4.2.2	Mutualism between Wildebeests and Zebras	82
4.2.3	Wildebeest, Zebra and Lion Prey-Predator Interactions	83
CHAPTER FIVE.....		85
CONCLUSION AND RECOMMENDATIONS.....		85
5.1	Conclusion.....	85
5.2	Recommendations	85
REFERENCES.....		87
APPENDICES.....		95
RESEARCH OUTPUTS		114

LIST OF TABLES

Table 1:	Parameters of wildebeest and zebra with their sources	34
Table 2:	Parameters of the two preys and predators with their sources	61
Table 3:	Calculated hourly average diffusion and advection movement parameters from the advection-diffusion equation for each year	65
Table 4:	Dry season hourly average diffusion and advection movement parameters calculated from the advection-diffusion equation for each year	67
Table 5:	Wet season hourly average diffusion and advection movement parameters calculated from the advection-diffusion equation for each year	70
Table 6:	Population dynamics of wildebeest and zebra in absence of predation and drought	71
Table 7:	Population dynamics of wildebeest and zebra in presence predation and moderate drought.....	72
Table 8:	Population dynamics of wildebeest and zebra in the presence of extreme drought	73
Table 9:	Wildebeest, zebra and lion population in absence of extreme drought.....	79
Table 10:	Wildebeest, zebra and lion population in presence of extreme drought	80

LIST OF FIGURES

Figure 1:	Wildebeest and Zebras in the Serengeti National Park during the great migration (www. safari. nathab. com/ blog/ spectacular -photos- from –tanzanias –great -migration/)	2
Figure 2:	Wildebeest and Zebra abundances from 1961 to 2010 (TAWIRI, 2010)	3
Figure 3:	The lion pride in the Serengeti National Park (https:// www. gettyimages. com/ photos/ serengeti-lion)	4
Figure 4:	Location of the Serengeti ecosystem in Northern Tanzania (Sagamiko <i>et al.</i> , 2015)	17
Figure 5:	Trajectories around the equilibrium points	29
Figure 6:	Individual random walk trajectories of 5 wildebeest on a 2D lattice (Units of the grid are in UTM)	64
Figure 7:	Individual random walk trajectories of 18 wildebeest on a 2D lattice (Units of the grid are in UTM)	64
Figure 8:	Individual random walk trajectories for 5 wildebeest on a 2D plane in the dry season (the units of the grid are in UTM)	66
Figure 9:	Individual random walk trajectories for 18 wildebeest on a 2D lattice in the dry season (the units of the grid are in UTM)	67
Figure 10:	Individual random walk trajectories for 5 wildebeest on a 2D lattice in the wet season (the grid units are in UTM)	68
Figure 11:	Individual random walk trajectories for 18 wildebeest on a 2D lattice in the wet season (the grid units are in UTM)	69
Figure 12:	Wildebeest and zebra population growth with time	70
Figure 13:	Effect of predation on wildebeest and zebra population growth with time	71
Figure 14:	Effect of intensive drought on wildebeest and zebra populations	72
Figure 15:	Wildebeest population distribution and average position of walkers on the xy – plane. The orange bars are the actual (empirical data and the red line is the prediction based on the advection-diffusion equation)	73
Figure 16:	Wildebeest population evolution on the xy – plane	74

Figure 17:	Zebra population distribution and average position of walkers on the xy – plane	74
Figure 18:	Zebra population evolution on the xy – plane	75
Figure 19:	Wildebeest population distribution and average position of walkers on the xy – plane.....	76
Figure 20:	Wildebeest population evolution on the xy – plane	76
Figure 21:	Zebra population distribution and average position of walkers on the xy -plane	77
Figure 22:	Zebra population evolution on the xy – plane	77
Figure 23:	Lion population distribution and average position of walkers on the xy – plane. The violet bars are the actual (empirical) data and the red line is the prediction based on the advection-diffusion equation	78
Figure 24:	Lion population evolution on the xy – plane	78
Figure 25:	Populations growth of interacting species with time	79
Figure 26:	Populations growth of interacting species in the presence of drought with time	80

LIST OF APPENDICES

Appendix 1:	Analysis for full wet season	95
Appendix 2:	Mutual Relationship between Wildebeests and Zebras during Migration	101
Appendix 3:	Analysis of Prey-Predator Relations of the three Species	107

LIST OF ABBREVIATIONS AND SYMBOLS

τ	Time Step
δ	Distance
r_w and r_z	Growth
K_w	Carrying Capacities
∇^2	Is the Laplacian Operator.
∇	Is the Gradient Operator
k_n	Stands for Wave Number,
ω_n	Is the Growth Rate of Pertubations
δ_{z_0}	Perturbation Amplitudes
P	Predator
$J(E_0)$	Jacobian matrix
E_2	Eigenvalues
2D	Two dimensional
BRW	Biased Random Walks
GPS	Global Positioning System
i.e.	Id est
IBM	Individual-Based model
NDVI	Normalized difference vegetation index
NM-AIST	Nelson Mandela African Institution of Science and Technology
PDE	Partial differential equations
PDF	Probability density function
SRW	Simple Random walk
TAWIRI	Tanzania Wildlife Research Institute
UK	United Kingdoms
UNESCO	United Nations Educational, Scientific and cultural organization
UTM	Universal Transverse Mercator

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

The increase of human interference in the environment has induced major changes in the functioning of different ecosystems leading to the extinction of many species including wildlife (Hagen *et al.*, 2012). Although some species move to different ecosystems in search of forage refuge, this mobility may not occur in the future due to increasing global population growth that contributes to the destruction, modification and fragmentation of wildlife habitats (Taylor-Brown *et al.*, 2019). Thus, there is a need to conserve wildlife (Hopcraft, 2010).

Conservation of the environment and wildlife protects nature, making it available for future generations to enjoy the natural world and its wonderful species and appreciate the importance of wildlife for humans and other species (Hagen *et al.*, 2012). However, studies on species conservation require detailed knowledge of their behaviour, how they acquire food and other resources for their survival, and how they protect themselves against dangers. It is also essential to understand why some animals migrate from one place to another looking for forage refuge (Gueron *et al.*, 1996). For instance, the great annual migration of wildebeests, zebras and predators, and other species in the Serengeti ecosystem.

The Serengeti ecosystem in northern Tanzania is one of the world heritage sites due to the great migration of wildebeest, zebras and other species (Hopcraft 2010). The great migration is caused by the search for green and nutritious pasture between the Serengeti ecosystem in Tanzania and Masai Mara game reserve in Kenya (Tourney *et al.*, 2018). In addition, this ecosystem is home to different animal species including the big five: the Lion, the Elephant, the Black Rhinoceros, the Leopard and the African Buffalo (Sagamiko *et al.*, 2015).

The Serengeti ecosystem is the best known and most treasured ecosystem (Sagamiko *et al.*, 2015). Wildlife conservation in the Serengeti ecosystem is important because it helps maintain a balanced ecological system and the environment as each organism has its place on the food chain (Kideghesho, 2010). Further, wild animals attract tourism (Kideghesho, 2010). The great annual migration of wildebeests, zebras and predators, and other species in the Serengeti ecosystem is so unique that it has been attracting tourists from different parts of the world who come to enjoy the nature of Serengeti (Mduma *et al.*, 1996). This has economic importance

because money is gained from tourism and hunting wild animals (Kideghesho, 2010). In this regard, we need to protect wildlife for the sustainability of the Serengeti ecosystem. This can be achieved by understanding different animal groups' dynamics, especially wildebeests and zebras that form a larger part of migrating animals in the Serengeti ecosystem.

The animal group dynamics in the Serengeti ecosystem have triggered much research interest for more than half a century. Biologists have studied such dynamics through mathematical models and statistical analyses. Besides, movement patterns of wildebeests and zebras have mainly been explained by a changing environment and predation pressure, mainly from a lion. However, only a few attempts have been made to model the foraging processes of wildebeests and how they interact with zebras and lions based on mathematical predictions. This study was motivated by the need to continue investigating the interaction dynamics of wildebeests, zebras, and lions in the Serengeti ecosystem through Mathematical Modelling.

1.1.1 Selection of Animals for this Study

The animals selected for this study were wildebeests (*Connochaetes taurinus*), zebras (*Equus burchellii*), and lions (*Panthera leo*) of the Serengeti National Park. Lion and the hyena are the main predators of grazers in Serengeti (Hopcraft, 2010), but the lion was selected because it accounts for more kills than the hyena (Hopcraft, 2010; Mduma, 1996). Also, zebras and wildebeests were chosen because the two have ecological similarities, including similar body sizes and similar spatial patterns of resource use and migration patterns (Hopcraft, 2010; Sinclair, 1977). Resource availability and predation are two major factors determining the abundance of zebra and wildebeest populations (Grange *et al.*, 2004).



Figure 1: Wildebeest and Zebras in the Serengeti National Park during the great migration ([www. isafari. nathab. com/ blog/ spectacular -photos- from- tanzanias -great-migration/](http://www.isafari.nathab.com/blog/spectacular-photos-from-tanzanias-great-migration/))

Wildebeests and zebras usually migrate together from the Serengeti National Park in northern Tanzania (May to November) to seek fresh grazing and water in the Masai Mara National Reserve in Kenya (Hopcraft, 2010; Tournay *et al.*, 2018; Holdo *et al.*, 2011). Then the animals come back again on the short grass plains of the southern Serengeti (December to May) to seek fresh grazing and calving. This migration is an annual mass movement of millions of ungulates and other species in groups (Holdo *et al.*, 2011). The great migration involves an estimated 1.3 million wildebeests, 200 000 zebras, and a multitude of gazelles, among various other hooved species (Grange *et al.*, 2004).

While the movement patterns of the migrating species have remained the same for decades, the wildebeest populations in the Serengeti ecosystem have been highly variable in size, increasing from 263 000 in 1961 to fluctuating around 1.5 million individuals in 2010 (Menard *et al.*, 2002; Mduma *et al.*, 1999; Hopcraft, 2010). Studies indicate, for example, that in the early 1960s, the numbers of zebras and wildebeests in the Serengeti were almost equal. However, time due to a rinderpest outbreak in the year 1960, which greatly affected wildebeests and buffalos (*Syncerus caffer*), the wildebeests' populations decreased while the zebra population in the Serengeti has significantly remained unchanged (Hopcraft, 2010; Duncan *et al.*, 1990; Menard *et al.*, 2002).

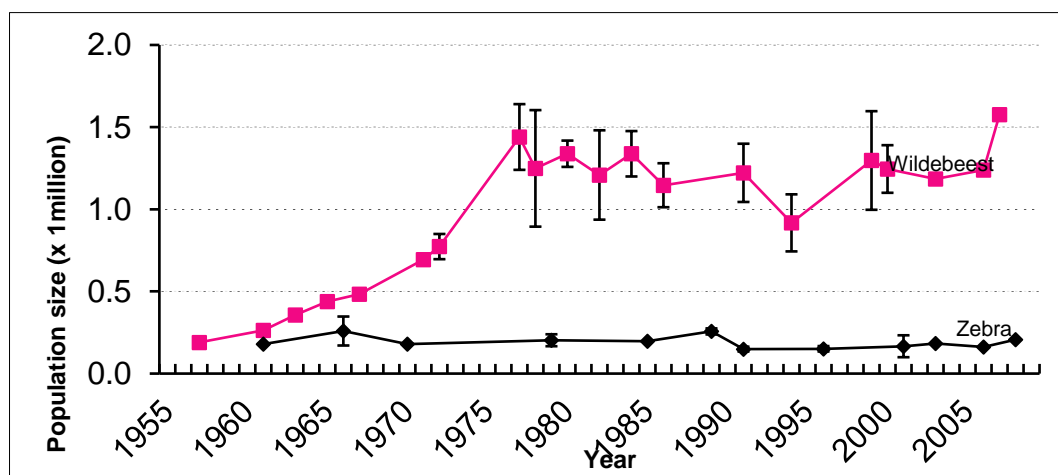


Figure 2: Wildebeest and Zebra abundances from 1961 to 2010 (Tanzania Wildlife Research Institute [TAWIRI], 2010)

In addition to the movement for food, wildebeests and zebras, like many other ungulate species, select their habitat use based on the presence of predators (Hopcraft, 2010). The main predators of wildebeests and zebras are lion (*Panthera leo*) and hyena (*Crocuta crocuta*) (Mduma *et al.*, 1999). Lions of the Serengeti are classified into two basic types: residents, which remain within a limited area for most of their lives, and the nomads, which wander widely, often following the

movements of the migratory herds (Schaller, 1977). Despite short-term environmental perturbations that have led to numerous gradual changes in prey availability and vegetative cover, populations of Serengeti lions have remained relatively stable (Sinclair *et al.*, 2008). Lions live in prides with an average pride size of four individuals; the pride size and composition change from daily depending on their needs.



Figure 3: The lion pride in the Serengeti National Park ([https:// www. gettyimages. com/ photos/ serengeti-lion](https://www.gettyimages.com/photos/serengeti-lion))

Some studies (Grange *et al.*, 2004; Ikanda & Packer, 2008; Sagamiko *et al.*, 2015; Hopcraft, 2010; Tournay *et al.*, 2018) have shed light on the dynamics of different species such as wildebeests (*Connochaetes taurinus*), zebras (*Equus burchellii*), and lions (*Panthera leo*). However, these studies focused on factors that limit the population abundance of different migrating species in the Serengeti ecosystem. Also, they concentrated on the prey-predator interactions mostly involving a two species approach (Sagamiko *et al.*, 2015). In addition, although these studies provided detailed explanations of the reasons for migration and abundance, such as rainfall and grass, they did little to consider the factors that lead to the foraging efficiency of these species, particularly wildebeests. These studies also overlooked the role of mathematical models in explaining the interaction of migrating species, which is mutualistic, and the prey-predator interaction that involves three species in the Serengeti ecosystem. The inclusion of mathematical models in explaining the interaction between prey

and predators helps to understand how proximate and ultimate causes for these behaviours work (Taylor-Brown *et al.*, 2019). Further, these mathematical models help predict future trends and contribute to understanding migration.

Therefore, this study focused on modelling wildebeests, zebras, and lion movement dynamics. In particular, this study intended to show wildebeest foraging processes through a mathematical model as these species undergo migration and diffusion. Furthermore, the study used mathematical models to investigate the dynamics of the mutual association of wildebeests, zebras, and other ungulates as they migrate together, and their associated prey-predator interaction with lions.

1.2 Statement of the Problem

The previously proposed models in the Serengeti and other ecosystems were based on prey-predator interactions, mostly involving wildebeests and lions (Sagamiko *et al.*, 2015; Fay & Greef, 1999). For example, Fay and Greef (2006) studied the dynamics of wildebeests, zebras, and lions as a prey-predators using an ordinary differential equation model in the Kruger National Park. The model was analysed to show the dynamics of the interacting species based on historical data. However, the model was carried out in one dimension leaving out the advection and diffusion components. Others include Lee (2007), who described the effects of remained Carcass on the stability of the dynamic system; and Murthy *et al.* (2017), who used a discrete three species mathematical model to study the fishery system with two-prey and two-predators (one species acts as prey and predator simultaneously). Their discrete model had no diffusion and migration components and was analysed in one dimension. Lian *et al.* (2012) studied the pattern formation in a cross-diffusive Holling type III ratio-dependent prey-predator model. While the model was useful to explain the system dynamics induced by diffusion, it was based on one prey-one predator and had no migration component. Various one prey and one predator models with diffusion and Holling type III functional response studies have been done (Wang, 2016; Liu, 2010). However, none of these studies has included three species (two mutualistic preys and one predator) and advection and diffusion parameters explaining different dynamics.

In contrast to the previously proposed models and studies, the mathematical models explaining the dynamics of migrating species in the Serengeti ecosystem have not been given enough attention in the literature. Therefore, this study aimed to understand these dynamics. The study

used the Holling type II functional response to describe the predator's per capita feeding rate. Furthermore, in this proposed prey-predator model, the effects of migration and diffusion were analysed in two spatial dimensions.

1.3 Rationale of the Study

The increase of human interference in the environment has induced major changes in the functioning of different ecosystems leading to the extinction of many species including wildlife (Hagen *et al.*, 2012). Thus, this necessitate the need to conserve wildlife (Hopcraft, 2010). Conservation of the environment and wildlife protects nature, making it available for future generations to enjoy the natural world and its wonderful species and appreciate the importance of wildlife for humans and other species (Hagen *et al.*, 2012). It is also essential to understand why some animals migrate from one place to another looking for forage refuge (Gueron *et al.*, 1996). For instance, the great annual migration of wildebeests, zebras and predators, and other species in the Serengeti ecosystem. The Serengeti ecosystem in northern Tanzania is one of the world heritage sites due to the great migration of wildebeest, zebras and other species (Hopcraft, 2010). In addition, this ecosystem is home to different animal species including the big five: The Lion, the Elephant, the Black Rhinoceros, the Leopard and the African Buffalo (Sagamiko *et al.*, 2015). Conserving and protecting Serengeti ecosystem, has economic importance because money is gained from tourism and hunting wild animals (Kideghesho, 2010). In this regard, we need to protect wildlife for the sustainability of the Serengeti ecosystem. This can be achieved by understanding different animal groups' dynamics, especially wildebeests and zebras that form a larger part of migrating animals in the Serengeti ecosystem.

1.4 Objectives of the Study

1.4.1 General Objective

The general objective of this study was to develop deterministic mathematical models and use them to investigate and analyse wildebeest foraging processes and their interaction with zebras and lions in the Serengeti ecosystem.

1.4.2 Specific Objectives

The specific objectives of this study were to:

- (i) Formulate a deterministic mathematical model and use it to analyse and investigate the foraging processes of herds of wildebeest using the concept of random walk and diffusive processes.
- (ii) Formulate a deterministic mathematical model with diffusion and advection parameters and use it to analyse and investigate the mutual relationship between wildebeests and zebras during migration.
- (iii) Formulate a deterministic mathematical model with diffusion and advection parameters and use it to analyse and investigate the dynamics of prey-predator relations of the three species.

1.5 Research Questions

The present study had the following research questions:

- (i) What is the possibility of formulating a deterministic mathematical model and using it to analyse and explain wildebeest foraging efficiency?
- (ii) How does the mutual relationship between wildebeests and zebras during migration work?
- (iii) What is the possibility of formulating a three species deterministic mathematical model and using it to investigate and analyse the dynamics of prey-predator relations of wildebeest, zebra and lion together?

1.6 Significance of the Study

This study is partly rooted in animal behaviour. Therefore, the findings of this study could increase awareness of wildlife behaviour among ecologists and conservationists. Furthermore, understanding animal dynamics such as animal movement and foraging patterns, predator-prey relations and social behaviour of these species would be useful to wildlife management and conservationists and future researchers as follows.

1.6.1 Wildlife Management and Conservationists

This study increases awareness of how environmental variations affect the survival of different species. This is because the protection of animals is linked to the environmental conditions (such as rainfall), reproduction rates, predation pressure and survival rates of the species (Holdo *et al.*, 2011; Torney *et al.*, 2018). Some of these dynamics are discussed in this study. The results, discussions, and recommendations from this study would be helpful in the management decisions for wildlife protection.

Furthermore, this study shows how different animal species (wildebeests, zebras and lions) can survive and adapt to different habitats and ecosystems. This information helps conserve wildlife, which requires that we know enough about natural behaviours (migration patterns, foraging demands and interaction with other groups) to develop effective protection measures (Sagamiko *et al.*, 2015). Such measures include detecting and acting on early clues of environmental degradation (Mduma, 1996) and changes in the reproductive outcomes of different species (Hocraft *et al.*, 2010), managing their population size to offset illegal hunting and establishing laws to protect the ecosystem (Kideghesho, 2010).

1.6.2 Contributions of the Study

This study has several contributions:

- (i) While some previously proposed models in the Serengeti and other ecosystems were based on prey-predator interactions, mostly involving wildebeests and lions, this study included three species (two mutualistic preys and one predator).
- (ii) Previous studies on the dynamics of wildebeests and zebras' mutual relations and wildebeests, zebras and lions as a prey-predator were carried out in Kruger National Park (Fay & Greef, 2006) to show the dynamics of the interacting species based on historical data. However, their models were carried out in one spatial dimension leaving out the advection and diffusion components. Therefore, this study analysed the interaction of wildebeests and zebras as mutual relations and wildebeests, zebras, and lions as prey-predator relations in two spatial and advection and diffusion parameters explaining different dynamics were used.

- (iii) Previous studies involving three species used Holling type III to analyse the predators' per capita feeding rate. This study Holling type II functional response to describe the predator's per capita feeding rate

1.6.3 Future Researchers

Ideas presented in the current study would be used as reference data in conducting new research or testing the validity of the findings of previous studies.

Wildebeest great migration is a natural phenomenon occurring for decades (Mduma, 1996). This study projects the wildebeest migration in the Serengeti ecosystem, which is the largest single movement of wild animals globally that United Nations Educational, Scientific and cultural organization (UNESCO) lists as one of the eight natural wonders of the world (Kideghesho, 2010). Future studies on the wildebeest migration can use the results and models as their reference in other ecosystem studies. Furthermore Serengeti National Park has been pivotal in building the tourism industry and contributing to economic growth in Tanzania (Mduma, 1996). Thus, research in wildlife protection in this ecosystem can use the results to ensure protection sustainability for the present and future generations.

1.7 Delineation of the Study

This study focused on modelling wildebeests, zebras and lion movement dynamics. In particular, this study intended to show wildebeest foraging processes through a mathematical model as these species undergo migration and diffusion. Furthermore, the study used mathematical models to investigate the dynamics of the mutual association of wildebeests, zebras and other ungulates as they migrate together, and their associated prey-predator interaction with lions. However, the formulated models and the results may not be perfect due to the following limitations:

- (i) Data, especially census data for lions, could not be obtained easily. This limited further analysis of the mathematical models. Furthermore, data on collared zebra and lion could not be obtained.
- (ii) Other studies on collared wildebeests, zebras, and lions must be undertaken. This will ensure real-time tracking and location of the three species that can be used to study their movement patterns in wet and dry seasons. This will help distinguish the movement patterns of both prey and predators and determine their nutritional requirements.

- (iii) Additionally, further studies should include small mammals such as antelopes and other predators such as black-spotted hyena, crocodile, leopards, cheetah, and the like to assess their contribution to the stability of the ecosystem.

The proposed mathematical models could also be extended to study different behaviours of the Serengeti ecosystem. For instance, the time delay mathematical models with harvesting, stochastic differential equations, and Markov chain models.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This Chapter presents a literature review on the works related to the modelling of wildebeest foraging processes and its interaction with zebras and lions in the Serengeti ecosystem. This study introduces various mathematical models and foraging processes, mutual associations of migrating species and diffusive prey-predator interactions.

2.2 Related Work

Despite the popular slogan “Serengeti shall never die” (Grzimek & Grzimek, 1960), the Serengeti ecosystem needs to be protected from different threats to ensure its survival for the benefit of both local and global communities. Besides, Kideghesho (2010) upholds that threats to the Serengeti ecosystem's survival, such as illegal hunting, habitat loss and human-wildlife conflicts, affect the dynamics of wildebeests and zebras and lions. Furthermore, prior studies in the Serengeti National Park explored the great migration and other dynamics of different species. The section presents these studies and establishes the gap needed to be addressed.

The studies in the Serengeti National Park had different focuses and areas of concentration. For instance, Some studies, including Mduma (1996), Sagamiko *et al.* (2015), Grange *et al.* (2004), and Ikanda and Packer (2008) explored population-level factors such as predation, poaching, and an outbreak of diseases. Studies on the search for greener pasture and water for herbivore species include Holdo *et al.* (2011). Other studies, such as Sagamiko *et al.* (2015), investigated the prey-predator interaction of only wildebeests and lions using mathematical models. Nevertheless, few attempts have been made to address wildebeest foraging processes using the concept of random walk and diffusive processes. Besides, few studies document the mutualism behaviour between wildebeest and zebra during migration, the prey-predator interaction between wildebeest, zebra and lion and the use of these for predicting movement patterns of these species in their natural environment.

Fay and Greef (2006) studied the dynamics of wildebeest, zebra, and lion as a prey-predator in the Kruger National Park. The authors developed a model that was analysed to show the dynamics of the interacting species based on historical data. However, the model was done in one dimension, where the advection and diffusion components were not considered.

In the current study, some mathematical models were developed based on partial differential equations to study and address factors that regulate foraging processes, the mutual benefit of the two prey species and how they interact with a lion in a prey-predator relationship.

2.3 Random Walk and Advection-Diffusion Equation

The advection-diffusion equation describes the transportation of different substances in moving fluids, including animals' movements (Okubo, 1986). Diffusion describes how a group of individual particles spreads out due to the irregular motion of each particle (Okubo & Levin, 2001). For animal movements, diffusion is useful in describing the distribution of a large population of animals or the expected location of an individual animal in space and time. Therefore, the advection-diffusion equation is an individual-based model (Sibert *et al.*, 1999).

The current study introduces the mathematical theory behind a simple random walk that follows Brownian motion and diffusive processes in general. The mathematical model derived includes a drift that ensures the probability of moving in a certain direction is greater, thus creating a drift-diffusion (advection-diffusion) equation (Okubo & Levin, 2001; codling *et al.*, 2008). Wildebeests in their group forage move towards their preferred forage target. Thus, such paths containing a consistent bias in a preferred forage target are biased random walks (BRWs) (Codling *et al.*, 2008). Using the Fokker-Plank equation, the BRWs were used to explain the foraging process in two dimensions.

During their lifetime, wildebeests undergo annual migration from the Serengeti national park in northern Tanzania to the Masai Mara game reserve in Kenya and back to Serengeti to search for a different favourable niche to inhabit (Holdo, 2011). The great migration involves the mass movement of millions of individual ungulates and other species in groups (Holdo, 2011). This strategy helps individuals to survive and reproduce. Wildebeest movements have been a subject of much research mostly addressing issues related to the causes of migration and how wildebeests encounter dangers during their journey (Kideghesho, 2010; Sagamiko, 2015; Hopcraft, 2014). These studies have yielded important insights into the population dynamics of wildebeest and other species in the Serengeti ecosystem. In addition, they have improved the management of different species within the ecosystem (Kideghesho, 2010). Some deterministic mathematical models have been developed to explain migrating species' different dynamics and potential drivers. However, these studies paid little attention to the role of random walk and diffusive processes in explaining forage efficiency. The current study presents the theory of

random walk and diffusive processes. It shows how these processes describe the foraging of wildebeests in the Serengeti National Park as they undergo great migration.

Random walks have been used to describe movements in different settings, and in particular, they have been used to describe movements of terrestrial vertebrates (Fagan *et al.*, 2019; Okubo & Levin, 2001). In addition, different scholars have examined other random walk models that have led to successful foraging at different times and scales. These models include; the Brownian model, which was sufficient in productive areas (Plank & James, 2008), and the optimal foraging theory (Plank & James, 2008), which predicts how an animal behaves when searching for food. Others are the correlated random walks and composite correlated random walk models (Auger-Méthé *et al.*, 2016; Raichlen *et al.*, 2014).

On the other hand, Levy's walk movement patterns have explained foraging patterns for many organisms (Raichlen *et al.*, 2014). A levy walk is a random walk search strategy used by a wide various organisms when searching for heterogeneously distributed food (Reynolds *et al.*, 2018). This search type mainly involves short movement steps (defined as the distance travelled before pausing or changing direction) combined with rarer longer movement steps creating fractal movement patterns with no characteristic scale (Raichlen *et al.*, 2014).

Levy walks have been found in human foraging patterns. For example, a study by Raichlen *et al.* (2014) demonstrated that northern Tanzania's Hadzabe hunters and gatherer perform Levy walks when searching for food. Also, Levy walks are evident in Me'Phaa of Mexico, the Brazilian Cariri, and the Amazon farmers (Reynolds *et al.*, 2018).

Random walks have been used to explain animals' foraging in connection with the advection-diffusion equation (Okubo & Levin, 2001). In addition, several studies have used the advection-diffusion equation to explain the movement of different species. For instance, Sibert *et al.* (1999) used the advection-diffusion model to describe a general quantitative framework for estimating the movement and mortality of fish populations (Skipjack) from tagging data. They used the finite difference method to solve the advection-diffusion model. The model was parameterized, and the model parameters were estimated by maximum likelihood. The authors found that Skipjack movements were highly variable at seasonal and interannual time scales.

Furthermore, the advection-diffusion equation was used to explain foraging processes to different ungulates. For instance, Fagan *et al.* (2019) considered the spatial dynamics of the population of foragers when they move via diffusion in which animals move randomly in search

of good forage and when foragers move via advection in which foragers move along the resource gradient-following search. The authors thoroughly explained how foragers actively switched between the random and gradient search following movement models as a function of their spatial context (Fagan *et al.*, 2019). However, the study was carried out in one dimension and did not show the connection between random walk and diffusive processes.

The previous studies that used the concept of the advection-diffusion equation were based on one dimension (Codling *et al.*, 2008; Raichlen *et al.*, 2014). Some did not show how diffusion and migration can explain foraging processes, especially for migrating species such as wildebeests in the Serengeti ecosystem. Therefore, the two-dimensional Fokker-Plank equation with diffusion and migration parameters was derived in this study. The model was equipped with data from the Serengeti ecosystem. The advection and diffusion movement parameters were calculated in different seasons of the year and showed how these components could be used to explain the foraging and migration of wildebeest. In addition, the relationship between random walks and advection-diffusion processes was established. Further, the effects of these parameters were used to explain the nutritional requirements of migrating wildebeests that move along different resource gradients. Data from 18 Global Positioning System (GPS) collared wildebeest were used to track their daily movement patterns. Finally, the advection-diffusion equation as an Individual-Based model (IBM) was used to arrive at the conclusions.

2.4 Diffusive Prey Predator Mathematical Models

Modelling the interaction of species is traced back to the classical works of Lotka in 1925 and Volterra in 1926, which inspired most of the studies in mathematical ecology (Wang, 2016). The Serengeti ecosystem is characterized by the interaction of different species and the natural environment. One of the most important interactions is mutualism between wildebeest (*Connochaetus taurinus*) and zebra (*Equus burchelli*). Previous studies on mutualism between different species did not consider several components, including diffusion and migration, and modelling on a two-dimensional space. For example, studies by Fay and Greeff (2006), Wang (2009), Wang (2016) and Kurowski *et al.* (2017) explained the asymptotic behaviour of solutions of a time-delayed mutualism Lotka-Volterra reaction-diffusion system. However, such studies were based on a one dimension space, and their models did not include a migration component. Ahmad and Budin (2012) analysed the effect of time delay on the stability of the mutualism model in one dimension; there were no advection and diffusion parameters.

In the current work, the Lotka-Volterra advection-diffusion reaction system explained the mutual relationship between the two prey species in two dimensions. In such differential equation models, when the diffusion component is included, the prey and predator interactions change the behaviour and nature of the whole model to be a partial differential equation (Fagan *et al.*, 2019; Ahmad & Budin, 2012). The resulting partial differential equation is viewed as part of the spatial population model and thus can be categorized as a reaction-diffusion system (Ahmad & Budin, 2012). The Lotka-Volterra predator-prey models with diffusion are represented by similar mathematical formulation to predator-prey interactions.

Such diffusive mathematical models have been used in literature to study different behaviours. For instance, Qiao *et al.* (2014) applied the prey-predator diffusive model to study the spread of diseases when the prey is infected. The authors established conditions for the stability of the ecosystem, but their model did not clearly show the role of diffusion in the spread of diseases.

Furthermore, Zhu and Zhang (2018) analysed the prey's group defence against the predators using diffusive mathematical models showing the role of system parameters leading to spatial-temporal patterns. Still their study did not show clearly how the prey groups are formed and what behaviours led to the group defence.

Other two-prey and one-predator mathematical models include the study by Lee (2007) on two-prey and one-predator that describes the effects of the remained Carcass on the stability of the dynamical system. Also, Murthy *et al.* (2017) used a discrete three species mathematical model to study a fishery system with two prey and two predators (one species acts as prey and predator simultaneously). The authors used a discrete model to analyse the dynamics of the model; however, their mathematical model had no diffusion and migration components and was analysed in one dimension. Furthermore, Lian *et al.* (2012) studied the pattern formation in a cross-diffusive Holling type III ratio-dependent prey-predator model. Their model was useful in explaining the system dynamics induced by diffusion. However, it was based on one prey-one predator and had no migration components.

Some studies, including Wang (2016) and Liu (2010), were based on one prey and one predator models with diffusion and Holling type III functional response. However, none of these has included three species (two mutualistic preys and one predator) and advection and diffusion parameters in explaining different dynamics.

A literature review of the current study has highlighted several gaps to be filled. This study used the concept of random walk and diffusive processes to study the foraging efficiency of wildebeest herds. In addition, the current study used advective and diffusive parameters were used to analyse how wildebeests interact with other individuals. In particular, the Holling type II functional response extensively explored zebras and the dynamics of wildebeests, zebras, and lions as prey-predator.

2.5 Chapter Summary

This chapter exhaustively reviewed the literature on mathematical models related to advection-diffusion and prey-predator dynamics of three migrating species in the Serengeti ecosystem. In the next chapter, the deterministic mathematical models are developed to describe the random wildebeest walk, the mutual association between wildebeests and zebras, and their prey-predator interaction with lions.

is home to other resident wildlife, particularly the big five: The Lion, the African Leopard, the African Elephant, the Black Rhino and the African Buffalo (Sagamiko *et al.*, 2015).

3.3 Data Sources

Data used in the current study were provided by the University of Glasgow, United Kingdoms (UK). These were data collected between 1999 and 2007 in the Serengeti ecosystem from 18 different collared wildebeests. Collars were configured with GPS to record parameters on wildebeest locations (in two dimensions) and distances. Other parameters include the date and the time, where 6 hours were the time step for recording wildebeest distances and locations.

In addition, this study used reports on rainfall and wildebeests and zebras data from Tanzania Wildlife Research Institute. The data were based on previous aerial census counts of wildebeest and zebra populations in the Serengeti ecosystem from 1961 to 2010. Meanwhile, rainfall data (in mm) spanned from July to November from 1961 to 1994. The following sections detail the methods used to achieve each study objective.

3.4 The Concept of Random Walk and Diffusive Processes

3.4.1 The Fokker-Planck Equation (2D Random Walk)

The Fokker-Planck equation (2D random walk) considers the Brownian motion in two dimensions to include movements and probabilities that are spatially dependent (Okubo & Levin, 2001).

Suppose that individual moves on a two-dimensional lattice; at each time step τ , an individual can move a distance δ either up, down, left, or right with probabilities dependent on location given by $u(x, y)$, $d(x, y)$, $l(x, y)$ and $r(x, y)$ respectively with $u + d + l + r \leq 1$, or remain at the same location with probability $1 - u(x, y) - d(x, y) - l(x, y) - r(x, y)$.

To calculate the probability of a walker jumping on a two-dimensional lattice at a one-time step based on the previous time step at a time $(m + 1)\tau$, should consider that the probability of the walker in position n is equal to the probability that it was already there times the probability that it stayed there, plus the probability that it was one position to the left times the probability that it jumped to the right, plus the probability that it was one position to the right times the probability that it jumped to the left, plus the probability that it was one position up times the

probability that it jumped down, plus the probability that it was one position down times the probability that it jumped up.

This can be expressed as follows:

$$p[n\delta, (m+1)\tau] = (1-r-l-u-d)p(n\delta, m\tau) + rp[(n-1)\delta, m\tau] + lp[(n+1)\delta, m\tau] + up[(n-1)\delta, m\tau] + dp[(n+1)\delta, m\tau] \quad (1)$$

Where $p(0,0) = 1$ and $p(n\delta, 0) = 0$ for $n \neq 0$

Rewriting Equation (1) gives the following:

$$p[n\delta, (m+1)\tau] - p(n\delta, m\tau) = -(r+l+u+d)p(n\delta, m\tau) + rp[(n-1)\delta, m\tau] + lp[(n+1)\delta, m\tau] + up[(n-1)\delta, m\tau] + dp[(n+1)\delta, m\tau] \quad (2)$$

Using Taylor series expansion to calculate the probability function $p[n\delta, (m+1)\tau]$:

$$\begin{aligned} \text{From } p[(n \pm 1)\delta, m\tau] &= p(n\delta, m\tau) \pm \delta \frac{\partial p}{\partial x} + \frac{1}{2} \delta^2 \frac{\partial^2 p}{\partial x^2} \pm \frac{1}{6} \delta^3 \frac{\partial^3 p}{\partial x^3} + O(\delta^4) + p(n\delta, m\tau) \pm \\ &\delta \frac{\partial p}{\partial y} + \frac{1}{2} \delta^2 \frac{\partial^2 p}{\partial y^2} \pm \frac{1}{6} \delta^3 \frac{\partial^3 p}{\partial y^3} + O(\delta^4) \end{aligned} \quad (3)$$

Then,

$$p[n\delta, (m+1)\tau] = p(n\delta, m\delta) + \tau \frac{\partial p}{\partial t} + O(\tau^2) \quad (4)$$

All the derivatives are taken at $(x = y = n\delta, t = m\tau)$

Therefore, the left-hand side of Equation (2) becomes:

$$p(n\delta, m\tau) + \tau \frac{\partial p}{\partial t} + O(\tau^2) - p(n\delta, m\tau) = \tau \frac{\partial p}{\partial t} + O(\tau^2) \quad (5)$$

The right-hand side of Equation (2) can be re written in the following ways:

$$\begin{aligned} &\frac{1}{2}(r+l)(p[(n-1)\delta, m\tau] - 2p(n\delta, m\tau) + p[(n+1)\delta, m\tau]) - \frac{1}{2}(r-l)(p[(n+1)\delta, m\tau] - \\ &p[(n-1)\delta, m\tau]) + \frac{1}{2}(u+d)(p[(n-1)\delta, m\tau] - 2p(n\delta, m\tau) + p[(n+1)\delta, m\tau]) - \frac{1}{2}(u- \\ &d)(p[(n+1)\delta, m\tau] - p[(n-1)\delta, m\tau]) \end{aligned} \quad (6)$$

The first term of the right-hand side of Equation (6) can be expanded by the Taylor series as follows:

$$\begin{aligned}
& \frac{1}{2}(r+l) \left(p(n\delta, m\tau) - \delta \frac{\partial p}{\partial x} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial x^2} - \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial x^3} + O(\delta^4) - 2p(n\delta, m\tau) + p(n\delta, m\tau) + \right. \\
& \left. \delta \frac{\partial p}{\partial x} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial x^2} + \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial x^3} + O(\delta^4) \right) + \frac{1}{2}(u+d) \left(p(n\delta, m\tau) - \delta \frac{\partial p}{\partial y} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial y^2} - \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial y^3} + \right. \\
& \left. O(\delta^4) - 2p(n\delta, m\tau) + p(n\delta, m\tau) + \delta \frac{\partial p}{\partial y} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial y^2} + \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial y^3} + O(\delta^4) \right) \quad (7)
\end{aligned}$$

Simplifying gives:

$$(r+l) \left(\frac{\delta^2}{2} \frac{\partial^2 p}{\partial x^2} + O(\delta^4) \right) + (u+d) \left(\frac{\delta^2}{2} \frac{\partial^2 p}{\partial y^2} + O(\delta^4) \right) \quad (8)$$

And the last term of the right-hand side of Equation (6) can be expanded by the Taylor series to give:

$$\begin{aligned}
& -\frac{1}{2}(r-l) \left(p(n\delta, m\tau) + \delta \frac{\partial p}{\partial x} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial x^2} + \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial x^3} + O(\delta^4) - p(n\delta, m\tau) + \delta \frac{\partial p}{\partial x} - \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial x^2} + \right. \\
& \left. \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial x^3} - O(\delta^4) \right) - \frac{1}{2}(u-d) \left(p(n\delta, m\tau) + \delta \frac{\partial p}{\partial y} + \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial y^2} + \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial y^3} + O(\delta^4) - \right. \\
& \left. p(n\delta, m\tau) + \delta \frac{\partial p}{\partial y} - \frac{1}{2}\delta^2 \frac{\partial^2 p}{\partial y^2} + \frac{1}{6}\delta^3 \frac{\partial^3 p}{\partial y^3} - O(\delta^4) \right) \quad (9)
\end{aligned}$$

Simplifying and ignoring higher-order terms gives:

$$-(r-l) \left(\delta \frac{\partial p}{\partial x} \right) - (u-d) \left(\delta \frac{\partial p}{\partial y} \right) \quad (10)$$

Combining the left-hand side Equation (8) and right-hand side Equation (10) gives:

$$\begin{aligned}
\tau \frac{\partial p}{\partial t} &= (r+l) \left(\frac{\delta^2}{2} \frac{\partial^2 p}{\partial x^2} \right) + (u+d) \left(\frac{\delta^2}{2} \frac{\partial^2 p}{\partial y^2} \right) - (r-l) \left(\delta \frac{\partial p}{\partial x} \right) - (u-d) \left(\delta \frac{\partial p}{\partial y} \right) + O(\delta^4) + \\
& O(\tau^2) \quad (11)
\end{aligned}$$

Divide by τ , gives:

$$\begin{aligned}
\frac{\partial p}{\partial t} &= (r+l) \left(\frac{\delta^2}{2\tau} \frac{\partial^2 p}{\partial x^2} \right) - (r-l) \left(\frac{\delta}{\tau} \frac{\partial p}{\partial x} \right) + O\left(\frac{\delta^4}{\tau}\right) + O(\tau) + (u+d) \left(\frac{\delta^2}{2\tau} \frac{\partial^2 p}{\partial y^2} \right) - (u-d) \\
& \left(\frac{\delta}{\tau} \frac{\partial p}{\partial y} \right) + O\left(\frac{\delta^4}{\tau}\right) + O(\tau) \quad (12)
\end{aligned}$$

Let $k_1 = r+l$; $k_2 = u+d$; $\epsilon_1 = r-l$; $\epsilon_2 = u-d$, it can be defined that:

$$\frac{\partial p}{\partial t} = \frac{k_1 \delta^2}{2\tau} \frac{\partial^2 p}{\partial x^2} - \frac{\epsilon_1 \delta}{\tau} \frac{\partial p}{\partial x} - \frac{k_2 \delta^2}{2\tau} \frac{\partial^2 p}{\partial y^2} - \frac{\epsilon_2 \delta}{\tau} \frac{\partial p}{\partial y} + O\left(\frac{\delta^4}{\tau}\right) + O(\tau) \quad (13)$$

Taking limits as $\tau \rightarrow 0$ and $\delta \rightarrow 0$, such that the following limits are positive and definite:

$$D_x = \lim_{\delta, \tau \rightarrow 0} \frac{k_1 \delta^2}{2\tau}, D_y = \lim_{\delta, \tau \rightarrow 0} \frac{k_2 \delta^2}{2\tau}, u_x = \lim_{\delta, \tau, \epsilon_1 \rightarrow 0} \frac{\epsilon_1 \delta}{\tau} \text{ and } u_y = \lim_{\delta, \tau, \epsilon_2 \rightarrow 0} \frac{\epsilon_2 \delta}{\tau}$$

Taking the limits as $\delta, \tau, \epsilon_1, \epsilon_2 \rightarrow 0$ such that D_x, D_y, u_x, u_y all tend to constants gives:

$$\frac{\partial p}{\partial t} = -\nabla \cdot (up) + \nabla \cdot (D\nabla p) \quad (14)$$

Equation (14) is the same as:

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x} \left(D_x \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_y \frac{\partial p}{\partial y} \right) - \frac{\partial}{\partial x} (u_x p) - \frac{\partial}{\partial y} (u_y p) \quad (15)$$

Solving Equation (15) gives the following probability mass function:

$$P(x, y, t) = \frac{1}{4\pi t \sqrt{D_x D_y}} e^{-\left(\frac{(x-u_x t)^2}{4D_x t} + \frac{(y-u_y t)^2}{4D_y t} \right)} \quad (16)$$

The simulations were done in a python computer program (the corresponding codes are attached in Appendix 1). The results and discussion for objective one are presented in Chapter four.

3.5 Mutualism Reaction-Diffusion Models for Wildebeests and Zebras

3.5.1 Assumptions of the Model

- (i) There are two populations, namely, wildebeests and zebras, whose population densities are denoted by w and z , respectively.
- (ii) The population densities of wildebeests and zebras grow according to the logistic law of growth r_w and r_z , respectively.
- (iii) Two prey species are mutualistic in nature.
- (iv) There is a death rate for both prey species from factors such as predation and drought.
- (v) The two prey species' growth approach the carrying capacities K_w and K_z , respectively.

Based on the above assumptions, the Lotka-Volterra mutualistic model (Fay & Greef, 2006; Wang, 2009; Wang, 2016) for the two prey species can be defined as:

$$\left. \begin{aligned} \frac{dw}{dt} &= \frac{r_w}{K_w} w(K_w - w + a_{12}z) - d_w w \\ \frac{dz}{dt} &= \frac{r_z}{K_z} z(K_z - z + a_{21}w) - d_z z \end{aligned} \right\} \quad (17)$$

The coefficients a_{12} and a_{21} denote the interaction (mutual relations) between wildebeests and zebras.

3.5.2 The Mutualism Model with Diffusion and Migration

The Lotka-Volterra model (17), which resembles the model developed in Kruger National Park by Fay and Greef (2006) was extended to include the diffusion and migration components of the mutualism model. The presence of diffusion and migration parameters (constant flow terms) for the two prey species is considered the principal process of motion; this changes the behaviour and nature of the whole model (Yamada *et al.*, 2007; Liu, 2010). It is now a partial differential equation and can be categorized as a reaction-diffusion system.

3.5.3 Reaction-Diffusion System

In mathematical biology, reaction-diffusion equations arise as models for describing the densities of substances or organisms that disperse through space by mechanisms like Brownian motion, random walks, or hydrodynamic turbulence. The models also describe organisms that react to each other and their surroundings to affect their local densities (Yamada *et al.*, 2007). These spatially explicit models describe population densities and treat space and time as continuous. The reaction-diffusion equations describe the behaviour of chemical systems where diffusion of material competes with the production of that material by some form of chemical reaction (Yamada *et al.*, 2007; Liu, 2010).

For wildebeest and zebra annual migration, the reaction refers to the mass movement of these prey species in the Serengeti ecosystem. As the animals move from place to place to search for forage resources through advection trends, they spread to acquire and utilize them through diffusion movements.

This study considers a reaction-diffusion equation with flow terms. The Equation considered in the current study comprises a reaction term, an advection term, and a diffusion term, i.e. the typical form of this Equation can be stated as follows:

$$\left. \begin{aligned} \frac{\partial w}{\partial t} + C_w \cdot \nabla w &= D_w \nabla^2 w + f(w, z) \\ \frac{\partial z}{\partial t} + C_z \cdot \nabla z &= D_z \nabla^2 z + g(w, z) \end{aligned} \right\} \quad (18)$$

The quantities w and z are state variables that describe the density/concentration of substance/individuals (wildebeest and zebra populations, respectively), which diffuse with their diffusion coefficients D_w and D_z respectively and grow according to their specific rules $f(w, z)$ and $g(w, z)$ respectively. The symbol ∇ is the gradient operator and ∇^2 is the Laplacian operator.

The quantities D_w and D_z are positive diagonal diffusion coefficient matrices for wildebeest and zebra, respectively. Their concentrations determine the velocity of animals as follows:

$$\left. \begin{aligned} C_w &= M_w \nabla w \\ C_z &= M_z \nabla z \end{aligned} \right\} \quad (19)$$

where M_w and M_z are constant parameters expressing the flow intensity for wildebeest and zebra, respectively.

3.5.4 The Reaction

In the mathematical ecology context, the reaction terms $f(w, z)$ and $g(w, z)$, in the reaction-diffusion equations, resemble the reaction terms in non-spatial population models based on ordinary differential equations. Thus, for two mutually interacting prey species (zebra and wildebeest), the Lotka–Volterra reaction terms (for the mathematical mutualism model) can be defined as:

$$\left. \begin{aligned} f(w, z) &= \frac{r_w}{K_w} w (K_w - w + a_{12} z) - d_w w \\ g(w, z) &= \frac{r_z}{K_z} z (K_z - z + a_{21} w) - d_z z \end{aligned} \right\} \quad (20)$$

From Equations (18) and (20), the two-dimensional reaction diffusion-advection model can be written as:

$$\left. \begin{aligned} \frac{\partial w}{\partial t} + C_w \cdot \nabla w &= \frac{r_w}{K_w} w (K_w - w + a_{12}z) - d_w w + D_w \nabla^2 w \\ \frac{\partial z}{\partial t} + C_z \cdot \nabla z &= \frac{r_z}{K_z} z (K_z - z + a_{21}w) - d_z z + D_z \nabla^2 z \end{aligned} \right\} \quad (21)$$

The symbol ∇ is the gradient operator defined in two dimensions as: $\nabla(\cdot) = \frac{\partial(\cdot)}{\partial x} \vec{i} + \frac{\partial(\cdot)}{\partial y} \vec{j}$, and

the Laplacian operator ∇^2 in two dimensions is defined as: $\nabla^2 = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2}$.

Equation 21 is the reaction-diffusion system. It treats space as a continuum and depicts the population densities of interacting species over time. The dynamical behaviour of wildebeests and zebras is based on Equation 21.

3.5.5 Local Stability Analysis

The local stability analysis is referred to as the Hopf bifurcation. The Hopf bifurcation is space independent and breaks the temporal symmetry of the system (Liu, 2010; Wang *et al.*, 2016), which causes uniform oscillations in space and time. The onset of Hopf bifurcation instability occurs when the pair of imaginary eigenvalues crosses the real axis from negative to positive (Liu, 2010; Wang *et al.*, 2016). This situation occurs only when diffusion and migration components vanish (Liu, 2010; Wang *et al.*, 2016; Kurowski *et al.*, 2017). When wildebeests and zebras occupy a niche, their dynamical behaviour must be analysed when diffusion and migration vanish. From Equation 21, when diffusion and migration components vanished, the following equation was obtained:

$$\left. \begin{aligned} \frac{r_w}{K_w} w (K_w - w + a_{12}z) - d_w w &= 0 \\ \frac{r_z}{K_z} z (K_z - z + a_{21}w) - d_z z &= 0 \end{aligned} \right\} \quad (22)$$

From Equation (22), using maple software and simplifying, four equilibrium points were obtained:

$$(w^*, z^*) = (0, 0), (w^*, z^*) = \left(0, \frac{K_z}{r_z} (r_z - d_z)\right), (w^*, z^*) = \left(\frac{K_w}{r_w} (r_w - d_w), 0\right), \text{ and}$$

$$(w^*, z^*) = \left(\frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}, \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \right)$$

The equilibrium point $(w^*, z^*) = (0, 0)$ corresponds to the extinction of both species.

The equilibrium point: $(w^*, z^*) = \left(0, \frac{K_z}{r_z}(r_z - d_z)\right)$ corresponds to the extinction of wildebeest. This implies that zebras can exist in the absence of wildebeests.

The equilibrium point: $(w^*, z^*) = \left(\frac{K_w}{r_w}(r_w - d_w), 0\right)$ corresponds to the extinction of zebras.

This implies that wildebeest can exist in the absence of zebras.

The only equilibrium point that can be reached in the first quadrant is:

$$(w^*, z^*) = \left(\frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}, \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \right)$$

The zero growth isoclines $w^* = \frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}$ and

$$z^* = \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})}$$
 may converge or diverge.

They converge if $1 - a_{12}a_{21} > 0 \Rightarrow a_{12}a_{21} < 1$ provided the birth rates of these species must be greater than their corresponding death rates, that is $r_w > d_w$, and $r_z > d_z$.

In this case, the two isoclines cross each other, and the orbits approach a stable node in the interior of the first quadrant. Since the slopes of the two zero-growth isoclines are positive, the coordinates of this equilibrium are greater than the carrying capacities K_w and K_z and each species surpasses its respective carrying capacity because of the mutualistic benefits that both species experience.

3.5.6 Local Stability Analysis

The local stability analysis is referred to as the Hopf bifurcation. The Hopf bifurcation is space independent and breaks the temporal symmetry of the system (Liu, 2010; Wang *et al.* 2016), which causes oscillations to occur that are uniform in space and periodic in time. The onset of Hopf bifurcation instability occurs when the pair of imaginary eigenvalues crosses the real axis from the negative to the positive side (Liu, 2010; Wang *et al.*, 2016). This situation occurs only when diffusion and migration components vanish (Liu, 2010; Wang *et al.*, 2016; Kurowski *et al.*, 2017). When wildebeest and zebra occupy a niche, their dynamical behaviour must be

analysed when diffusion and migration vanish. From Equation 21, when diffusion and migration components vanish, the following Equation was obtained:

$$\left. \begin{aligned} \frac{r_w}{K_w} w(K_w - w + a_{12}z) - d_w w &= 0 \\ \frac{r_z}{K_z} z(K_z - z + a_{21}w) - d_z z &= 0 \end{aligned} \right\} \quad (22)$$

From Equation (22), using maple software and simplifying, four equilibrium points were obtained:

$$(w^*, z^*) = (0,0), (w^*, z^*) = \left(0, \frac{K_z}{r_z}(r_z - d_z)\right), (w^*, z^*) = \left(\frac{K_w}{r_w}(r_w - d_w), 0\right), \text{ and}$$

$$(w^*, z^*) = \left(\frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}, \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \right)$$

The equilibrium point $(w^*, z^*) = (0,0)$ corresponds to the extinction of both species.

The equilibrium point $(w^*, z^*) = \left(0, \frac{K_z}{r_z}(r_z - d_z)\right)$ corresponds to the extinction of wildebeest. This implies that, zebra can exist in the absence of wildebeest.

The equilibrium point $(w^*, z^*) = \left(\frac{K_w}{r_w}(r_w - d_w), 0\right)$ corresponds to the extinction of zebra. This implies that, wildebeest can exist in the absence of zebra.

The only equilibrium point that can occur in the first quadrant is:

$$(w^*, z^*) = \left(\frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}, \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \right)$$

The zero growth isoclines $w^* = \frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}$ and

$$z^* = \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \text{ may converge or diverge.}$$

They converge if $1 - a_{12}a_{21} > 0 \Rightarrow a_{12}a_{21} < 1$ provided the birth rates of these species must be greater than their corresponding death rates, that is $r_w > d_w$, and $r_z > d_z$.

In this case, the two isoclines cross each other, and the orbits approach a stable node in the interior of the first quadrant. Since the slopes of the two zero-growth isoclines are positive, the coordinates of this equilibrium are greater than the carrying capacities K_w and K_z and each

species surpasses its respective carrying capacity because of the mutualistic benefits that both species experience.

3.5.7 Local Stability of the Equilibrium Points

The local asymptotic stability of the co-existence equilibrium point was studied by computing the Jacobian matrix and testing whether the respective eigenvalues were negative. From the system of Equation (21), the system can be defined as:

$$\begin{cases} f(w, z) = \frac{r_w}{K_w} w(K_w - w + a_{12}z) - d_w w \\ g(w, z) = \frac{r_z}{K_z} z(K_z - z + a_{21}w) - d_z z \end{cases} \quad (23)$$

The Jacobian matrix of the system (23) is given by:

$$J(E_i) = \begin{pmatrix} \frac{\partial f}{\partial w} & \frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial w} & \frac{\partial g}{\partial z} \end{pmatrix} \quad (24)$$

for $i = 0, 1, 2, 3$

This gives:

$$J(E_i) = \begin{pmatrix} \frac{r_w}{K_w} (K_w - 2w^* + a_{12}z^*) - d_w & \frac{r_w}{K_w} a_{12}w^* \\ \frac{r_z}{K_z} a_{21}z^* & \frac{r_z}{K_z} (K_z - 2z^* + a_{21}w^*) - d_z \end{pmatrix} \quad (25)$$

For the equilibrium point $E_0(w^*, z^*) = (0, 0)$, the Jacobian matrix of the system (25) is:

$$J(E_0) = \begin{pmatrix} (r_w - d_w) & 0 \\ 0 & (r_z - d_z) \end{pmatrix}.$$

Since the birth rate should exceed the death rate, the eigenvalues of the system are $r_w - d_w > 0$ and $r_z - d_z > 0$. Hence the point is unstable.

For the equilibrium point $E_1(w^*, z^*) = \left(0, \frac{K_z}{r_z} (r_z - d_z)\right)$, the Jacobian matrix of the system

$$\text{is } J(E_1) = \begin{pmatrix} (r_w - d_w) + \frac{a_{12}r_w K_z}{r_z K_w} (r_z - d_z) & 0 \\ a_{21}(r_z - d_z) & -(r_z - d_z) \end{pmatrix}$$

The eigenvalues of the system are $-(r_z - d_z) < 0$ and $(r_w - d_w) + \frac{a_{12}r_w K_z}{r_z K_w}(r_z - d_z) > 0$.

This gives a saddle point; hence, the point is unstable.

For the equilibrium point: $(w^*, z^*) = \left(\frac{K_w}{r_w}(r_w - d_w), 0\right)$, the Jacobian matrix of the system is

$$J(E_2) = \begin{pmatrix} -(r_w - d_w) & a_{12}(r_w - d_w) \\ 0 & (r_z - d_z) + \frac{a_{21}r_z K_w}{r_w K_z}(r_w - d_w) \end{pmatrix}.$$

The eigenvalues of the system are $-(r_w - d_w) < 0$ and $(r_z - d_z) + \frac{a_{21}r_z K_w}{r_w K_z}(r_w - d_w) > 0$.

This gives a saddle point; hence the equilibrium point is unstable.

From the co-existence equilibrium point $E_3(w^*, z^*)$, where:

$$E_3(w^*, z^*) = \left(\frac{-a_{12}d_z k_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w}{r_z r_w (1 - a_{12}a_{21})}, \frac{-a_{21}d_w k_w r_z + a_{21}k_w r_z r_w - d_z k_z r_w + k_z r_z r_w}{r_z r_w (1 - a_{12}a_{21})} \right), \text{ the}$$

Jacobian matrix of the system using maple software is given as:

$$J(E_3) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

Where,

$$A_{11} = \frac{a_{12}d_z K_z r_w - a_{12}k_z r_z r_w + d_w k_w r_z - k_w r_z r_w}{K_w r_z (1 - a_{12}a_{21})}$$

$$A_{21} = \frac{a_{12}(-a_{12}d_z K_z r_w + a_{12}k_z r_z r_w - d_w k_w r_z + k_w r_z r_w)}{K_w r_z (1 - a_{12}a_{21})}$$

$$A_{12} = \frac{a_{21}(-a_{21}d_w K_w r_z + a_{21}k_w r_w r_z - d_z k_z r_w + k_z r_w r_z)}{K_z r_w (1 - a_{12}a_{21})}$$

$$A_{22} = \frac{a_{21}d_w K_w r_z - a_{21}k_w r_w r_z + d_z k_z r_w - k_z r_w r_z}{K_z r_w (1 - a_{12}a_{21})}$$

Theorem 1: Let $E_3(w^*, z^*)$ be the positive equilibrium point of the model (22). If $1 - a_{12}a_{21} > 0$, then $E_3(w^*, z^*)$ is asymptotically stable (Ahmad & Budin, 2012; Kurowski *et al.*, 2017).

The stability analysis of the four equilibrium points can be illustrated in Fig. 5.

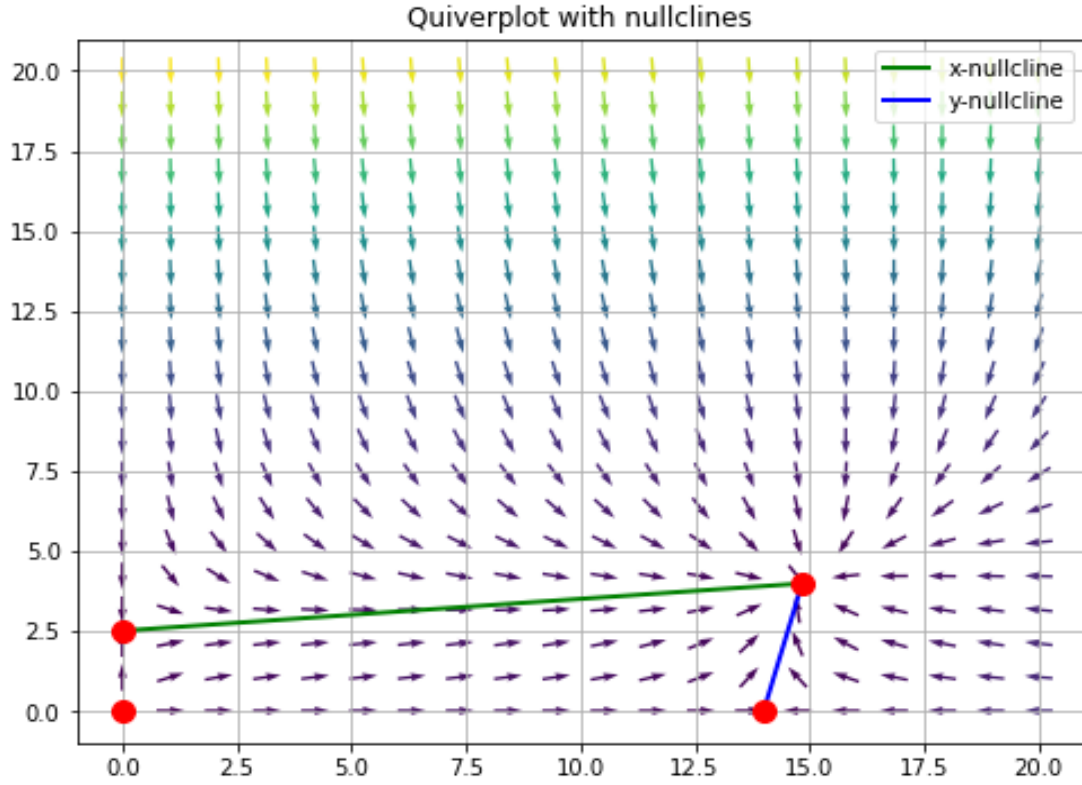


Figure 5: Trajectories around the equilibrium points

In Fig. 5, the red circles indicate the four equilibrium points, and the arrows indicate the directional field showing that the equilibrium point $E_3(w^*, z^*)$ in the first quadrant is asymptotically stable.

3.5.8 Stability Analysis of the Model with Diffusion and Migration

From the mathematical model (21), let

$$\left. \begin{aligned} f(w, z) &= \frac{r_w}{K_w} w(K_w - w + a_{12}z) - d_w w \\ g(w, z) &= \frac{r_z}{K_z} z(K_z - z + a_{21}w) - d_z z \end{aligned} \right\} \quad (26)$$

We assume the diffusion and migration in each spatial dimension are the same. Equation 21 can be rearranged as:

$$\left. \begin{aligned} \frac{\partial w}{\partial t} &= D_w \nabla^2 w - C_w \nabla w + \frac{r_w}{K_w} w(K_w - w + a_{12}z) - d_w w \\ \frac{\partial z}{\partial t} &= D_z \nabla^2 z - C_z \nabla z + \frac{r_z}{K_z} z(K_z - z + a_{21}w) - d_z z \end{aligned} \right\} \quad (27)$$

Equations (27) can be written as:

$$\begin{cases} \frac{\partial w}{\partial t} = f(w, z) + D_w \nabla^2 w - C_w \nabla w \\ \frac{\partial z}{\partial t} = g(w, z) + D_z \nabla^2 z - C_z \nabla z \end{cases} \quad (28)$$

where $f(w, z)$ and $g(w, z)$ are nonlinear dynamic systems (reaction kinetics) (Liu, 2010; Wang *et al.*, 2016).

The homogeneous steady-state values of w and z are solutions to the reaction terms $f(w, z) = 0$ and $g(w, z) = 0$.

Diffusion and advection play a crucial role; as the prey species migrate from place to place, they lead to diffusion and advection-induced instability. In particular, the instability criteria of the spatially homogeneous state determine the pattern forming processes in a nonequilibrium system.

The conditions of diffusion and migration induced instability can be obtained as follows:

Assume small perturbations δ_w, δ_z about the equilibrium stable state (w_0, z_0) as $w = w_0 + \delta_w$ and $z = z_0 + \delta_z$ and taking Taylor expansion of the non-linear functions $f(w, z)$ and $g(w, z)$ about the stationary state; thus, for the diffusion, one can write:

$$\begin{cases} \frac{\partial \delta_w}{\partial t} = \left(\frac{\partial f}{\partial w} \right)_{(w_0, z_0)} \delta_w + \left(\frac{\partial f}{\partial z} \right)_{(w_0, z_0)} \delta_z + D_w \nabla^2 \delta_w \\ \frac{\partial \delta_z}{\partial t} = \left(\frac{\partial g}{\partial w} \right)_{(w_0, z_0)} \delta_w + \left(\frac{\partial g}{\partial z} \right)_{(w_0, z_0)} \delta_z + D_z \nabla^2 \delta_z \end{cases} \quad (29)$$

And for migration, it is written as:

$$\begin{cases} \frac{\partial \delta_w}{\partial t} = \left(\frac{\partial f}{\partial w} \right)_{(w_0, z_0)} \delta_w + \left(\frac{\partial f}{\partial z} \right)_{(w_0, z_0)} \delta_z - C_w \nabla_w \delta_w + D_w \nabla^2 \delta_w \\ \frac{\partial \delta_z}{\partial t} = \left(\frac{\partial g}{\partial w} \right)_{(w_0, z_0)} \delta_w + \left(\frac{\partial g}{\partial z} \right)_{(w_0, z_0)} \delta_z - C_z \nabla_z \delta_z + D_z \nabla^2 \delta_z \end{cases} \quad (30)$$

The perturbation can be assumed to be harmonic in space, and the spatial variation can be expressed as $e^{ik_n \chi}$ while the temporal variation which allows this perturbation to grow can be expressed as $e^{\omega_n t}$. The parameters k_n stands for wave number, $\chi = (x, y)$ is the spatial vector in two dimensions and ω_n is the growth rate of perturbations. Thus, one can write:

$$\delta_w = \delta_{w_0} e^{ik_n \chi} e^{\omega_n t}, \quad \delta_z = \delta_{z_0} e^{ik_n \chi} e^{\omega_n t} \quad (31)$$

This leads to an eigenvalue Equation of the form:

$$A\delta\chi = \omega_n\delta\chi \quad (32)$$

With,

$$A = \begin{pmatrix} f_w - D_w k_n^2 & f_z \\ g_w & g_z - D_z k_n^2 \end{pmatrix}, \quad \delta\chi = \begin{pmatrix} \delta w_0 \\ \delta z_0 \end{pmatrix} \text{ for diffusion and}$$

$$A = \begin{pmatrix} f_w + C_w - D_w k_n & f_z \\ g_w & g_z + C_z - D_z k_n \end{pmatrix}, \quad \delta\chi = \begin{pmatrix} \delta w_0 \\ \delta z_0 \end{pmatrix} \text{ for migration}$$

The perturbation amplitudes δ_{z_0} and δ_{w_0} can be non-zero if and only if:

$$\det(A - \omega_n I) = 0.$$

This gives the characteristic polynomial defined as:

$$\omega_n^2 + [(D_w + D_z)k_n^2 - f_w - g_z]\omega_n + D_w D_z k_n^4 - k_n^2(D_z f_w + D_w g_z) + f_w g_z - f_z g_w = 0 \quad (33)$$

for diffusion and

$$\omega_n^2 + [(-C_w - C_z)k_n - f_w - g_z]\omega_n + C_w C_z k_n^2 + k_n(C_z f_w + C_w g_z) + f_w g_z - f_z g_w = 0, \text{ for migration.} \quad (34)$$

A fluctuation associated with the eigenvalue ω_n grows if:

$$Re(\omega_n) > 0.$$

Thus,

$$\text{At the onset of instability, } Re(\omega_n) = 0.$$

In that case, the term independent of ω_n in equations (33 and 34) is reduced to zero. This determines the critical wave number k_c for diffusion as:

$$k_c^2 = \frac{(D_z f_w + D_w g_z) \pm \sqrt{(D_z f_w + D_w g_z)^2 - 4D_w D_z (f_w g_z - f_z g_w)}}{2D_w D_z} \quad (35)$$

and for migration, the following can be obtained:

$$k_c = \frac{-(C_z f_w + C_w g_z) \pm \sqrt{(C_z f_w + C_w g_z)^2 - 4C_w C_z (f_w g_z - f_z g_w)}}{2C_w C_z} \quad (36)$$

which in turn determines the critical length $L_c(n) = \frac{n\pi}{k_c}$. Thus, the fluctuations associated with the frequency ω_n are amplified in the system of length L if $L > L_c(n)$. This is a necessary condition for the development of an inhomogeneous state by the growth of fluctuations associated with wavenumber k_n . Further, the most unstable wavenumber k_{max} , can be obtained from the condition:

$$\frac{\partial}{\partial k} Re[\omega(k)] = 0 \text{ at } k = k_{max}$$

Giving:

$$k_{max} = \left[\frac{D_z f_w + D_w g_z}{2D_w D_z} \right]^{1/2} = \left[\frac{(f_w g_z - f_z g_w)}{D_w D_z} \right]^{1/4} \quad (37)$$

for diffusion and

$$k_{max} = - \left[\frac{C_z f_w + C_w g_z}{2C_w C_z} \right] = \left[\frac{(f_w g_z - f_z g_w)}{D_w D_z} \right]^{1/2} \text{ for migration.} \quad (38)$$

Therefore, from equations (35, 36) and (37, 38), the necessary conditions for the onset of Turing instability for model (27) are as follows:

$$f_w + g_z < 0 \quad (39)$$

$$f_w g_z - f_z g_w > 0 \quad (40)$$

$$D_z f_w + D_w g_z > 0 \quad (41a)$$

$$(C_z f_w + C_w g_z) < 0 \quad (41b)$$

$$(D_z f_w + D_w g_z)^2 > 4D_w D_z (f_w g_z - f_z g_w) \quad (42a)$$

$$(C_z f_w + C_w g_z)^2 > 4C_w C_z (f_w g_z - f_z g_w) \quad (42b)$$

It can be observed that the dispersion relation that drives the system of equations to instability depends on the variation of diffusion and migration terms.

3.5.9 Numerical Methods

The explicit Euler method, numerical solution, was used to discretize Equation (21), where the model was discretized in time derivative by using the forward difference rule.

Let $C_w = C_{11} = C_{12}$; $C_z = C_{21} = C_{22}$ and $D_z = D_{21} = D_{22}$ $D_w = D_{11} = D_{12}$ as defined below:

$$\begin{aligned} \left. \frac{\partial w}{\partial t} \right|_{i,j}^{n+1} &= \frac{w_{i,j}^{n+1} - w_{i,j}^n}{\Delta t}, & \left. \frac{\partial z}{\partial t} \right|_{i,j}^{n+1} &= \frac{z_{i,j}^{n+1} - z_{i,j}^n}{\Delta t}, \\ C_{11} \frac{\partial w}{\partial x} &= C_{11} \left(\frac{w_{i+1,j}^n - w_{i,j}^n}{\Delta x} \right), & C_{12} \frac{\partial w}{\partial y} &= C_{12} \left(\frac{w_{i,j+1}^n - w_{i,j}^n}{\Delta y} \right) \\ C_{21} \frac{\partial z}{\partial x} &= C_{21} \left(\frac{z_{i+1,j}^n - z_{i,j}^n}{\Delta x} \right), & C_{22} \frac{\partial z}{\partial y} &= C_{22} \left(\frac{z_{i,j+1}^n - z_{i,j}^n}{\Delta y} \right) \\ D_{11} \frac{\partial^2 w}{\partial x^2} &= D_{11} \left(\frac{w_{i+1,j}^n - 2w_{i,j}^n + w_{i-1,j}^n}{\Delta x^2} \right), & D_{12} \frac{\partial^2 w}{\partial y^2} &= D_{12} \left(\frac{w_{i,j+1}^n - 2w_{i,j}^n + w_{i,j-1}^n}{\Delta y^2} \right), \\ D_{21} \frac{\partial^2 z}{\partial x^2} &= D_{21} \left(\frac{z_{i+1,j}^n - 2z_{i,j}^n + z_{i-1,j}^n}{\Delta x^2} \right), & D_{22} \frac{\partial^2 z}{\partial y^2} &= D_{22} \left(\frac{z_{i,j+1}^n - 2z_{i,j}^n + z_{i,j-1}^n}{\Delta y^2} \right), \\ \frac{r_w}{K_w} w(K_w - w + a_{12}z) &= \frac{r_w}{K_w} w_{i,j}^n (K_w - w_{i,j}^n + a_{12}z_{i,j}^n) \\ \frac{r_z}{K_z} z(K_z - z + a_{21}w) &= \frac{r_z}{K_z} z_{i,j}^n (K_z - z_{i,j}^n + a_{21}w_{i,j}^n). \end{aligned}$$

Taking $\Delta x = \Delta y = h$, $\Delta t = n$, the following system of Equations was obtained:

$$\begin{aligned} \frac{w_{i,j}^{n+1} - w_{i,j}^n}{\Delta t} &= D_{11} \left(\frac{w_{i+1,j}^n - 2w_{i,j}^n + w_{i-1,j}^n}{h^2} \right) + D_{12} \left(\frac{w_{i,j+1}^n - 2w_{i,j}^n + w_{i,j-1}^n}{h^2} \right) - C_{11} \left(\frac{w_{i+1,j}^n - w_{i,j}^n}{h} \right) - \\ &C_{12} \left(\frac{w_{i,j+1}^n - w_{i,j}^n}{h} \right) + \frac{r_w}{K_w} w_{i,j}^n (K_w - w_{i,j}^n + a_{12}z_{i,j}^n) - d_w w_{i,j}^n \end{aligned} \quad (43a)$$

$$\begin{aligned} \frac{z_{i,j}^{n+1} - z_{i,j}^n}{\Delta t} &= D_{21} \left(\frac{z_{i+1,j}^n - 2z_{i,j}^n + z_{i-1,j}^n}{h^2} \right) + D_{22} \left(\frac{z_{i,j+1}^n - 2z_{i,j}^n + z_{i,j-1}^n}{h^2} \right) - C_{21} \left(\frac{z_{i+1,j}^n - z_{i,j}^n}{h} \right) - \\ &C_{22} \left(\frac{z_{i,j+1}^n - z_{i,j}^n}{h} \right) + \frac{r_z}{K_z} z_{i,j}^n (K_z - z_{i,j}^n + a_{21}w_{i,j}^n) - d_z z_{i,j}^n \end{aligned} \quad (43b)$$

Parameters and their sources are shown in Table 1. Other parameters that are not easily obtained have been allowed to vary within corresponding possible intervals.

Table 1: Parameters of wildebeest and zebra with their sources

Parameter Description	Symbol	Value	Units	Source
Per capita intrinsic growth rate for wildebeest	r_w	1	individuals/year	Mduma (1996)
Per capita intrinsic growth rate for zebra	r_z	0.34	individuals/year	Fay and Greef (2006)
Carrying capacity for wildebeest	K_w	18×10^5	individuals	Assumed
Carrying capacity for zebra	K_z	3×10^4	individuals	Assumed
Mutual benefit for wildebeest	a_{12}	0.015	unitless	Fay and Greef (2006)
Mutual benefit for zebra	a_{21}	0.02	unitless	Fay and Greef (2006)
Wildebeest population	w	13×10^5	individuals	Grant (2006)
Zebra population	z	2.5×10^4	individuals	Grant (2006)
The mortality rate of wildebeest due to drought	β	0.015	individuals/year	Sinclair <i>et al.</i> (2008)
The mortality rate of zebra due to drought	π	0.015	individuals/year	Assumed
The mortality rate of wildebeest due to predation	α	0.03	individuals/year	Assumed
The mortality rate of zebra due to predation	ρ	0.06	individuals/year	Assumed

The simulations were done in a python computer program (the corresponding codes are attached in Appendix 2). The results and discussions are presented in Chapter Four.

3.6 Diffusive Prey-predator Relations between Wildebeest, Zebra and Lion

3.6.1 The Mathematical Model

This mathematical model extends the mutualism model (21) with one predator. In addition, this model extends the model by Fay and Greef (2006), who developed a two prey-one predator

mathematical model. However, their model had no migration and diffusion parameters. Furthermore, the analysis of the dynamical behaviour of the three species was not performed. Therefore, this study included the diffusion and migration movement parameters. A detailed analysis of the dynamic behaviour of the three species was carried out. The dispersion relation of wildebeests, zebras, and lions was carried out to analyse the effects of diffusion and migration on the system's stability. Finally, the numerical method of the system of equations was carried out.

(i) Assumptions of the Model

- There are three populations, namely, two preys (wildebeests and zebras) whose population densities are denoted by w and z , and the predator whose population density is denoted by P .
- In the absence of the predator, the prey population density grows according to the logistic law of growth.
- Two prey species are mutualistic.
- The predator needs sufficient handling time for the prey. These are incorporated using Holling type II functional response.
- There is a death rate for both prey species from factors such as drought.
- The rate of an increase in the predator population depends on the amount of prey biomass it converts as food.

(ii) Parameters of the Model

The following are the parameters used in the model:

- r_1 and r_2 are per capita intrinsic growth rates for prey w and z , respectively.
- K_1 and K_2 are carrying capacities for prey N_1 and N_2 respectively.
- b_1 and b_2 are coefficients for mutualistic relations between prey w and z , respectively.
- h_1 and h_2 are capturing rates of predator P on w and z , respectively.

- a_1/h_1 and a_2/h_2 are the predator's handling time on prey w and z , respectively.
- λ_1 and λ_2 are the mortality rates for prey w and z respectively due to drought, while c is the natural mortality of the predator P .
- d_1 and d_2 are coefficients that measure the predator's efficiency to convert prey biomass of w and z , respectively into fertility (reproductivity).
- The quantity D_p represents diffusion of lion. The velocity of lions flow is defined as $C_p = M_p \nabla P$, where M_p is a constant parameter expressing the flow intensity for lion.

Therefore, the resulting mathematical model that describes the interaction of the three species is given as:

$$\left. \begin{aligned} \frac{\partial w}{\partial t} &= D_w \nabla^2 w - C_z \cdot \nabla w + r_1 w \left(1 - \frac{w}{K_1}\right) + b_1 w z - \frac{h_1 w P}{1 + a_1 z} - \lambda_1 w \\ \frac{\partial z}{\partial t} &= D_z \nabla^2 z - C_z \cdot \nabla z + r_2 z \left(1 - \frac{z}{K_2}\right) + b_2 w z - \frac{h_2 z P}{1 + a_2 z} - \lambda_2 z \\ \frac{\partial P}{\partial t} &= D_p \nabla^2 P - C_p \cdot \nabla P + P \left(-c + d_1 \frac{h_1 w}{1 + a_1 w} + d_2 \frac{h_2 z}{1 + a_2 z}\right) \end{aligned} \right\} \quad (44)$$

The terms $\frac{h_1 w}{1 + a_1 w}$ and $\frac{h_2 z}{1 + a_2 z}$ are known as a Holling type II functional response.

3.6.2 Local Stability Analysis

The local stability analysis was conducted to see the local system dynamics. This is performed by letting the diffusive parameters $D_p = D_z = D_w = 0$ (Kurowski *et al.*, 2017).

$$\left. \begin{aligned} \frac{dw}{dt} &= r_1 w \left(1 - \frac{w}{K_1}\right) + b_1 w z - \frac{h_1 w P}{1 + a_1 w} - \lambda_1 w \\ \frac{dz}{dt} &= r_2 z \left(1 - \frac{z}{K_2}\right) + b_2 w z - \frac{h_2 z P}{1 + a_2 z} - \lambda_2 z \\ \frac{dP}{dt} &= P \left(-c + d_1 \frac{h_1 w}{1 + a_1 w} + d_2 \frac{h_2 z}{1 + a_2 z}\right) \end{aligned} \right\} \quad (45)$$

For ease of computations, the non-dimensionalization of the model is represented by Equations (45):

Let $X = a_1 w$, $Y = a_2 z$

This gives the following:

$$\left. \begin{aligned} \frac{dX}{dt} &= r_1 \left[X \left(1 - \frac{1}{a_1 k_1} X \right) + \frac{b_1}{a_2 r_1} XY - \frac{h_1 XP}{r_1(1+X)} - \lambda_1 X \right] \\ \frac{dY}{dt} &= r_2 \left[Y \left(1 - \frac{1}{a_2 K_2} Y \right) + \frac{b_2}{a_1 r_2} XY - \frac{h_2 YP}{r_2(1+Y)} - \lambda_2 Y \right] \\ \frac{dP}{dt} &= c \left[-P + \frac{d_1 h_1 XP}{c a_1 (1+X)} + \frac{d_2 h_2 YP}{c a_2 (1+Y)} \right] \end{aligned} \right\} \quad (46)$$

$$\text{Let } \beta_1 = \frac{1}{a_1 k_1}, \quad \gamma_1 = \frac{b_1}{a_2 r_1}, \quad \delta_1 = \frac{h_1}{r_1}, \quad \beta_2 = \frac{1}{a_2 K_2}, \quad \gamma_2 = \frac{b_2}{a_1 r_2}, \quad \delta_2 = \frac{h_2}{r_2},$$

$$\gamma_3 = \frac{d_1 h_1}{c a_1}, \quad \delta_3 = \frac{d_2 h_2}{c a_2}, \quad M_1 = \frac{\lambda_1}{r_1}, \quad M_2 = \frac{\lambda_2}{r_2}$$

Simplifying, the system of Equations (46) gives the following:

$$\left. \begin{aligned} \frac{dX}{dt} &= r_1 X \left[1 - \beta_1 X - M_1 + \gamma_1 Y - \delta_1 \frac{P}{(1+X)} \right] \\ \frac{dY}{dt} &= r_2 Y \left[1 - \beta_2 Y - M_2 + \gamma_2 X - \delta_2 \frac{P}{(1+Y)} \right] \\ \frac{dP}{dt} &= c P \left[-1 + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)} \right] \end{aligned} \right\} \quad (47)$$

3.6.3 Existence of Equilibrium Points in the System

The conditions for the existence of the equilibrium points of the system were established. By Equating (47) to zero, the system has seven possible nonnegative equilibria, namely:

$$E_0(0,0,0), \quad E_1(X^*, 0, 0), \quad E_2(0, Y^*, 0), \quad E_3(X^*, Y^*, 0), \quad E_4(X^*, 0, P^*), \quad E_5(0, Y^*, P^*)$$

And,

$$E_6(X^*, Y^*, P^*).$$

The existence of $E_0(0,0,0)$ is trivial. The existences of other equilibria are shown as follows:

Existence of $E_1(X^*, 0, 0)$ with $X^* > 0$.

Let $Y = 0$ and $P = 0$. From Equation (47), one obtains:

$$r_1 X (1 - \beta_1 X - M_1) = 0. \quad X^* = \frac{1-M_1}{\beta_1}.$$

Thus,

$E_1(X^*, 0, 0) = E_1\left(\frac{1-M_1}{\beta_1}, 0, 0\right)$. The equilibrium exists if $M_1 < 1$.

This condition implies that $\lambda_1 < r_1$.

Thus, in the absence of prey z and predator P , the death rate of prey w must be less than its intrinsic growth rate r_1 for equilibrium $E_1(X^*, 0, 0)$ to exist.

Existence of $E_2(0, Y^*, 0)$ with $Y^* > 0$.

Let $X = 0$ and $P = 0$. From Equation (47), the following is obtained:

$$r_2 Y(1 - \beta_2 Y - M_2) = 0, \quad Y^* = \frac{1-M_2}{\beta_2}.$$

Thus,

$$E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right). \text{ The equilibrium exists if } M_2 < 1.$$

This condition implies that $\lambda_2 < r_2$. Thus, in the absence of prey w and predator P , the death rate of prey z must be less than its intrinsic growth rate r_2 for the equilibrium $E_2(0, Y^*, 0)$ to occur.

Existence of $E_3(X^*, Y^*, 0)$, with $X^*, Y^* > 0$.

Let $P = 0$. From Equation (47), the following is obtained:

$$r_1 X(1 - \beta_1 X + \gamma_1 Y - M_1) = 0$$

$$r_2 Y(1 - \beta_2 Y + \gamma_2 X - M_2) = 0$$

$$Y^* = \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2}$$

$$X^* = \frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2}$$

Thus,

$$E_3(X^*, Y^*, 0) = E_3\left(\frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2}, \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2}, 0\right),$$

$$M_1, M_2 < 1, \quad \beta_1\beta_2 > \gamma_1\gamma_2.$$

This condition implies that:

$$\frac{1}{a_1 k_1} \frac{1}{a_2 K_2} > \frac{b_1}{a_2 r_1} \frac{b_2}{a_1 r_2}, \Rightarrow \frac{r_1}{K_1} \frac{r_2}{K_2} > b_1 b_2.$$

In the absence of a predator, there is mutualism among the species. This is because the species grow to their respective carrying capacities and since the interspecies competition between wildebeests and zebras is small, the two species will co-exist.

Existence of $E_4(X^*, 0, P^*)$, $X^* > 0$, $P^* > 0$.

Let $Y^* = 0$

$$r_1 X \left(1 - \beta_1 X - \delta_1 \frac{P}{(1+X)} - M_1 \right) = 0$$

$$cP \left(-1 + \gamma_3 \frac{X}{(1+X)} \right) = 0$$

$$X^* = \frac{1}{\gamma_3 - 1} \text{ and } P^* = \left(1 + \frac{1}{(\gamma_3 - 1)} \right) \left(\frac{1 - M_1}{\delta_1} - \frac{\beta_1}{\delta_1 (\gamma_3 - 1)} \right) \text{ provided } M_1 < 1, \delta_1 > 0, \gamma_3 > 1.$$

From $\gamma_3 > c \Rightarrow d_1 > c \frac{a_1}{h_1}$, the proportion of biomass d_1 of the prey w converted into food by the predator P must be greater than the product of the predator's natural mortality rate, c and the time it takes to handle the prey, $\frac{a_1}{h_1}$.

Existence of $E_5(0, Y^*, P^*)$, with $Y^* > 0$ and $P^* > 0$.

Let $X = 0$, then

$$r_2 Y (1 - \beta_2 Y - M_2) - \delta_2 \frac{YP}{(1+Y)} = 0$$

$$cP(-1 + \delta_3 \frac{Y}{(1+Y)}) = 0.$$

Solving gives:

$$Y^* = \frac{1}{\delta_3 - 1} \text{ and } P^* = \left(1 + \frac{1}{(\delta_3 - 1)} \right) \left(\frac{1 - M_2}{\delta_2} - \frac{\beta_2}{\delta_2 (\delta_3 - 1)} \right) \text{ with } M_2 < 1, \delta_3 > 0, \delta_3 > 1.$$

From $\delta_3 > 1 \Rightarrow d_2 > c \frac{a_2}{h_2}$, the proportion of biomass d_2 of the prey z converted into food by the predator P must be greater than the product of the predator's natural mortality rate, c and the time it takes to handle the prey, $\frac{a_2}{h_2}$.

Co-existence equilibrium point: $E_6(X^*, Y^*, P^*)$.

Equating Equations (47) to zero and from this, two functions of $f(X, Y)$ and $g(X, Y)$ can be obtained which intersect at the equilibrium point $E_6(X^*, Y^*, P^*)$. Equating Equation (47) to zero, the following relations are obtained:

$$(1 - \beta_1 X - M_1) + \gamma_1 Y - \delta_1 \frac{P}{(1+X)} = 0 \quad (47a)$$

$$(1 - \beta_2 Y - M_2) - \gamma_2 X - \delta_2 \frac{P}{(1+Y)} = 0 \quad (47b)$$

$$-1 + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)} = 0 \quad (47c)$$

$$\text{From (47a) } P = \frac{(1+X)(1-\beta_1 X - M_1 + \gamma_1 Y)}{\delta_1} \quad (47d)$$

$$\text{From (47b) } P = \frac{(1+Y)(1-\beta_2 Y - M_2 + \gamma_2 X)}{\delta_2} \quad (47e)$$

$$\text{From (47c), } f(X, Y) = -1 + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)} = 0 \quad (47f)$$

From (47d) and (47e)

$$g(X, Y) = \frac{(1+X)(1-\beta_1 X - M_1 + \gamma_1 Y)}{\delta_1} - \frac{(1+Y)(1-\beta_2 Y - M_2 + \gamma_2 X)}{\delta_2} = 0 \quad (47g)$$

Equations (47f) and (47g) are two functions of X and Y . To prove the existence of $E_6(X^*, Y^*, P^*)$, conditions under which $f(X, Y)$ and $g(X, Y)$ meet in the interior of positive (X, Y) plane at the point (X^*, Y^*) are found. Knowing (X^*, Y^*) , P^* can be obtained from (47d). From equation (47f) as $X \rightarrow 0$, Y tends to Y_f given by $Y_f = \frac{1}{\delta_3 - 1}$ where $\delta_3 - 1 > 0$. It can be noticed that Y_f is the same as Y^* of $E_5(0, Y^*, P^*)$.

From (47g) as $X \rightarrow 0$, Y tends to Y_g given by $Y_g = \frac{-D_2 + \sqrt{D_2^2 - 4D_1 D_3}}{2D_1}$ where:

$$D_1 = \delta_1\beta_2, D_2 = \delta_2\gamma_1 + \delta_1\beta_2 + \delta_1M_2 - \delta_1, D_3 = \delta_2(1 - M_1) - \delta_1(1 - M_2)$$

Since the points $M_1 < 1$ and $M_2 < 1$, then the point Y_g is positive and real if,

$$\delta_2(1 - M_1) - \delta_1(1 - M_2) < 0.$$

Therefore, Y_f and Y_g are points at which the functions $f(X, Y)$ and $g(X, Y)$ would cut the Y axis in the (X, Y) plane respectively.

From Equation (47f) $\frac{dY}{dX} = \frac{\partial f}{\partial X} / \frac{\partial f}{\partial Y}$ where $\frac{\partial f}{\partial X} = \frac{\gamma_3}{(1+X)^2}$ and $\frac{\partial f}{\partial Y} = \frac{\delta_3}{(1+Y)^2}$. It is noted that $\frac{dY}{dX} > 0$.

Similarly, $\frac{dY}{dX} = \frac{\partial g}{\partial X} / \frac{\partial g}{\partial Y}$, where $\frac{\partial g}{\partial X} = \frac{-\beta_1\delta_2(1+2X)+\delta_2(1-M_1)+\delta_1\gamma_2(1+Y)+\gamma_1\delta_2Y}{\delta_1\delta_2}$ and

$$\frac{\partial g}{\partial Y} = \frac{\gamma_1\delta_2(1+X)-\beta_2\delta_1(1+2Y)+\delta_1(1-M_2)+\gamma_2\delta_1X}{\delta_1\delta_2}.$$

Therefore, $\frac{dY}{dX} < 0$ provided $\frac{\partial g}{\partial X} < 0$ and $\frac{\partial g}{\partial Y} > 0$.

Since $f(X, Y)$ it can be obtained $\frac{dY}{dX} > 0$ and $g(X, Y)$ which gives $\frac{dY}{dX} < 0$, then $f(X, Y)$ and $g(X, Y)$ will meet if $Y_f < Y_g$.

The existence of the positive equilibrium point $E_6(X^*, Y^*, P^*)$ is stated in the following theorem:

Theorem 2: *The positive equilibrium point $E_6(X^*, Y^*, P^*)$ will exist if for $M_1 < 1$ and $M_2 < 1$ then the condition $\delta_3 - 1 > 0$ and $Y_f < Y_g$ must be satisfied.*

In terms of the original parameters $M_1 < 1$ and $M_2 < 1$ implies that $\lambda_1 < r_1$ and $\lambda_2 < r_2$. The mortality rate of wildebeests and zebras should be less than their respective intrinsic growth rates. For the condition $\delta_3 - 1 > 0 \Rightarrow d_2 > c \frac{a_2}{h_2}$, the proportion of biomass d_2 of the prey z (zebra) converted into food by the predator P (lion) must be greater than the product of the predator's mortality rate, c and the time it takes to handle the prey, $\frac{a_2}{h_2}$.

3.6.4 Local Stability of the Equilibrium Points

The local asymptotic stability of each equilibrium point is studied by computing the Jacobian matrix and finding the eigenvalues evaluated at each equilibrium point. For stability of the

equilibrium points, the real parts of the eigenvalues of the Jacobian matrix must be negative.

From Equations (47), the Jacobian matrix of the system is given by:

$$J(E_i) = \begin{pmatrix} \frac{\partial f_1}{\partial X} & \frac{\partial f_1}{\partial Y} & \frac{\partial f_1}{\partial P} \\ \frac{\partial f_2}{\partial X} & \frac{\partial f_2}{\partial Y} & \frac{\partial f_2}{\partial Z} \\ \frac{\partial f_3}{\partial X} & \frac{\partial f_3}{\partial Y} & \frac{\partial f_3}{\partial Z} \end{pmatrix} \quad (48)$$

For $i = 0, 1, 2, 3, 4, 5, 6$. This gives:

$$J(E_i) = \begin{pmatrix} A^{**} & \gamma_1 X & \frac{-\delta_1 X}{(1+X)} \\ \gamma_2 Y & B^{**} & \frac{-\delta_2 Y}{(1+Y)} \\ \frac{\gamma_3 P}{(1+X)^2} & \frac{\delta_2 P}{(1+Y)^2} & C^{**} \end{pmatrix} \quad (49)$$

Where $A^{**} = 1 - 2\beta_1 X - M_1 + \gamma_1 Y - \delta_1 \frac{P}{(1+X)^2}$, $B^{**} = 1 - 2\beta_2 Y - M_2 + \gamma_2 X - \delta_2 \frac{P}{(1+Y)^2}$ and,

$$C^{**} = -c + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)}$$

The local asymptotic stability for each equilibrium point is analysed as below:

$E_0(0, 0, 0, 0)$. The Jacobian matrix evaluated at E_0 gives:

$$J(E_0) = \begin{pmatrix} 1 - M_1 & 0 & 0 \\ 0 & 1 - M_2 & 0 \\ 0 & 0 & -c \end{pmatrix}$$

The eigenvalues of the $J(E_0)$ are $1 - M_1$, $1 - M_2$ and $-c$. Since $M_1 < 1$ and $M_2 < 1$ implies that $1 - M_1 > 0$ and $1 - M_2 > 0$. Hence a saddle point (Unstable) is found.

Theorem 3: E_0 is always a saddle-node, and there cannot be total system extinction (47) for favourable initial conditions.

$E_1(X^*, 0, 0) = E_1\left(\frac{1-M_1}{\beta_1}, 0, 0\right)$. The Jacobian matrix of E_1 gives:

$$J(E_1) = \begin{pmatrix} (M_1 - 1) & \frac{\gamma_1(1-M_1)}{\beta_1} & \frac{-\delta_1(1-M_1)}{1-M_1+\beta_1} \\ 0 & (1-M_2) + \frac{\gamma_2(1-M_1)}{\beta_1} & 0 \\ 0 & 0 & -c + \frac{\gamma_3(1-M_1)}{\beta_1+1-M_1} \end{pmatrix}$$

The eigenvalues of the $J(E_1)$ are $(M_1 - 1)$, $(1 - M_2) + \frac{\gamma_2(1-M_1)}{\beta_1}$ and $-c + \frac{\gamma_3(1-M_1)}{\beta_1+1-M_1}$

Since $M_1 < 1$, $M_2 < 1$, then $(M_1 - 1) < 0$, $(1 - M_2) + \frac{\gamma_2(1-M_1)}{\beta_1} > 0$ and

$$-c + \frac{\gamma_3(1-M_1)}{\beta_1+1-M_1} < 0 \text{ if } \beta_1 > M_1 - 1.$$

Therefore, $E_1(X^*, 0, 0) = E_1\left(\frac{1-M_1}{\beta_1}, 0, 0\right)$ is a saddle point hence unstable.

$E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right)$. The Jacobian matrix of E_2 gives:

$$J(E_2) = \begin{pmatrix} 1 - M_1 + \frac{\gamma_1(1-M_2)}{\beta_2} & 0 & 0 \\ \frac{\gamma_2(1-M_2)}{\beta_2} & M_2 - 1 & \frac{-\delta_2(1-M_2)}{1-M_2+\beta_2} \\ 0 & 0 & -c + \frac{\delta_3(1-M_2)}{\beta_2+1-M_2} \end{pmatrix}$$

The eigenvalues of the $J(E_2)$ are $1 - M_1 + \frac{\gamma_1(1-M_2)}{\beta_2}$, $M_2 - 1$ and $-c + \frac{\delta_3(1-M_2)}{\beta_2+1-M_2}$.

Since $M_1 < 1$, $M_2 < 1$, then $1 - M_1 + \frac{\gamma_1(1-M_2)}{\beta_2} > 0$, $M_2 - 1 < 0$ and

$$-c + \frac{\delta_3(1-M_2)}{\beta_2+1-M_2} < 0 \text{ if } \beta_2 < M_2 - 1.$$

Therefore $E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right)$ is a saddle point hence unstable.

$E_3(X^*, Y^*, 0) = E_3\left(\frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}, \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}, 0\right)$. The Jacobian matrix of E_3 gives:

$$J(E_3) = \begin{pmatrix} D^{**} & W^{**} & E^{**} \\ Z^{**} & F^{**} & G^{**} \\ 0 & 0 & H^{**} \end{pmatrix}$$

$$\text{Where } D^{**} = 1 - M_1 - 2\beta_1 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2} + \gamma_1 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}$$

$$W^{**} = \gamma_1 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}$$

$$E^{**} = -\delta_1 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{(\beta_1\beta_2-\gamma_1\gamma_2+\beta_2(1-M_1)+\gamma_1(1-M_2))}$$

$$F^{**} = 1 - M_2 - 2\beta_2 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2} + \gamma_2 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}$$

$$Z^{**} = \gamma_2 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}$$

$$G^{**} = -\delta_2 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{(\beta_1\beta_2-\gamma_1\gamma_2+\beta_1(1-M_2)+\gamma_2(1-M_1))}$$

$$H^{**} = -c + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)}, \text{ substituting gives:}$$

$$H^{**} = -c + \gamma_3 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{(\beta_1\beta_2-\gamma_1\gamma_2+\beta_2(1-M_1)+\gamma_1(1-M_2))} + \delta_3 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{(\beta_1\beta_2-\gamma_1\gamma_2+\beta_1(1-M_2)+\gamma_2(1-M_1))}$$

The eigenvalues of the $J(E_3)$ are obtained by solving the system:

$$\det \begin{pmatrix} D^{**} - \lambda & W^{**} & E^{**} \\ Z^{**} & F^{**} - \lambda & G^{**} \\ 0 & 0 & H^{**} - \lambda \end{pmatrix} = 0. \text{ This gives:}$$

$$(H^{**} - \lambda)[(F^{**} - \lambda)(D^{**} - \lambda) - Z^{**}W^{**}] = 0 \text{ which simplifies the characteristic Equation}$$

$$\lambda^3 - (D + F + H)\lambda^2 + (DF + DH + FH - ZW)\lambda + ZWH - DFH = 0 \text{ which is of the form}$$

$$\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0.$$

By Routh-Hurwitz criteria (Murray, 1989) the λ 's are negative if $a_1 > 0$, $a_3 > 0$ and

$$a_1a_2 - a_3 > 0.$$

Each of these conditions is considered next as follows:

$$(i) \quad a_1 > 0 \Rightarrow -(D + F + H) > 0 \text{ or } +F + H < 0.$$

$$D + F + H < 0. \text{ This can be satisfied if } H^{**} < 0, D^{**} < 0, F^{**} < 0$$

$$\text{For } H^{**} < 0 \text{ implies } -c + \gamma_3 \frac{X}{(1+X)} + \delta_3 \frac{Y}{(1+Y)} < 0. \text{ Simplifying gives:}$$

$$XY(\gamma_3 + \delta_3 - c) + (\delta_3 - c)Y + (\gamma_3 - c)X - c < 0.$$

This will hold if, for $M_1 < 1$, $M_2 < 1$, $\gamma_3 + \delta_3 < c$.

In terms of original parameters $\gamma_3 + \delta_3 < c$ gives $\frac{d_1 h_1}{a_1} + \frac{d_2 h_2}{a_2} < c$.

This implies that the predator's efficiency in converting the biomass of both preys into fertility or reproductivity must be less than the predator's natural mortality rate and the time it takes to handle both preys.

For $D^{**} < 0$ implies:

$1 - 2\beta_1 X - M_1 + \gamma_1 Y < 0$. Substituting for X^* and Y^* gives:

$$1 - M_1 - 2\beta_1 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2} + \gamma_1 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}.$$

$$\frac{\beta_1\beta_2(M_1-1)-\beta_1\gamma_1(M_2-1)}{\beta_1\beta_2-\gamma_1\gamma_2} < 0.$$

Therefore,

$$D^{**} < 0 \text{ if } M_1 < 1, M_2 < 1, \beta_1\beta_2 > \gamma_1\gamma_2.$$

For $F^{**} < 0$,

$1 - 2\beta_2 Y - M_2 + \gamma_2 X < 0$ Substituting for X^* and Y^* the following equation was obtained:

$$1 - M_2 - 2\beta_2 \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2} + \gamma_2 \frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}.$$

$$\frac{\beta_1\beta_2(M_2-1)-\beta_2\gamma_2(M_1-1)}{\beta_1\beta_2-\gamma_1\gamma_2} < 0.$$

Therefore,

$$F^{**} < 0 \text{ if } M_1 < 1, M_2 < 1, \beta_1\beta_2 > \gamma_1\gamma_2$$

(ii) $a_3 > 0$ Implies $H(ZW - DF) > 0$. This is satisfied if $H < 0$, $ZW - DF < 0$, which gives:

$$\begin{aligned}
ZW - DF &= \gamma_2 \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2} \gamma_1 \frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2} \\
&\quad - \left(1 - M_1 - 2\beta_1 \frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2} + \gamma_1 \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2} \right) \left(1 \right. \\
&\quad \left. - M_2 - 2\beta_2 \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2} + \gamma_2 \frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2} \right) < 0
\end{aligned}$$

Let $L_1 = \frac{\beta_1(1-M_2) + \gamma_2(1-M_1)}{\beta_1\beta_2 - \gamma_1\gamma_2}$, $L_2 = \frac{\beta_2(1-M_1) + \gamma_1(1-M_2)}{\beta_1\beta_2 - \gamma_1\gamma_2}$

The inequality $\gamma_1\gamma_2 L_1 L_2 - (1 - M_1 - 2\beta_1 L_2 + \gamma_1 L_2)(1 - M_2 - 2\beta_2 L_1 + \gamma_2 L_2) < 0$ is obtained.

Therefore,

$$ZW - DF < 0 \text{ if } M_1 < 1, M_2 < 1, \beta_1\beta_2 > \gamma_1\gamma_2.$$

(iii) $a_1 a_2 - a_3 > 0$ implies $-(D + F + H)(DF + DH + FH - ZW) - H(ZW - DF) > 0$

This simplifies to $(D + F)[(ZW - DF) - H(D + F + H)] > 0$.

This is satisfied if $D < 0, W < 0, H < 0$ and $(ZW - DF) < 0$ which have been prior established.

Therefore,

$E_3(X^*, Y^*, 0)$ is locally asymptotically stable if the following conditions are satisfied:

$$M_1 < 1, M_2 < 1, \beta_1\beta_2 > \gamma_1\gamma_2$$

$$E_4(X^*, 0, P^*) = \left(\frac{c}{\gamma_3 - c}, 0, \left(1 + \frac{c}{(\gamma_3 - c)} \right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)} \right) \right)$$

The Jacobian Matrix evaluated at E_4 give:

$$J(E_4) = \begin{pmatrix} E^+ & \frac{\gamma_1 c}{\gamma_3 - c} & \frac{-\delta_1 c}{\gamma_3} \\ 0 & F^+ & 0 \\ G^+ & H^+ & 0 \end{pmatrix}$$

$$E^+ = 1 - 2\beta_1 \frac{c}{\gamma_3 - c} - M_1 - \delta_1 \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right) \frac{(\gamma_3 - c)^2}{\gamma_3^2},$$

$$F^+ = 1 - M_2 + \gamma_2 \frac{c}{\gamma_3 - c} - \delta_2 \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right) \text{ and}$$

$$G^+ = \frac{(\gamma_3 - c)^2}{\gamma_3} \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right),$$

$$H^+ = \frac{\delta_2 P}{(1 + \gamma)^2} = \delta_2 \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right),$$

The eigenvalues of the matrix $J(E_4)$ have negative real parts if $E^+ < 0$ and $F^+ < 0$. Further $E^+ < 0$ implies:

$$1 - 2\beta_1 \frac{c}{\gamma_3 - c} - M_1 - \delta_1 \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right) \frac{(\gamma_3 - c)^2}{\gamma_3^2} < 0.$$

Simplifying gives:

$$\frac{(1 - M_1)(\gamma_3 - c) - 2\beta_1 c}{(\gamma_3 - c)} - \left[\frac{(1 - M_1)(\gamma_3 - c) - \beta_1 c}{\gamma_3} \right] < 0$$

$$\text{Also } F^+ < 0 \text{ implies } 1 - M_2 + \gamma_2 \frac{c}{\gamma_3 - c} - \delta_2 \left(1 + \frac{c}{(\gamma_3 - c)}\right) \left(\frac{1 - M_1}{\delta_1} - \frac{c\beta_1}{\delta_1(\gamma_3 - c)}\right) < 0$$

$$(1 - M_1)(\gamma_3 - c) + \gamma_2 c - \delta_2 \gamma_3 \left[\frac{(1 - M_1)(\gamma_3 - c) - \beta_1 c}{\delta_1(\gamma_3 - c)} \right] < 0$$

Therefore, $E_4(X^*, 0, P^*)$ is locally asymptotically stable if conditions $M_1 < 1$, $M_2 < 1$, $\gamma_3 < c$ are satisfied.

$$E_5(0, Y^*, P^*) = \left(0, \frac{c}{\delta_3 - c}, \left(1 + \frac{c}{(\delta_3 - c)}\right) \left(\frac{1 - M_2}{\delta_2} - \frac{c\beta_2}{\delta_2(\delta_3 - c)}\right)\right). \text{ The Jacobian matrix of } E_5 \text{ is}$$

$$J(E_5) = \begin{pmatrix} A^+ & 0 & 0 \\ P^+ & B^+ & Q^+ \\ D^+ & E^+ & 0 \end{pmatrix}$$

Where,

$$A^+ = (1 - M_1)(\delta_3 - c) + \gamma_1 c - \frac{\delta_1 \delta_3}{\delta_2(\delta_3 - c)} \left((1 - M_2)(\delta_3 - c) - c\beta_2 \right),$$

$$B^+ = (1 - M_2)(\delta_3 - c) - 2\beta_2 c - \left(\frac{(1 - M_2)(\delta_3 - c) - \beta_2 c}{\delta_3} \right)$$

$$D^+ = \frac{\gamma_3 \delta_3}{(\delta_3 - c)} \frac{(1 - M_2)(\delta_3 - c) - \beta_2 c}{\delta_2(\delta_3 - c)}, \quad P^+ = \frac{c\gamma_2}{\delta_3 - c}, \quad Q^+ = \frac{-c\delta_2}{\delta_3}$$

$$E^+ = \left(\frac{(1 - M_2)(\delta_3 - c) - \beta_2 c}{\delta_3^2} \right) (\delta_3 - c)$$

The eigenvalues of the matrix $J(E_5)$ are $\det \begin{pmatrix} A^+ - \lambda & 0 & 0 \\ P^+ & B^+ - \lambda & Q^+ \\ D^+ & E^+ & 0 - \lambda \end{pmatrix} = 0$

Simplifying the following characteristic Equation was obtained:

$$\lambda^3 - (A + B)\lambda^2 + (AB - EQ)\lambda + AEQ = 0 \text{ which is of the form } \lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0.$$

By Routh-Hurwitz criteria (Murray, 1989), the λ 's are negative if $a_1 > 0$, $a_3 > 0$ and

$a_1 a_2 - a_3 > 0$. Each of these conditions is considered next as follows:

If $a_1 > 0 \Rightarrow -(A^+ + B^+) > 0$ or $A^+ + B^+ < 0$

$A^+ + B^+ < 0$. This can be satisfied if $A^+ < 0$, $B^+ < 0$,

If $A^+ < 0$ implies $1 - M_1 + \gamma_1 Y - \delta_1 P < 0$.

This will hold if, for $M_1 < 1$, $M_2 < 1$, $\delta_3 < c$

If $B^+ < 0$ implies This will hold if, for $M_1 < 1$, $M_2 < 1$, $\delta_3 < c$

If $a_3 > 0$ implies $AEQ > 0$. Since $A^+ < 0$, $Q^+ = \frac{-c\delta_2}{\delta_3} < 0$ and

$$E^+ = \left(\frac{(1 - M_2)(\delta_3 - c) - \beta_2 c}{\delta_3^2} \right) (\delta_3 - c) > 0 \text{ if } M_1 < 1, M_2 < 1, \delta_3 < c$$

If $a_1 a_2 - a_3 > 0$ implies that $-(A^+ + B^+)(AB - EQ) - AEQ > 0$

This simplifies to $(A^+ + B^+) - B^+ E^+ D^+ < 0$.

Hence $A^+ < 0$, $B^+ < 0$, $D^+ < 0$ and $E^+ > 0$

Therefore,

$E_5(0, Y^*, P^*)$ is locally asymptotically stable if conditions $M_1 < 1$, $M_2 < 1$, $\delta_3 < c$ are satisfied.

3.6.5 Global Stability of the Steady States

The global stability of the equilibrium points was analysed as follows:

(i) Global Stability of E_1 , E_2 and E_3

The global stability of E_1 , E_2 and E_3 was analysed by transforming the system of Equations (47) into a linear system and then choosing a suitable Lyapunov function to analyse each equilibrium point.

By letting $X = X^* + x$, $Y = Y^* + y$ and $P = P^* + p$, where x, y and p , are small perturbations about X^* , Y^* and P^* respectively. The system of Equations (47) is turned into a linear system of the form $\dot{x}_i = J(E_i)x_i$, where $J(E_i)x_i$ is the Jacobian Matrix of the system of Equations (47). Thus the system of Equations (47) is:

$$\left. \begin{aligned} \frac{dx}{dt} &= \left(-\beta_1 X^* - \delta_1 \frac{P^* X^*}{(1+X^*)^2} \right) x + (\gamma_1 X^*) y - \left(\frac{\delta_1 X^*}{(1+X^*)} \right) p \\ \frac{dy}{dt} &= (\gamma_2 Y^*) x + \left(-\beta_2 Y^* - \delta_2 \frac{P^* Y^*}{(1+Y^*)^2} \right) y - \left(\frac{\delta_2 Y^*}{(1+Y^*)} \right) p \\ \frac{dp}{dt} &= \left(\frac{\gamma_3 P^*}{(1+X^*)^2} \right) x + \left(\frac{\delta_2 P^*}{(1+Y^*)^2} \right) y + (0)p \end{aligned} \right\} \quad (50)$$

(ii) Global Stability of $E_1(X^*, 0, 0) = \left(\frac{1-M_1}{\beta_1}, 0, 0 \right)$

The Lyapunov function can be defined as $V(x, y, p) = \frac{x^2}{2X^*} + \frac{y^2}{2} + \frac{p^2}{2}$ where X^* is the component of the equilibrium point $E_1(X^*, 0, 0) = \left(\frac{1-M_1}{\beta_1}, 0, 0 \right)$. It is clear that $V(x, y, p)$ is a positive definite function. Differentiating V with respect to t gives:

$V'(x, y, p) = \frac{x}{X^*} \dot{x} + y\dot{y} + p\dot{p}$. Substituting for \dot{x} , \dot{y} and \dot{p} using (50) gives:

$$V'(x, y, p) = x \left[\left(-\beta_1 - \delta_1 \frac{P^*}{(1+X^*)^2} \right) x + (\gamma_1)y - \left(\frac{\delta_1}{(1+X^*)} \right) p \right] + yY^* \left[(\gamma_2)x + \left(-\beta_2 - \delta_2 \frac{P^*}{(1+Y^*)^2} \right) y - \left(\frac{\delta_2}{(1+Y^*)} \right) p \right] + pP^* \left[\left(\frac{\gamma_3}{(1+X^*)^2} \right) x + \left(\frac{\delta_2}{(1+Y^*)^2} \right) y + (0)p \right]$$

For $E_1(X^*, 0, 0)$, implies $V'(x, y, p) = -\beta_1 x^2 + \gamma_1 xy - \frac{\delta_1}{(1+X^*)} xp$

Therefore, $E_1(X^*, 0, 0)$ is Lyapunov stable if $V'(x, y, p) = 0$ and uniformly asymptotically stable if $V'(x, y, p) < 0$.

Thus, $E_1(X^*, 0, 0)$ is globally asymptotically stable if $M_1 < 1$. In terms of original parameters, this condition implies that $\lambda_1 < r_1$ in the absence of Zebra (z) and lion (P), the population of Serengeti wildebeests (w) is globally stable, provided that the death rate of wildebeests must be less than its intrinsic growth rate r_1 .

(iii) Global stability of $E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right)$

Defines a Lyapunov Function $V(x, y, p) = \frac{x^2}{2} + \frac{y^2}{2Y^*} + \frac{p^2}{2}$ where Y^* is the component of the equilibrium point $E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right)$. Clearly, $V(x, y, p)$ is a positive definite function. Differentiating V with respect to t gives:

$V'(x, y, p) = x\dot{x} + \frac{y}{Y^*}\dot{y} + p\dot{p}$. Substituting for \dot{x} , \dot{y} and \dot{p} using (47) to obtain

$$V'(x, y, p) = xX^* \left[\left(-\beta_1 - \delta_1 \frac{P^*}{(1+X^*)^2} \right) x + (\gamma_1)y - \left(\frac{\delta_1}{(1+X^*)} \right) p \right] + y \left[(\gamma_2)x + \left(-\beta_2 - \delta_2 \frac{P^*}{(1+Y^*)^2} \right) y - \left(\frac{\delta_2}{(1+Y^*)} \right) p \right] + pP^* \left[\left(\frac{\gamma_3}{(1+X^*)^2} \right) x + \left(\frac{\delta_2}{(1+Y^*)^2} \right) y + (0)p \right]$$

$E_2(0, Y^*, 0) = E_2\left(0, \frac{1-M_2}{\beta_2}, 0\right)$, implies $V'(x, y, p) = \gamma_2 xy - \beta_2 y^2 - \delta_2 py$.

Therefore,

$E_1(X^*, 0, 0)$ is Lyapunov stable if $V'(x, y, p) = 0$ and uniformly asymptotically stable if $V'(x, y, p) < 0$.

Thus, $E_2(0, Y^*, 0)$ is globally asymptotically stable if $M_2 < 1$. In terms of original parameters, this condition implies that $\lambda_2 < r_2$ in the absence of the wildebeest (w) and the lion (P), the

population of Serengeti Zebras (z) is globally stable, provided that the death rate of Zebras must be less than its intrinsic growth rate r_2 .

(iv) **Global Stability of** $E_3(X^*, Y^*, 0) = E_3\left(\frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}, \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}, 0\right)$

The Lyapunov function can be defined as $V(x, y, p) = \frac{x^2}{2X^*} + \frac{y^2}{2Y^*} + \frac{p^2}{2}$ where Y^* and X^* are the components of the equilibrium point:

$$E_3(X^*, Y^*, 0) = E_3\left(\frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}, \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}, 0\right)$$

$V(x, y, p)$ is a positive definite function. Differentiating V with respect to t one gets:

$$V'(x, y, p) = \frac{x}{X^*} \dot{x} + \frac{y}{Y^*} \dot{y} + p\dot{p}.$$

Substituting for \dot{x} , \dot{y} and \dot{p} using (50) gives:

$$V'(x, y, p) = x \left[\left(-\beta_1 - \delta_1 \frac{P^*}{(1+X^*)^2} \right) x + (\gamma_1)y - \left(\frac{\delta_1}{(1+X^*)} \right) p \right] + y \left[(\gamma_2)x + \left(-\beta_2 - \delta_2 \frac{P^*}{(1+Y^*)^2} \right) y - \left(\frac{\delta_2}{(1+Y^*)} \right) p \right] + pP^* \left[\left(\frac{\gamma_3}{(1+X^*)^2} \right) x + \left(\frac{\delta_2}{(1+Y^*)^2} \right) y + (0)p \right]$$

$$E_3(X^*, Y^*, 0) = E_3\left(\frac{\beta_2(1-M_1)+\gamma_1(1-M_2)}{\beta_1\beta_2-\gamma_1\gamma_2}, \frac{\beta_1(1-M_2)+\gamma_2(1-M_1)}{\beta_1\beta_2-\gamma_1\gamma_2}, 0\right), \text{ this implies that:}$$

$$V'(x, y, p) = -\beta_1 x^2 + (\gamma_1 + \gamma_2)xy - \beta_2 y^2 - \frac{\delta_1}{(1+X^*)} xp - \frac{\delta_2}{(1+Y^*)} py$$

This is globally stable if the conditions $M_1 < 1$, $M_2 < 1$ and $\beta_1\beta_2 > \gamma_1\gamma_2$

In terms of original parameters $M_1 < 1$ implies $\lambda_1 < r_1$, $M_2 < 1$ implies that $\lambda_2 < r_2$ and

$$\beta_1\beta_2 > \gamma_1\gamma_2 \text{ implies } \frac{1}{a_1 k_1} \frac{1}{a_2 K_2} > \frac{b_1}{a_2 r_1} \frac{b_2}{a_1 r_2}, \Rightarrow \frac{r_1}{K_1} \frac{r_2}{K_2} > b_1 b_2.$$

The intrinsic growth rates of the Serengeti wildebeests and Zebras are greater than their death rates. Furthermore, due to mutualism among the species and the absence of a predator, the two species grow to their respective carrying capacities, and the two species will co-exist.

(v) **Global Stability of E_4 and E_5**

To prove the global stability of E_4 and E_5 by using Bendixson-Dulac's criteria and finding conditions for the nonexistence of periodic orbits within the positive plane containing each equilibrium point as in Dubey and Upadhyay (2004) and Castillo-Chavez and Brauer (1999).

For E_4 , defines a continuously differentiable function in the $X > 0, P > 0$ planes as $H_1(X, P) = \frac{1}{XP^2}$. The system of Equations (47) gives:

$$h_1(X, P) = X \left(1 - \beta_1 X - M_1 - \delta_1 \frac{P}{(1+X)} \right) \text{ and } h_2(X, P) = P \left(-c + \gamma_3 \frac{X}{(1+X)} \right)$$

This gives:

$$H_1 h_1 = \frac{1}{P^2} \left(1 - \beta_1 X - M_1 - \delta_1 \frac{P}{(1+X)} \right) \text{ and } H_1 h_2 = \frac{1}{XP} \left(-c + \gamma_3 \frac{X}{(1+X)} \right)$$

$$\text{Computing } \frac{\partial H_1 h_1}{\partial X} \text{ and } \frac{\partial H_2 h_2}{\partial P} \text{ give } \frac{\partial H_1 h_1}{\partial X} = \frac{1}{P^2} \left(-\beta_1 + \delta_1 \frac{P}{(1+X)^2} \right) \text{ and}$$

$$\frac{\partial H_1 h_2}{\partial P} = -\frac{1}{XP^2} \left(-c + \gamma_3 \frac{X}{(1+X)} \right). \text{ From this one obtains:}$$

$$\frac{\partial H_1 h_1}{\partial X} + \frac{\partial H_2 h_2}{\partial P} = \frac{1}{P^2} \left[\frac{c - \beta_1 X}{X} + \frac{\delta_1 P - \gamma_3 - \gamma_3 X}{(1+X)^2} \right] \text{ which is negative in the plane } X > 0, P > 0 \text{ in the region } \Omega_1 = \left\{ 0 < \frac{c}{\beta_1} < X, \quad 0 < P < \frac{\gamma_3}{\delta_1} \right\}.$$

Therefore,

by Bendixson-Dulac criteria: $E_4(X, P)$ is globally asymptotically stable in Ω_1 .

For E_5 , define a continuously differentiable function in the $Y > 0, P > 0$ planes as $H_2(Y, P) = \frac{1}{YP^2}$.

From the system of Equations (47), it can be obtained:

$$J_1(Y, P) = Y \left(1 - \beta_2 Y - M_2 - \delta_2 \frac{P}{(1+Y)} \right) \text{ and } J_2(Y, P) = P \left(-c + \delta_3 \frac{Y}{(1+Y)} \right)$$

This gives:

$$H_2J_1 = \frac{1}{P^2} \left(1 - \beta_2 Y - M_2 - \delta_2 \frac{P}{(1+Y)} \right) \text{ and } H_2J_2 = \frac{1}{YP} \left(-c + \delta_3 \frac{Y}{(1+Y)} \right) \text{ I compute } \frac{\partial H_2J_1}{\partial Y} \text{ and } \frac{\partial H_2J_2}{\partial P} \text{ which gives } \frac{\partial H_2J_1}{\partial Y} = \frac{1}{P^2} \left(-\beta_2 + \delta_2 \frac{P}{(1+Y)^2} \right) \text{ and}$$

$$\frac{\partial H_2J_2}{\partial P} = -\frac{1}{YP^2} \left(-c + \delta_3 \frac{Y}{(1+Y)} \right). \text{ The following is obtained:}$$

$$\frac{\partial H_2J_1}{\partial Y} + \frac{\partial H_2J_2}{\partial P} = \frac{1}{P^2} \left[\frac{c - \beta_2 Y}{Y} + \frac{\delta_2 P - \delta_3 - \delta_3 Y}{(1+Y)^2} \right] \text{ which is negative in the plane } Y > 0, P > 0 \text{ in the region } \Omega_2 = \left\{ 0 < \frac{c}{\beta_2} < Y, \quad 0 < P < \frac{\delta_3}{\delta_2} \right\}.$$

Therefore,

by Bendixson-Dulac criteria: $E_5(Y, P)$ is globally asymptotically stable in Ω_2 .

(vi) Global stability of the co-existence equilibrium point $E_6(X^*, Y^*, P^*)$

A stable Lyapunov function was used from which conditions for the global asymptotic stability of the co-existence point $E_6(X^*, Y^*, P^*)$ are derived. First, a lemma was provided to establish a region of attraction for the system represented by Equations (47). The approach was based on work by Takeuchi (1996), Chaudhuri and Kar (2002) and Dubey and Upadhyay (2004).

Lemma 1

$$\text{The set } \Omega = \left\{ 0 \leq X \leq \frac{1}{\beta_1}, \quad 0 \leq Y \leq \frac{1}{\beta_2}, \quad 0 \leq \epsilon_1 X + \epsilon_2 Y + P \leq \frac{\rho}{\eta} \right\}$$

Where $\epsilon_1 = \frac{c\gamma_3}{\delta_1}$, $\epsilon_2 = \frac{c\delta_3}{\delta_2}$, $\rho = \frac{\epsilon_1}{\beta_1}(1 + \eta) + \frac{\epsilon_2}{\beta_2}(1 + \eta)$ and $\eta \leq c$ is a region of attraction for all solutions initiated in the interior of the positive region (X, Y, P) .

Proof

From the Equation (47) note that $\frac{dX}{dt} \leq X(1 - \beta_1 X)$. This gives $X(t) \leq \frac{\Gamma}{e^{-t} + \Gamma\beta_1}$, where

$$\Gamma = \frac{X(0)}{1 - X(0)\beta_1}. \text{ As } t \rightarrow \infty \text{ gives } X(t) \leq \frac{1}{\beta_1}$$

Similarly from the second Equation of (47), it can be obtained $Y(t) \leq \frac{1}{\beta_2}$

Define a function $W(t) = \epsilon_1 X(t) + \epsilon_2 Y(t) + P(t)$ where $\epsilon_1 = \frac{c\gamma_3}{\delta_1}$, $\epsilon_2 = \frac{c\delta_3}{\delta_2}$ for a real positive number η .

$$\dot{W}(t) + \eta W(t) = \epsilon_1 \dot{X}(t) + \epsilon_2 \dot{Y}(t) + \dot{P}(t) + \eta(\epsilon_1 X(t) + \epsilon_2 Y(t) + P(t)) \quad (51)$$

Substituting $\dot{X}(t), \dot{Y}(t), \dot{P}(t)$ using (47) into equation (51) and simplifying gives:

$$W(t) + \eta W(t) = \epsilon_1 X(1 + \eta) + \epsilon_2 Y(1 + \eta) - \epsilon_1 \beta_1 X^2 - \epsilon_2 \beta_2 Y^2 + \epsilon_1 \gamma_1 XY + \epsilon_2 \gamma_2 XY - \epsilon_1 M_1 X - \epsilon_2 M_2 Y + (-c + \eta)P.$$

$$\text{Choose } \eta \leq c \text{ to obtain } \frac{dW}{dt} + \eta W \leq \frac{\epsilon_1}{\beta_1}(1 + \eta) + \frac{\epsilon_2}{\beta_2}(1 + \eta) = \rho.$$

This gives $W(t) \leq \frac{\rho}{\eta}(1 - e^{-\eta t}) + W(0)e^{-\eta t}$. As $t \rightarrow \infty$, $0 \leq W(t) \leq \frac{\rho}{\eta}$. This completes the proof.

Theorem 4: *Let the following inequalities hold in the region Ω defined in Lemma 1 then, the co-existence equilibrium point $E_6(X^*, Y^*, P^*)$ is globally asymptotically stable with respect to all solutions initiated in the interior of Ω .*

Proof

Consider the following Lyapunov function:

$$V(X, Y, P) = \left(X - X^* - X^* \ln \left(\frac{X}{X^*} \right) \right) + \left(Y - Y^* - Y^* \ln \left(\frac{Y}{Y^*} \right) \right) + \left(P - P^* - P^* \ln \left(\frac{P}{P^*} \right) \right).$$

Obviously, V is positive definite.

Differentiating V with respect to time t gives:

$$\dot{V}(X, Y, P) = \frac{(X - X^*)}{X} \dot{X}(t) + \frac{(Y - Y^*)}{Y} \dot{Y}(t) + \frac{(P - P^*)}{P} \dot{P}(t)$$

Substituting the expressions for $\dot{X}(t), \dot{Y}(t), \dot{P}(t)$ from Equation (47) gives:

$$\dot{V}(X, Y, P) = (X - X^*) \left(1 - \beta_1 X - M_1 + \gamma_1 Y - \frac{\delta_1 P}{(1+X)} \right) + (Y - Y^*) \left(1 - \beta_2 Y - M_2 + \gamma_2 X - \frac{\delta_2 P}{(1+Y)} \right) + (P - P^*) \left(-c + \frac{\gamma_3 X}{(1+X)} + \frac{\delta_3 Y}{(1+Y)} \right).$$

From this one obtains:

$$\begin{aligned}\dot{V}(X, Y, P) = & (X - X^*) \left((\beta_1 X^* - \gamma_1 Y^* + \frac{\delta_1 P^*}{(1 + X^*)}) - (\beta_1 X - \gamma_1 Y + \frac{\delta_1 P}{(1 + X)}) \right) \\ & + (Y - Y^*) \left((\beta_2 Y^* - \gamma_2 X^* + \frac{\delta_2 P^*}{(1 + Y^*)}) - (\beta_2 Y - \gamma_2 X + \frac{\delta_2 P}{(1 + Y)}) \right) \\ & + (P - P^*) \left((\frac{\gamma_3 X}{(1 + X)} + \frac{\delta_3 Y}{(1 + Y)}) - (\frac{\gamma_3 X^*}{(1 + X^*)} + \frac{\delta_3 Y^*}{(1 + Y^*)}) \right)\end{aligned}$$

This simplifies to:

$$\begin{aligned}\dot{V}(X, Y, P) = & -(X - X^*)^2 \left(\beta_1 - \frac{\delta_1 P^*}{w} \right) - (X - X^*)(Y - Y^*)(\gamma_1 + \gamma_2) - (X - X^*)(P - P^*) \left(\frac{\delta_1(1+X^*)-\gamma_3}{w} \right) \\ & - (Y - Y^*)^2 \left(\beta_2 - \frac{\delta_2 P^*}{z} \right) - (Y - Y^*)(P - P^*) \left(\frac{\delta_2(1+Y^*)-\delta_3(1+P^*)}{z} \right) - (P - P^*)^2 \left(\frac{\delta_3 Y^*}{z} \right).\end{aligned}$$

Where $w = (1 + X)(1 + X^*)$, $z = (1 + Y)(1 + Y^*)$

Thus, $\dot{V}(X, Y, P)$ is a quadratic function that can be expressed as:

$\dot{V} = -D^T A D$, where $D^T = (X - X^*, Y - Y^*, P - P^*)$ and A is a symmetric matrix given by

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}$$

$$\begin{aligned}\text{where } a_{11} = & \beta_1 - \frac{\delta_1 P^*}{w}, a_{12} = \frac{\gamma_1 + \gamma_2}{2}, a_{13} = \frac{\delta_1(1+X^*)-\gamma_3}{2w}, a_{22} = \beta_2 - \frac{\delta_2 P^*}{z}, a_{23} = \\ & \frac{\delta_2(1+Y^*)-\delta_3(1+P^*)}{2z}, a_{33} = \frac{\delta_3 Y^*}{z}.\end{aligned}$$

It is noted that $\dot{V} < 0$ if the matrix A is positive definite (Chaudhuri & Kar, 2004).

The matrix A is positive definite if $a_{11} > 0, a_{13} > 0, a_{22} > 0, a_{23} > 0, a_{33} > 0$, and $a_{11}a_{22} - a_{12}^2 > 0$. It is observed that $a_{11} > 0$ and $a_{22} > 0$ give $P^* < \frac{\beta_1 w}{\delta_1}$ and $P^* < \frac{\beta_2 z}{\delta_2}$, $a_{13} = 0$ gives $X^* = \frac{\gamma_3 - \delta_1}{\delta_1}$ provided $\gamma_3 > \delta_1$, $a_{23} = 0$ gives:

$$Y^* < \frac{\delta_3(\delta_1 + \beta_1 w) - \delta_1 \delta_2}{\delta_1 \delta_2} \text{ and } a_{11}a_{22} - a_{12}^2 > 0 \text{ gives:}$$

$\left(\beta_1 - \frac{\delta_1 P^*}{w}\right) \left(\beta_2 - \frac{\delta_2 P^*}{z}\right) - \left(\frac{\gamma_1 + \gamma_2}{2}\right)^2 > 0$ provided $\gamma_1 + \gamma_2 > 2\sqrt{\beta_1 \beta_2}$. This completes the proof.

Therefore, the conditions for the existence of all the seven possible equilibrium points (steady states) were established. It was found that wildebeests can exist on their own or in the presence of zebras and/ or lions only if the wildebeests' intrinsic rate of birth is greater than the rate at which they die. Since wildebeests and zebras are mutualistic, the two species would co-exist without a lion provided that the intrinsic rate of growth of wildebeests and zebras is greater than their death rates. The existence of the lion with either the wildebeest alone or zebra alone required the proportion of biomass of either prey species converted into fertility (reproductivity rate) by the predator to be greater than the product of the predator's natural mortality rate, and the time it takes to handle the prey. The ecosystem would be globally asymptotically stable. The dynamic behaviour of the three species is shown in Fig. 6

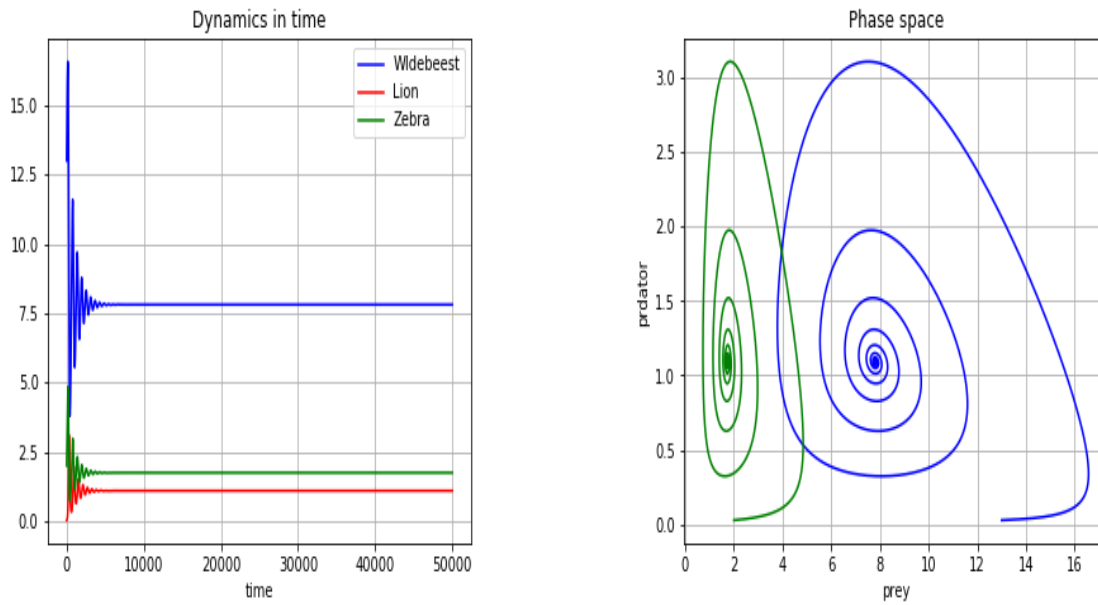


Figure 6: Wildebeest, zebra and lion population dynamics

3.6.6 Dispersion Relation

This section investigated the pattern dynamics of wildebeests, zebras, and lions caused by diffusion and advection. To consider the effects of the flow on the dispersion relations, we derived the phase dynamics from Equation 44 and estimated them from the phase wave. Thus, the set of conditions that lead the system of Equations (44) to instability due to diffusion and advection was established as follows:

Let the diffusion in each spatial dimension be the equal, i.e., $D_{11} = D_{12} = D_w$, $D_{21} = D_{22} = D_z$, $D_{31} = D_{32} = D_p$.

Also, let migration in each spatial dimension be equal, i.e., $C_{11} = C_{12} = C_w$, $C_{21} = C_{22} = C_z$ and $C_{31} = C_{32} = C_p$

where:

D_w and C_w diffusion and advection parameters for prey w in two dimensions

D_z and C_z diffusion and advection parameters for prey z in two dimensions

D_p and C_p diffusion and advection parameters for predator p in two dimensions

Equation (44) can be redefined to obtain the following Equation:

$$\begin{aligned}\frac{\partial w}{\partial t} &= D_w \nabla^2 w - C_w \nabla w + r_1 w \left(1 - \frac{w}{K_1}\right) + b_1 wz - \frac{h_1 w P}{1+a_1 w} - \lambda_1 w \\ \frac{\partial z}{\partial t} &= D_z \nabla^2 z - C_z \nabla z + r_2 z \left(1 - \frac{z}{K_2}\right) + b_2 wz - \frac{h_2 z P}{1+a_2 z} - \lambda_2 z \\ \frac{\partial P}{\partial t} &= D_p \nabla^2 p - C_p \nabla p + P \left(-c + d_1 \frac{h_1 w}{1+a_1 w} + d_2 \frac{h_2 z}{1+a_2 z}\right)\end{aligned}\quad (52)$$

$$\text{Let } \left. \begin{aligned} f(w, z, p) &= r_1 w \left(1 - \frac{w}{K_1}\right) + b_1 wz - \frac{h_1 w}{1+a_1 w} P - \lambda_1 w \\ g(w, z, p) &= r_2 z \left(1 - \frac{z}{K_2}\right) + b_2 wz - \frac{h_2 z}{1+a_2 z} P - \lambda_2 z \\ h(w, z, p) &= P \left(-c + d_1 \frac{h_1 w}{1+a_1 w} + d_2 \frac{h_2 z}{1+a_2 z}\right) \end{aligned} \right\} \quad (53)$$

For the linear stability analysis, re-writing Equation (53) gives:

$$\left. \begin{aligned} \frac{\partial w}{\partial t} &= f(w, z, p) - C_w \nabla w + D_w \nabla^2 w \\ \frac{\partial z}{\partial t} &= g(w, z, p) - C_z \nabla z + D_z \nabla^2 z \\ \frac{\partial p}{\partial t} &= h(w, z, p) - C_p \nabla p + D_p \nabla^2 p \end{aligned} \right\} \quad (54)$$

Suppose that the homogeneous spatial system of Equations (54) has a limiting cycle with frequency ω_0 .

That is:

$$\left. \begin{aligned} w &= w_0(\tau) \\ z &= z_0(\tau) \\ p &= p_0(\tau) \end{aligned} \right\} \quad (55)$$

With $\tau = \omega_0 t$. This leads to:

$$\left. \begin{aligned} \omega_0 \frac{\partial w_0}{\partial \tau} &= f(w_0, z_0, p_0), & w_0(\tau + 2\pi) &= w_0(\tau) \\ \omega_0 \frac{\partial z_0}{\partial \tau} &= g(w_0, z_0, p_0), & z_0(\tau + 2\pi) &= z_0(\tau) \\ \omega_0 \frac{\partial p_0}{\partial \tau} &= h(w_0, z_0, p_0), & p_0(\tau + 2\pi) &= p_0(\tau) \end{aligned} \right\} \quad (56)$$

System of Equations (54) is invariant under time translation, and it has a solution:

$$\left. \begin{aligned} w &= w_0(\tau + \psi) \\ z &= z_0(\tau + \psi) \\ p &= p_0(\tau + \psi) \end{aligned} \right\} \text{ where } \psi \text{ is an arbitrary constant.}$$

For a slow spatial modulation, some multiple scales (spatial coordinates X and temporal coordinates T) are introduced as follows:

$$X = \sqrt{\epsilon} x, \quad \tau = \omega_0 t, \quad T = \epsilon t, \quad (57)$$

and asymptotic expansion:

$$\left. \begin{aligned} w &= w_0(\tau + \psi) + \epsilon w_1(\tau + \psi) + \dots \\ z &= z_0(\tau + \psi) + \epsilon z_1(\tau + \psi) + \dots \\ p &= p_0(\tau + \psi) + \epsilon p_1(\tau + \psi) + \dots \end{aligned} \right\} \quad (58)$$

Where ϵ is a small parameter and $\psi = \psi(X, T)$. Substituting Equations (57) and (58) into Equations (54) yields a hierarchy of linear equations for each order in ϵ :

$$\left. \begin{aligned} \omega_0 \frac{\partial w_0}{\partial \tau} &= f(w_0, z_0, p_0) \\ \omega_0 \frac{\partial z_0}{\partial \tau} &= g(w_0, z_0, p_0) \\ \omega_0 \frac{\partial p_0}{\partial \tau} &= h(w_0, z_0, p_0) \end{aligned} \right\} \quad (59)$$

and

$$\mathcal{L} \begin{pmatrix} w_j \\ z_j \\ p_j \end{pmatrix} = \begin{pmatrix} A_j \\ B_j \\ H_j \end{pmatrix} \quad (60)$$

Where,

$$\mathcal{L} = \begin{pmatrix} \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial f}{\partial w}(w_0, z_0, p_0) & -\frac{\partial f}{\partial z}(w_0, z_0, p_0) & -\frac{\partial f}{\partial p}(w_0, z_0, p_0) \\ -\frac{\partial g}{\partial w}(w_0, z_0, p_0) & \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial g}{\partial z}(w_0, z_0, p_0) & -\frac{\partial g}{\partial p}(w_0, z_0, p_0) \\ -\frac{\partial h}{\partial w}(w_0, z_0, p_0) & -\frac{\partial h}{\partial z}(w_0, z_0, p_0) & \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial h}{\partial p}(w_0, z_0, p_0) \end{pmatrix} \quad (61)$$

A_j, B_j and H_j are denoted as the inhomogeneous term of the j th order equation for $j = 1, 2, \dots$.

For the first-order equation of the inhomogeneous term is:

$$\left. \begin{aligned} A_1 &= -w'_0 \frac{\partial \psi}{\partial T} - C_w w'_0 \nabla \psi + D_w (w''_0 |\nabla \psi|^2 + w'_0 \nabla^2 \psi) \\ B_1 &= -z'_0 \frac{\partial \psi}{\partial T} - C_z z'_0 \nabla \psi + D_z (z''_0 |\nabla \psi|^2 + z'_0 \nabla^2 \psi) \\ H_1 &= -p'_0 \frac{\partial \psi}{\partial T} - C_p p'_0 \nabla \psi + D_p (p''_0 |\nabla \psi|^2 + p'_0 \nabla^2 \psi) \end{aligned} \right\} \quad (62)$$

Denote that $s_0 = (w_0, z_0, p_0)^T$, $s_1 = (w_1, z_1, p_1)^T$ and $\bar{s} = (\bar{w}, \bar{z}, \bar{p})^T$ is the nontrivial periodic solution to the adjoint differential equation $\bar{\mathcal{L}}\bar{s} = 0$ where:

$$\bar{\mathcal{L}} = \begin{pmatrix} \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial f}{\partial w}(w_0, z_0, p_0) & -\frac{\partial f}{\partial z}(w_0, z_0, p_0) & -\frac{\partial f}{\partial p}(w_0, z_0, p_0) \\ -\frac{\partial g}{\partial w}(w_0, z_0, p_0) & \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial g}{\partial z}(w_0, z_0, p_0) & -\frac{\partial g}{\partial p}(w_0, z_0, p_0) \\ -\frac{\partial h}{\partial w}(w_0, z_0, p_0) & -\frac{\partial h}{\partial z}(w_0, z_0, p_0) & \omega_0 \frac{\partial}{\partial \tau} - \frac{\partial h}{\partial p}(w_0, z_0, p_0) \end{pmatrix} \quad (63)$$

By using solvability conditions for s_1 yields:

$$\langle \bar{s}, s'_0 \rangle \frac{\partial \psi}{\partial T} = \langle \bar{s}, \xi_1 \rangle \nabla \psi + \langle \bar{s}, \xi_2 \rangle \nabla^2 \psi + \langle \bar{s}, \xi_3 \rangle |\nabla \psi|^2 \quad (64)$$

This leads to the dynamics of the phase waves:

$$\frac{\partial \psi}{\partial T} = \rho_1 \frac{\partial \psi}{\partial X} + \rho_2 \frac{\partial^2 \psi}{\partial X^2} + \rho_3 \left| \frac{\partial \psi}{\partial T} \right|^2 \quad (65)$$

Where, $\rho_i = \frac{\langle \bar{s}, \xi_i \rangle}{\langle \bar{s}, s'_0 \rangle}$

With $\xi_1 = (-C_w w'_0, -C_z z'_0, -C_p p'_0)^T$, $\xi_2 = (-D_w w'_0, -D_z z'_0, -D_p p'_0)^T$, $\xi_3 = (D_w w''_0, D_z z''_0, D_p p''_0)^T$

And

$$\langle \bar{s}, \xi \rangle = \int_0^{2\pi} (\bar{s}, \xi) d\tau. \quad (66)$$

This results in the solvability condition of \bar{s} leads to:

$$\begin{aligned} \int_0^{2\pi} \bar{w} \left[-w'_0 \frac{\partial \psi}{\partial \tau} - \square_w w'_0 \nabla \psi + D_w (w''_0 |\nabla \psi|^2 + w'_0 \nabla^2 \psi) \right] d\tau + \int_0^{2\pi} \bar{z} \left[-z'_0 \frac{\partial \psi}{\partial \tau} - C_z z'_0 \nabla \psi + \right. \\ \left. D_z (z''_0 |\nabla \psi|^2 + z'_0 \nabla^2 \psi) \right] d\tau + \int_0^{2\pi} \bar{p} \left[-p'_0 \frac{\partial \psi}{\partial \tau} - C_p p'_0 \nabla \psi + D_p (p''_0 |\nabla \psi|^2 + p'_0 \nabla^2 \psi) \right] d\tau = 0 \end{aligned} \quad (67)$$

Where \bar{w} , \bar{z} and \bar{p} satisfy the Equations:

$$\omega_0 \begin{pmatrix} \frac{\partial w_0}{\partial \tau} \\ \frac{\partial z_0}{\partial \tau} \\ \frac{\partial p_0}{\partial \tau} \end{pmatrix} = \begin{pmatrix} f(w_0, z_0, p_0) \\ g(w_0, z_0, p_0) \\ h(w_0, z_0, p_0) \end{pmatrix} \quad (68)$$

And

$$-\omega_0 \begin{pmatrix} \frac{\partial \bar{w}}{\partial \tau} \\ \frac{\partial \bar{z}}{\partial \tau} \\ \frac{\partial \bar{p}}{\partial \tau} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial w}(w_0, z_0, p_0) & \frac{\partial g}{\partial w}(w_0, z_0, p_0) & \frac{\partial h}{\partial w}(w_0, z_0, p_0) \\ \frac{\partial f}{\partial z}(w_0, z_0, p_0) & \frac{\partial g}{\partial z}(w_0, z_0, p_0) & \frac{\partial h}{\partial z}(w_0, z_0, p_0) \\ \frac{\partial f}{\partial p}(w_0, z_0, p_0) & \frac{\partial g}{\partial p}(w_0, z_0, p_0) & \frac{\partial h}{\partial p}(w_0, z_0, p_0) \end{pmatrix} \begin{pmatrix} \bar{w} \\ \bar{z} \\ \bar{p} \end{pmatrix} \quad (69)$$

Substituting $\emptyset = \omega_0 t + \psi$, Equation (65) takes the following form:

$$\frac{\partial \emptyset}{\partial \tau} = \omega_0 + \rho_1 \nabla \emptyset + \rho_2 \nabla^2 \emptyset + \rho_3 |\nabla \emptyset|^2 \quad (70)$$

Through the wave characteristic $\omega = \frac{\partial \emptyset}{\partial t}$ and $k = \nabla \emptyset$, the dispersion relation is determined from the phase Equation (70):

$$\omega = \omega_0 + \rho_1 k + \rho_3 k^2 + \dots \quad (71)$$

$$\rho_1 = \frac{\int_0^{2\pi} \bar{w}(-C_w w'_0) + \bar{z}(-C_z z'_0) + \bar{p}(-C_p p'_0) d\tau}{\int_0^{2\pi} (\bar{w} w'_0 + \bar{z} z'_0 + \bar{p} p'_0) d\tau}, \quad \rho_3 = \frac{\int_0^{2\pi} (\bar{w} D_w w''_0 + \bar{z} D_z z''_0 + \bar{p} D_p p''_0) d\tau}{\int_0^{2\pi} (\bar{w} w'_0 + \bar{z} z'_0 + \bar{p} p'_0) d\tau} \quad (72)$$

The effects of varying the values of C_w , C_z and C_p and D_w , D_z and D_p depend on the dispersion relation presented by Equation (72). Increasing the values of C_w , C_z and C_p , the frequency ω increases for fixed values of wave number k . Therefore, the advection (migration) term greatly affects the dynamic behaviour of the model. The effects of diffusion and migration on the system's stability can further be analysed using the numerical methods of model 72.

3.6.7 Numerical Methods

The Python computer program performed a numerical simulation of the model represented by the advection-diffusion reaction Equation (44). The following are the parameters used for simulation. Other parameters were assumed to vary within the corresponding intervals.

Table 2: Parameters of the two preys and predators with their sources

Parameter Description	symbol	value	source
Per capita intrinsic growth rate for prey	r_1 and r_2	1 and 0.8	Mduma (1996)
Carrying capacities for prey w and z	K_1 and K_2	18×10^5 & 3×10^4	Assumed
Mutualism between prey w and z	b_1 and b_2	0.015 & 0.02	Fay and Greef (2006)
Capturing rates of the predator P	h_1 and h_2	0.674 & 0.75	Fryxell (2007)
Prey biomass handled per unit time	a_1 and a_2	0.03 & 0.032	Assumed
Efficiency to convert prey biomass of w and z respectively into fertility	d_1 and d_2	0.371 & 0.525	Sagamiko (2014)
The natural mortality rate of predator	c	1.2	Schaller (1972)
Death rates due to drought for prey	λ_1 and λ_2	0.1 and 0.08	Mduma (1996)

The mathematical model (44) was discretized subject to the following boundary condition:

$$w(x, 0, t) = w(L_x, 0, t) = 0, w(0, y, t) = w(0, L_y, t) = 0 \text{ and } w(x, y, 0) = w^0,$$

$$z(x, 0, t) = z(L_x, 0, t) = 0, z(0, y, t) = z(0, L_y, t) = 0 \text{ and } z(x, y, 0) = z^0,$$

$$P(x, 0, t) = P(L_x, 0, t) = 0, P(0, y, t) = P(0, L_y, t) = 0 \text{ and } P(x, y, 0) = P_0,$$

The space was set to $0 < x < L_x, 0 < y < L_y$ and time was set to be $0 < t < T$.

For simulation purposes, the parameters were set to be $L_x = 1, L_y = 4$ and $T = 100$.

The diffusion parameters were set to be $D_w = D_z = 0.01, D_p = 0.01$.

And,

Migration (advective) parameters were $C_w = C_{11} = C_{12} = 0.005$, $C_z = C_{21} = C_{22} = 0.005$, $C_p = C_{31} = C_{32} = 0.005$. $M_w = M_z = M_p = 1$

The numerical solution used in the current study was the explicit Euler method in which the discretization of time derivative was performed by using the forward difference rule and central difference approximation of the Laplace equations in two dimensions, as shown below:

$$\begin{aligned} \frac{w_{i,j}^{n+1} - w_{i,j}^n}{n} = & D_{11} \left(\frac{w_{i+1,j}^n - 2w_{i,j}^n + w_{i-1,j}^n}{h^2} \right) + D_{12} \left(\frac{w_{i,j+1}^n - 2w_{i,j}^n + w_{i,j-1}^n}{h^2} \right) - C_{11} \left(\frac{w_{i+1,j}^n - w_{i,j}^n}{h} \right) - \\ & C_{12} \left(\frac{w_{i,j+1}^n - w_{i,j}^n}{h} \right) + r_1 w_{i,j}^n \left(1 - \frac{w_{i,j}^n}{K_1} \right) + b_1 w_{i,j}^n z_{i,j}^n - \frac{h w_{i,j}^n p_{i,j}^n}{1 + a w_{i,j}^n} - \lambda_1 w_{i,j}^n \end{aligned} \quad (3.73a)$$

$$\begin{aligned} \frac{z_{i,j}^{n+1} - z_{i,j}^n}{k} = & D_{21} \left(\frac{z_{i+1,j}^n - 2z_{i,j}^n + z_{i-1,j}^n}{h^2} \right) + D_{22} \left(\frac{z_{i,j+1}^n - 2z_{i,j}^n + z_{i,j-1}^n}{h^2} \right) - C_{21} \left(\frac{z_{i+1,j}^n - z_{i,j}^n}{h} \right) - \\ & C_{22} \left(\frac{z_{i,j+1}^n - z_{i,j}^n}{h} \right) + r_2 z_{i,j}^n \left(1 - \frac{z_{i,j}^n}{K_w} \right) + b_2 w_{i,j}^n z_{i,j}^n - \frac{d h z_{i,j}^n p_{i,j}^n}{1 + a z_{i,j}^n} - \lambda_2 z_{i,j}^n \end{aligned} \quad (3.73b)$$

$$\begin{aligned} \frac{p_{i,j}^{n+1} - p_{i,j}^n}{k} = & D_{31} \left(\frac{p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n}{h^2} \right) + D_{32} \left(\frac{p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n}{h^2} \right) - C_{31} \left(\frac{p_{i+1,j}^n - p_{i,j}^n}{h} \right) - \\ & C_{32} \left(\frac{p_{i,j+1}^n - p_{i,j}^n}{h} \right) - c p_{i,j}^n + \frac{d h w_{i,j}^n p_{i,j}^n}{1 + a w_{i,j}^n} + \frac{d h p_{i,j}^n z_{i,j}^n}{1 + a z_{i,j}^n} \end{aligned} \quad (73c)$$

The systems of Equations (73) simulations were performed in the python computer program (the corresponding codes are attached in appendix three). Finally, the numerical simulations, results, and discussions are presented in Chapter Four.

3.7 Chapter Summary

This chapter has given insight into how different mathematical models were formulated and how the data analyses have been carried out. The next chapter presents the results, and discussions of the study.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

The results and discussions presented in this chapter have highlighted the dynamics of wildebeest, zebra and lion.

4.1.1 Foraging Processes of Wildebeest

(i) Random Walk

The GPS collared wildebeest data on 2D lattice were fitted to show random walk trajectories for each wildebeest. The positions (x and y) show how different animals walk randomly in the Serengeti ecosystem. First, random walk trajectories for five wildebeests were plotted to visualize the movement trends (Fig. 7) and second, all 18 wildebeest trajectories were plotted (Fig. 8). A consistent movement of wildebeest from Southern Serengeti to the western part that continues to the northern part heading towards Masai Mara in Kenya was observed. In addition, animals are consistently moving in a specified direction (biased random walk) (Fig. 1).

The average position and distance of each walker were calculated. The GPS data used in the current study were recorded after each time step τ (6 hours), where an individual animal can move a distance δ either up, down, left, or right with probabilities dependent on the location given by $u(x, y)$, $d(x, y)$, $l(x, y)$ and $r(x, y)$ respectively with $u + d + l + r \leq 1$, or remain at the same location with probability $1 - u(x, y) - d(x, y) - l(x, y) - r(x, y)$. Each animal generated its jump probabilities depending on its movement patterns and foraging needs as recorded by the GPS data.

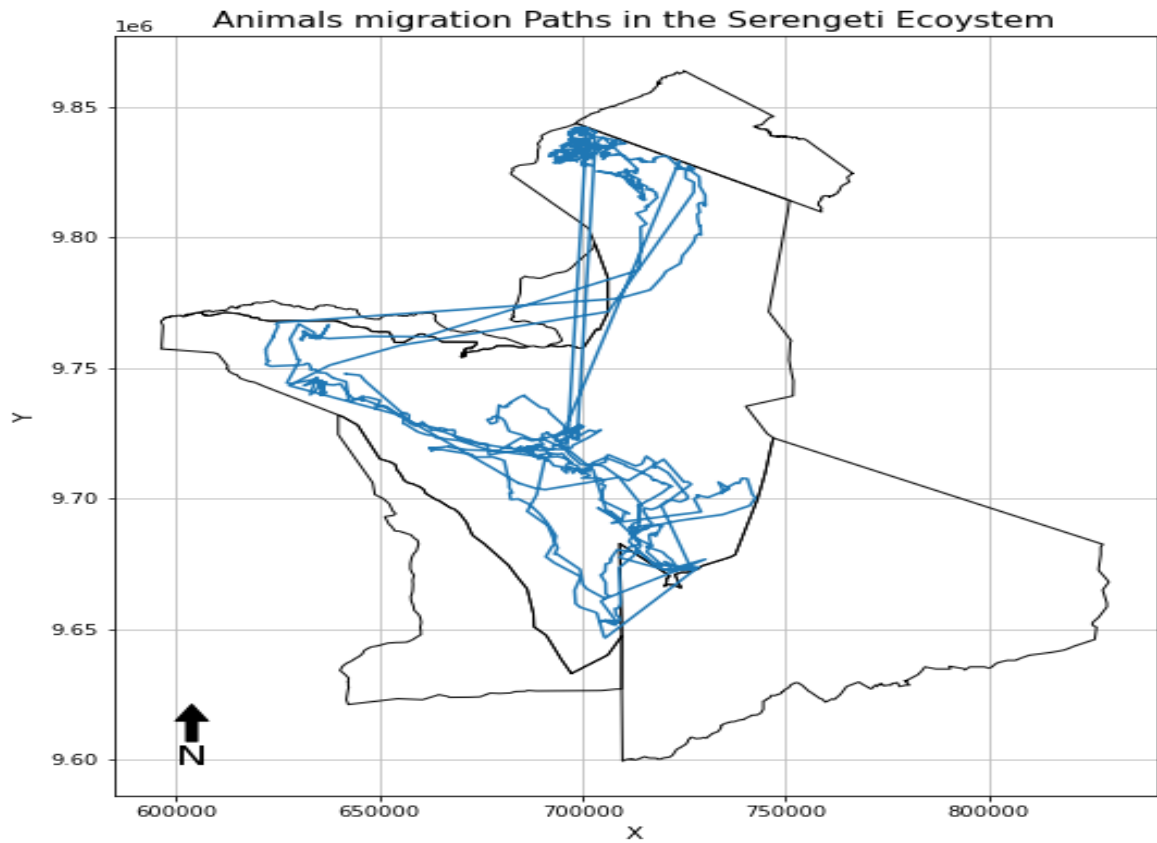


Figure 6: Individual random walk trajectories of 5 wildebeest on a 2D lattice (Units of the grid are in Universal Transverse Mercator [UTM])

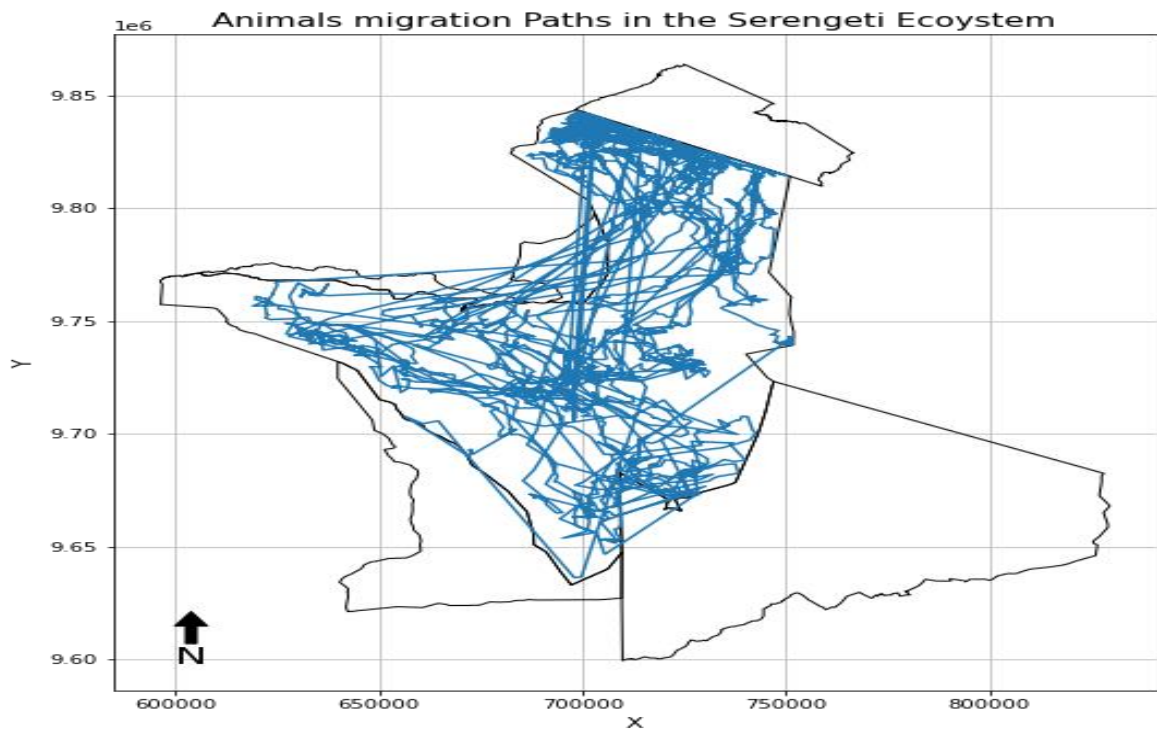


Figure 7: Individual random walk trajectories of 18 wildebeest on a 2D lattice (Units of the grid are in UTM)

The random walk was motivated by diffusion and advection movement parameters in two dimensions (Equation 73). The diffusion parameters were D_x and D_y in x and y directions, respectively, while the advective parameters were u_x and u_y in x and y directions, respectively. The average movement parameters calculated by the mathematical model (Equation 73) for each year are shown in Table 3.

Table 3: Calculated hourly average diffusion and advection movement parameters from the advection-diffusion equation for each year

Year	$(D_x M^2/\text{Hour})$	$D_y (M^2/\text{Hour})$	$u_x (\text{M/ Hour})$	$u_y (\text{M/ Hour})$
1999	112429.80	102578.94	12.74	19.47
2000	113845.89	101548.44	-8.33	-14.96
2003	116051.93	104569.41	-12.95	5.24
2004	116139.19	104336.58	22.80	6.76
2005	115445.82	104335.65	2.38	12.66
2006	116139.19	104648.03	11.87	23.79
2007	113988.47	102710.10	14.78	23.84

There is a large variability between directed (u_x and u_y) and dispersive (D_x and D_y) components of the movement in different years. The dispersive components seem larger than the directed components (Table 3). But it must be understood that the unit for D_x and D_y is M^2 while the unit of u_x and u_y is M . There is a slight variability in diffusion parameters for different years. This shows that the annual rate at which animals spread in search of forage resources at different seasons of the year is constant. However, the migration parameters (advection) seem to be changing over the years. In the following section, seasonal movement patterns of wildebeests were analysed.

(ii) Wildebeest Movement Patterns in the Dry Season

Usually, the dry season starts in June when most columns Serengeti wildebeests seem to be migrating north to seek fresh grazing and water (Hopcraft, 2010). Wildebeests slowly start (in May) their movements from southern Serengeti following fresh grass in central Serengeti (Hopcraft, 2010). Large wildebeest movements are observed heading to seek forage refuge in the northern woodlands of the Serengeti National park and Masai Mara National Reserve in Kenya when the southern highlands (south Serengeti ecosystem) go dry (Fig. 9 and 10). As the dry season starts to hit in June, large concentrations of wildebeest are observed in the west of Serengeti and on the southern banks of the Grumeti River (Mduma, 1996). Wildebeests

congregate around this area, and form large herds before crossing dangerous rivers. This is the toughest moment of their journey as each wildebeest must face the challenge of crossing the crocodile-infested rivers. In late June and July, large herds of wildebeest and zebra continue to head north, crossing the Mara River north of Serengeti to Masai Mara National Reserve in Kenya. By July, most wildebeest herds are foraging around the northern woodlands of Serengeti National Park and Masai Mara National Reserve in Kenya. This area is characterized by short diffusion movements with little migration from place to place (Fig. 9 and Table 4).

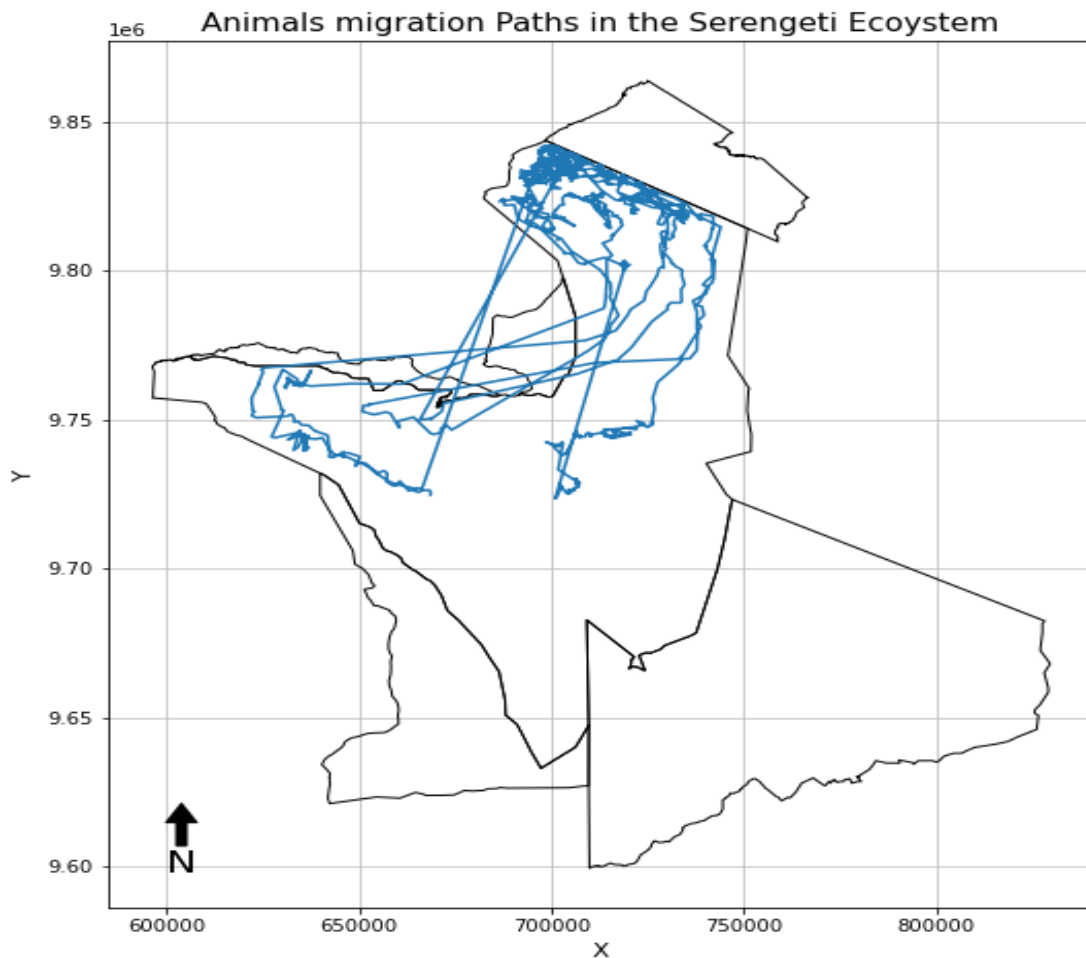


Figure 8: Individual random walk trajectories for 5 wildebeest on a 2D plane in the dry season (the units of the grid are in UTM)

The trajectories show the migration from the central Serengeti towards the western part of the ecosystem. Wildebeest arrive in the northern woodlands of Serengeti national park and Masai Mara in Kenya. Figure 8 and 9 show the northern part of the Serengeti ecosystem, where most of the movements occur in the dry season.

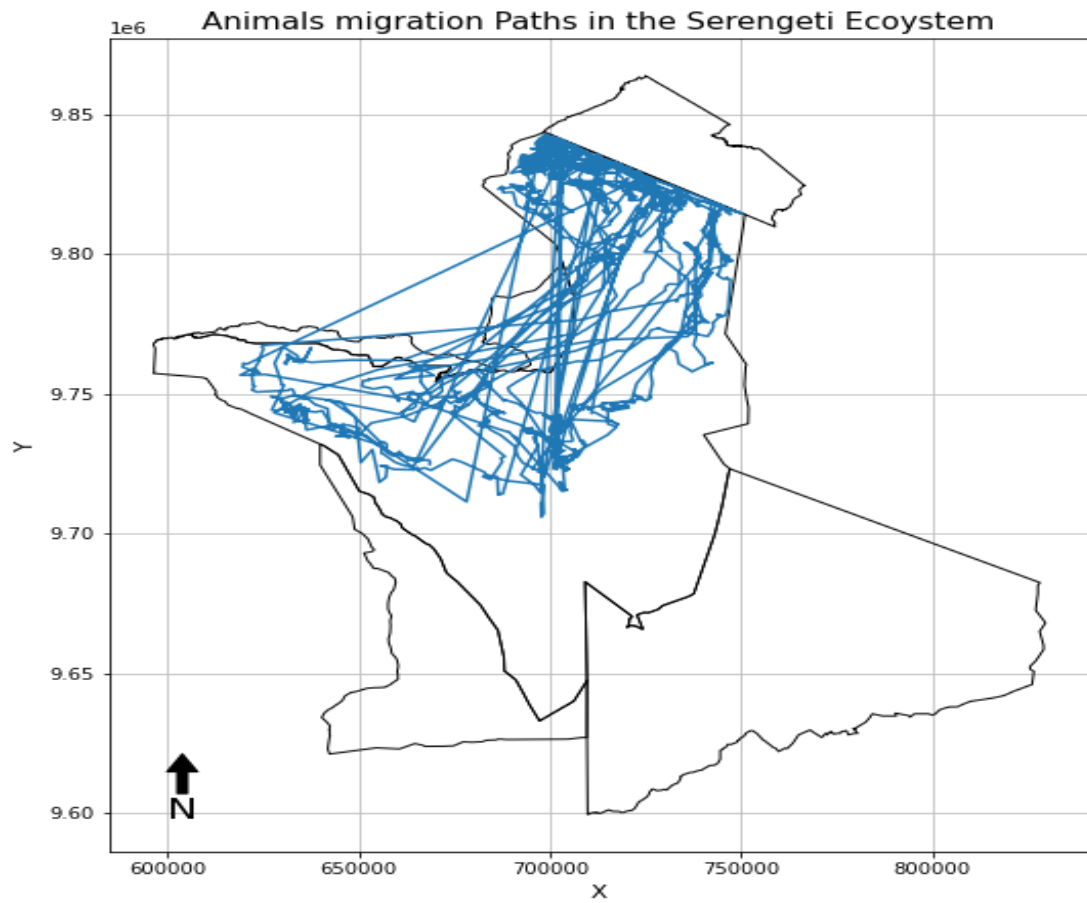


Figure 9: Individual random walk trajectories for 18 wildebeest on a 2D lattice in the dry season (the units of the grid are in UTM)

The average advection and diffusion movement parameters for the dry season are summarized in Table 4.

Table 4: Dry season hourly average diffusion and advection movement parameters calculated from the advection-diffusion equation for each year

Year	$D_x(M^2/Hour)$	$D_y(M^2/Hour)$	$u_x (M/ Hour)$	$u_y(M/ Hour)$
1999	88243.43	84281.22	11.32	24.89
2000	89815.63	82734.75	-5.12	-1.10
2003	91633.66	85883.85	-15.50	12.82
2005	91070.901	85671.36	0.50	20.18
2006	91743.01	85986.33	11.43	26.86
2007	90965.52	85257.63	-8.04	10.00

(iii) Wildebeest Movement Patterns in the Wet Season

The wet season starts from November to May, and short rains start in early November (Tourney *et al.*, 2018) when wildebeests start their journey back to the Serengeti ecosystem. By December, most of the wildebeest herds arrive on the short-grass plains of the southern Serengeti from the north. Large wildebeest movements (both advection and diffusion) are observed during this season (Fig. 11 and Table 5).

During the wet season, wildebeests spread south and east of Seronera, Ndutu and around the Ngorongoro conservation area. They disperse across these plains, feeding fresh and nutritious grass (Mduma, 1996) to most wildebeests calve in late January or early February (Hopcraft, 2010). They stay there around January, February, and March and gradually spread west across these plains. There are high average distances travelled by wildebeests in the wet season compared to the dry season (Table 5). The rate of spread (diffusion) is the highest compared to other seasons of the year.

Around April, wildebeests start to drift north-west towards the fresh grass of the central Serengeti, drawing thousands of zebra, Thomson's Gazelles and other ungulates (Holdo, 2011). Therefore, the annual wildebeest migration responds to local cues when they search, especially for rainfall, nutritious grass and water.

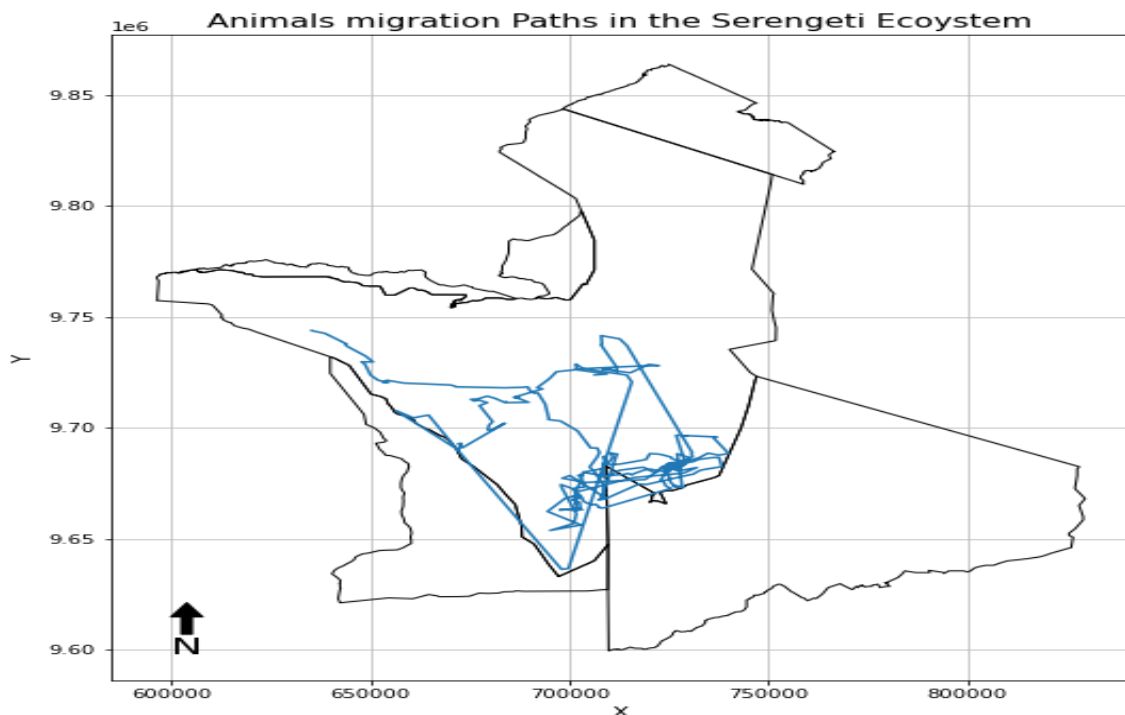


Figure 10: Individual random walk trajectories for 5 wildebeest on a 2D lattice in the wet season (the grid units are in UTM)

Wildebeest arrive in the Serengeti national park from the Masai Mara in Kenya. Figures 11 and 12 show the southern part of the Serengeti plains. Most of the movements take place in this area during the wet season with little migration towards the central Serengeti.

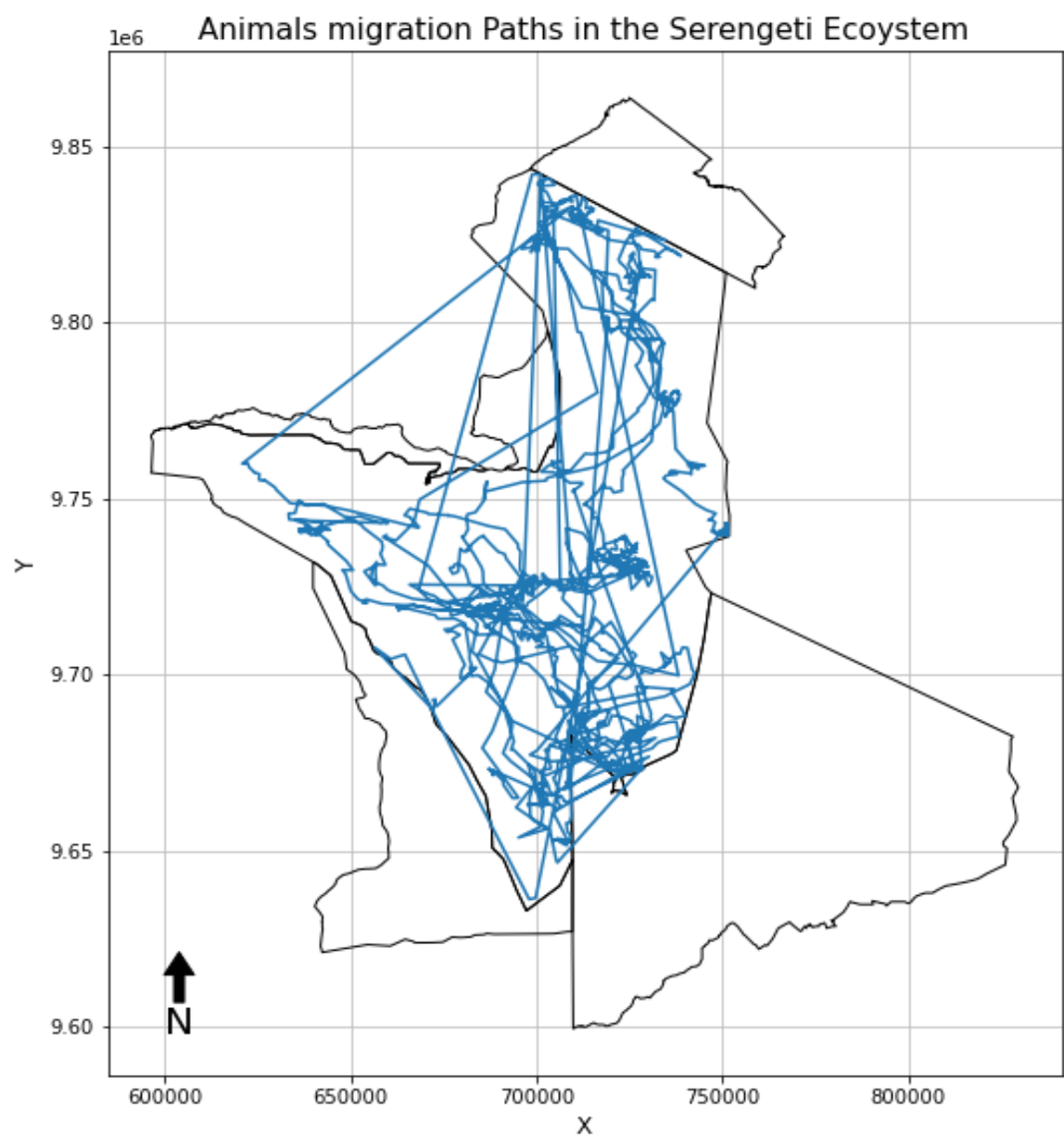


Figure 11: Individual random walk trajectories for 18 wildebeest on a 2D lattice in the wet season (the grid units are in UTM)

The average advection and diffusion movement parameters for the dry season are summarized in Table 5.

Table 5: Wet season hourly average diffusion and advection movement parameters calculated from the advection-diffusion equation for each year

Year	$D_x(M^2/Hour)$	$D_y(M^2/Hour)$	$u_x (M/ Hour)$	$u_y(M/ Hour)$
1999	169107.17	18125.18	16.01	10.41
2000	170390.74	159220.77	-16.48	-57.39
2003	173406.51	161325.01	-4.74	13.70
2004	173406.51	160844.88	30.08	9.07
2005	173406.51	161325.01	10.35	-49.94
2006	173406.51	161325.01	-15.04	-87.03
2007	168043.42	156335.58	56.43	54.43

4.1.2 Mutualism between Wildebeests and Zebras

Wildebeests and zebras populations were allowed to grow from their current estimates (1.3 million wildebeests and 200 000 zebras) to the set (assumed) carrying capacities of 1.8 million wildebeests and 300 000 zebras. The aim was to understand mathematically how wildebeests and zebras interact and what factors regulate their movement patterns. From the dynamic behaviour of the two species, there is coexistence, and both species approach their respective carrying capacities (Fig. 13) logistically. The population dynamics of the two species are summarised in Fig. 13 and Table 6.

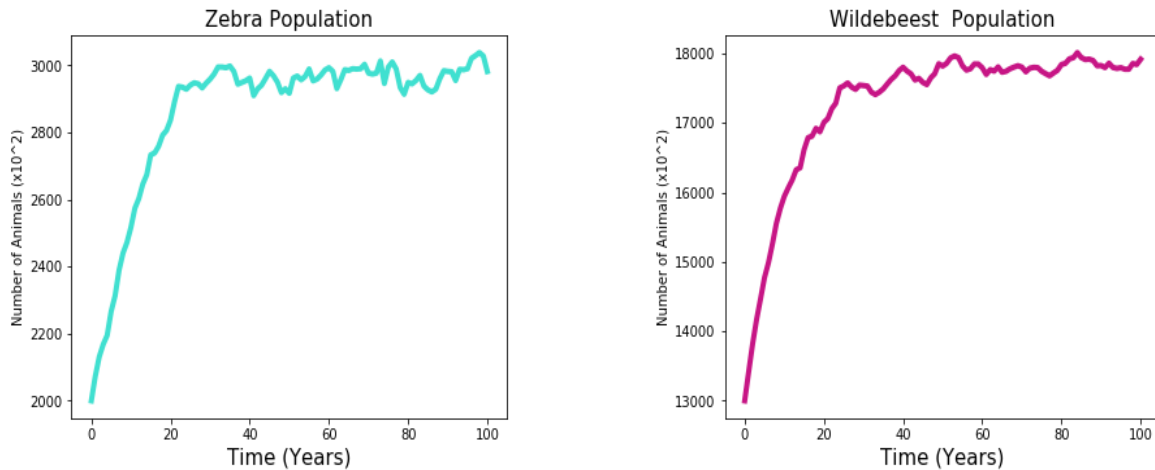


Figure 12: Wildebeest and zebra population growth with time

The dynamics in time show the population growth of wildebeest to their carrying capacities. The parameters used are taken from Table 1. Table 6 summarizes wildebeest and zebra population growth with time.

Table 6: Population dynamics of wildebeest and zebra in absence of predation and drought

Time	Zebra Number	Wildebeest Number
0	200 000	1 300 000
10	250 600	1 589 200
20	287 100	1 717 400
40	291 100	1 773 600
60	298 300	1 800 800
80	300 000	1 804 600
100	301 000	1 804 000

From Table 6, it can be observed that the wildebeest population increased logistically due to the absence of predation and drought. However, when predation and periodic drought seasons were allowed in the model, the population growth increased from 1.3 million wildebeests and 200 000 zebras to about 1.315 million wildebeests and 210 000 zebras (Fig. 14 and Table 7).

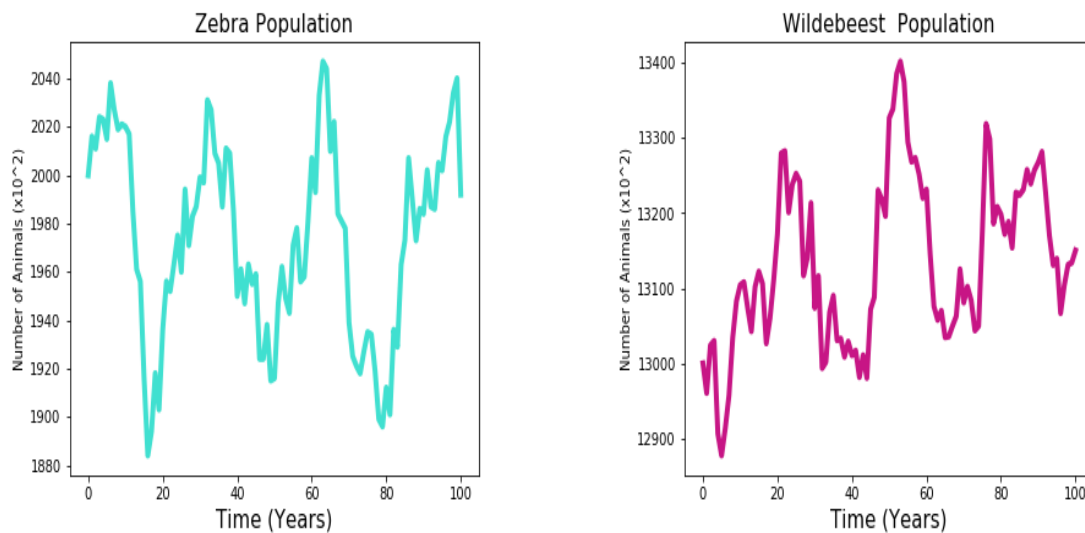


Figure 13: Effect of predation on wildebeest and zebra population growth with time

Table 7 summarizes wildebeest and zebra population growth with time.

Table 7: Population dynamics of wildebeest and zebra in presence predation and moderate drought

Time	Zebra Number	Wildebeest Number
0	200 000	1 300 000
10	201 100	1 312 100
20	202 500	1 312 300
40	204 600	1 323 800
60	206 000	1 343 100
80	207 000	1 313 400
100	210 000	1 315 300

Further, when rainfall is not enough and drought seasons hit the ecosystem, the two species do not get enough grass and water for survival (Fig. 14). This results in a decline in the population of both species. Wildebeest decreased from 1.3 million to about 0.95 million individuals, while zebra decreased from 200 000 to 127 000 individuals (Fig. 15 and Table 8).

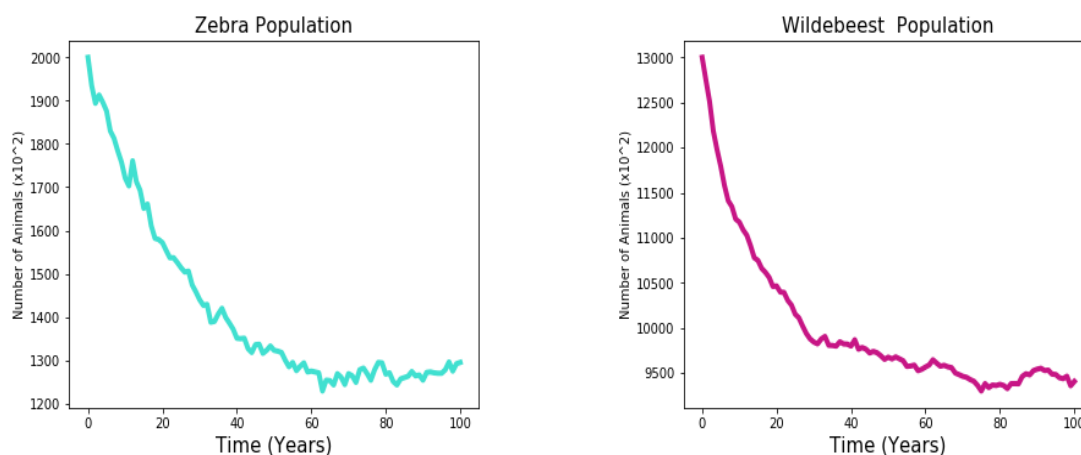


Figure 14: Effect of intensive drought on wildebeest and zebra populations

The dynamics in time show the population decline of wildebeest and zebra from their current estimates. Table 8 summarizes wildebeest and zebra population growth with time.

Table 8: Population dynamics of wildebeest and zebra in the presence of extreme drought

Time	Zebra Number	Wildebeest Number
0	200 000	1 300 000
10	172 100	1 117 300
20	157 200	1 046 500
40	137 100	979 900
60	127 500	956 600
80	126 800	937 400
100	129 500	940 600

From Fig. 13 to Fig. 15, the PDEs for the mutual reaction between wildebeest and zebra mimic the real situation within the Serengeti ecosystem. The PDEs for the mutual reaction between wildebeests and zebras can be observed to mimic the real situation within the Serengeti ecosystem. The death rate due to predation (from lions) and drought rates are the main parameters determining the abundance of wildebeests and zebras during the great migration.

Further, the results of the advection-diffusion reaction system (Equation 21) were numerically discretized to get Equation 73. Simulations for the interaction of mutualistic behaviour of wildebeests and zebras were performed to mimic the real movement of the two species. All parameters in the model were defined (Table 1) to allow individuals to navigate freely. Each individual was allowed to perform a random walk in any preferred direction. After every time step, the distribution of animals on the xy -plane was recorded. These distributions were plotted on a 3D histogram based on the empirical data and the advection-diffusion reaction equation as a surface. Also, the walkers' distributions on 2D histograms along x and y axes were plotted. The results are presented in Fig. 16 to Fig. 19.

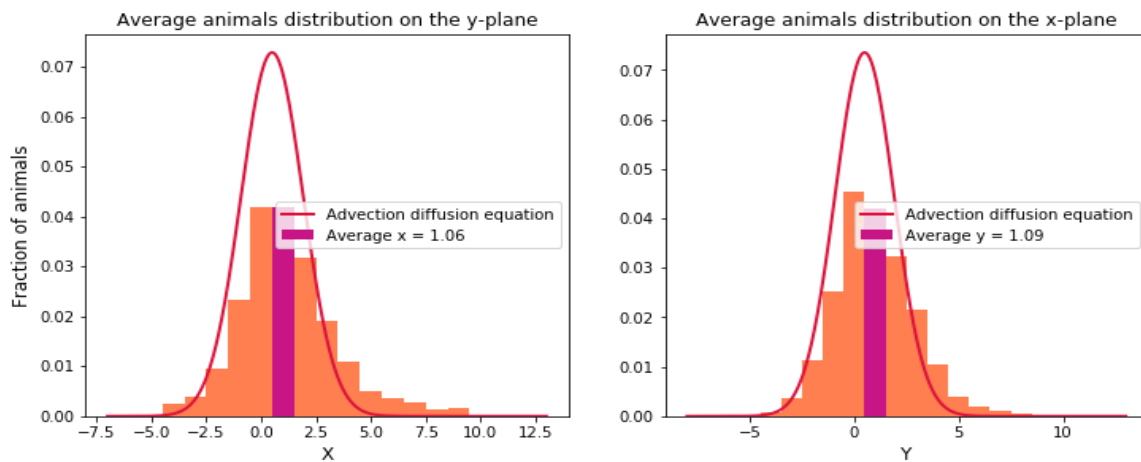


Figure 15: Wildebeest population distribution and average position of walkers on the xy – plane. The orange bars are the actual (empirical data and the red line is the prediction based on the advection-diffusion equation)

Distribution of Animals on the XY plane at $T = 100$

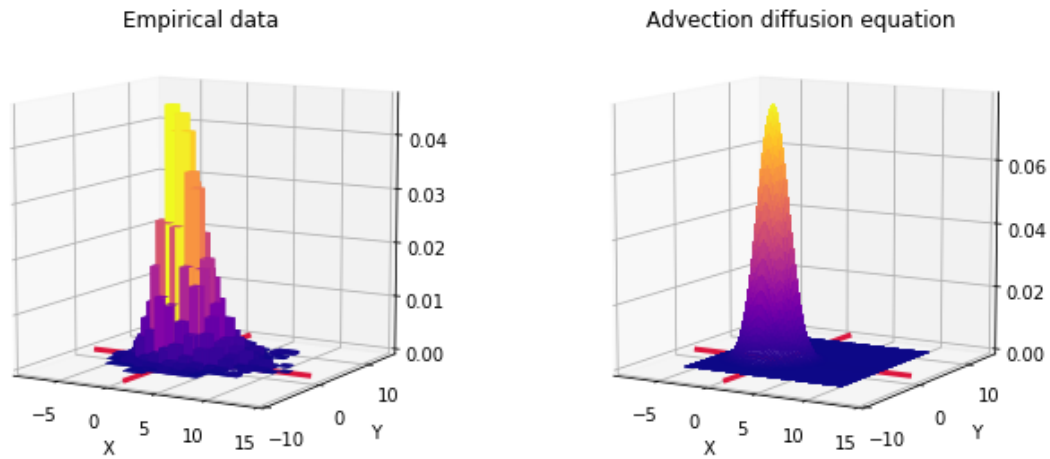


Figure 16: Wildebeest population evolution on the xy – plane

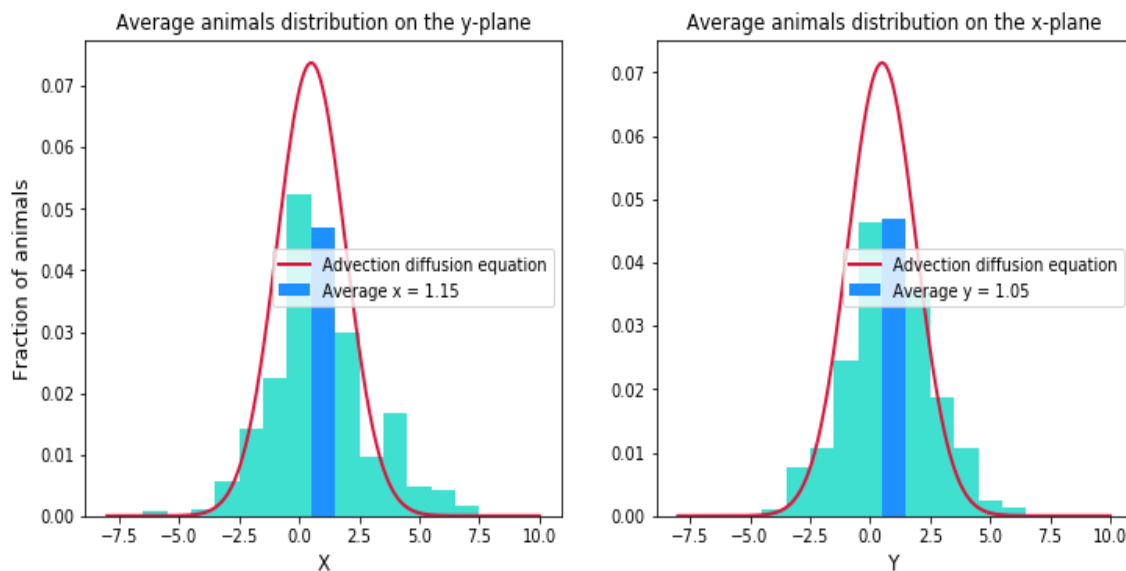


Figure 17: Zebra population distribution and average position of walkers on the xy – plane

The green bars are the actual (empirical data), and the red line is the prediction (based on advection-diffusion equation).

Distribution of Animals on the XY plane at T = 100

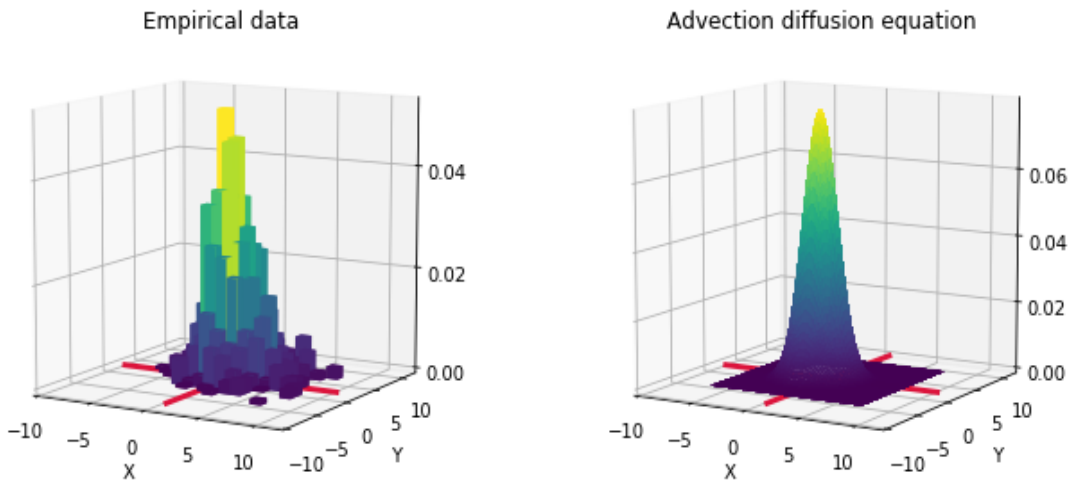


Figure 18: Zebra population evolution on the xy – plane

Figures 16 to 19 illustrate the patterns that diffusion-advection-driven variabilities can form. These are expected movement patterns formed when wildebeests and zebras move via diffusion and advection processes. The empirical data generated random walks, and the advection-diffusion equation produced Gaussian movement patterns.

4.1.3 Wildebeest, Zebra and Lion Prey-Predator Interactions

The prey-predator interaction between wildebeests, zebras and lions was simulated to mimic the real interaction of the three species in the Serengeti ecosystem. The aim was to understand mathematically how the three species interact and the factors that regulate their movement patterns. The current estimates of wildebeests, zebras, and lions abundances of 1.3 million, 200 000 and 3000 were used. In addition, the carrying capacities for wildebeests and zebras of 1.8 million and 300 000 were set for a balanced growing population.

The movement parameters of advection and diffusion were defined to allow animals to navigate freely in the ecosystem (Table 2). Each individual started with their initial position and was allowed to perform a random walk in any preferred direction. Every animal in the population can die by predation or drought at each step, interact with other herbivore partners, or step in any randomly chosen direction. After running the model, the population evolution of walkers (the optimal time step was 100) was recorded. Finally, the model calculates the average position and the advection-diffusion -reaction of the resulting population on the xy – plane.

The distributions of animals on the xy – plane were recorded. These distributions (from the empirical data) on a 3D histogram were plotted using the advection-diffusion reaction equation as a surface. Furthermore, the walker's distributions on 2D histograms along x and y axes with the other coordinates equal to its average position value were plotted.

The effects of diffusion and advection movement parameters on the population dynamics of wildebeests and zebras in response to lion attacks were investigated. The advection and diffusion parameters in different scenarios were varied to mimic the real movement patterns of both prey and predator systems.

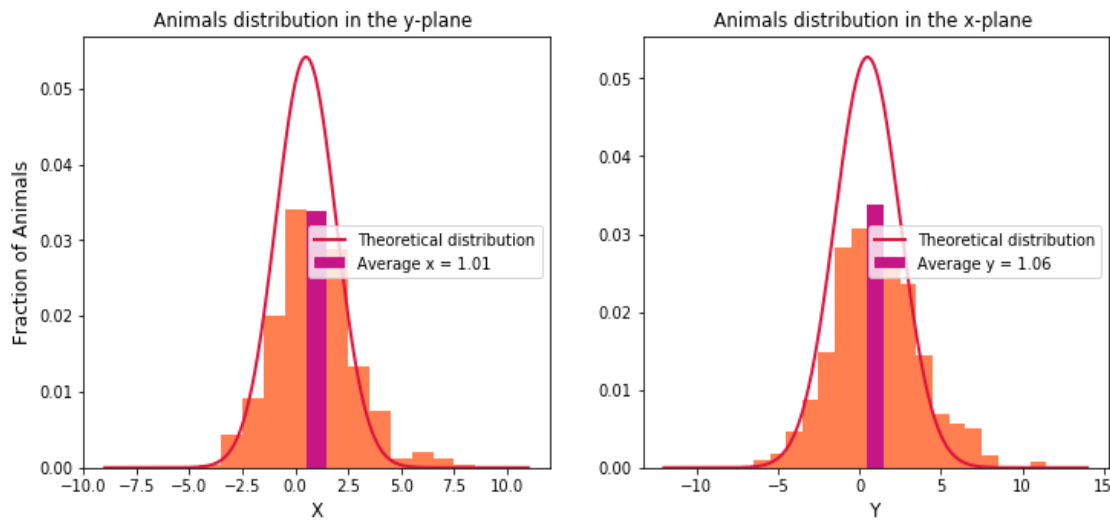


Figure 19: Wildebeest population distribution and average position of walkers on the xy – plane

The orange bars are the actual (empirical) data, and the red line is the prediction based on the advection-diffusion equation.

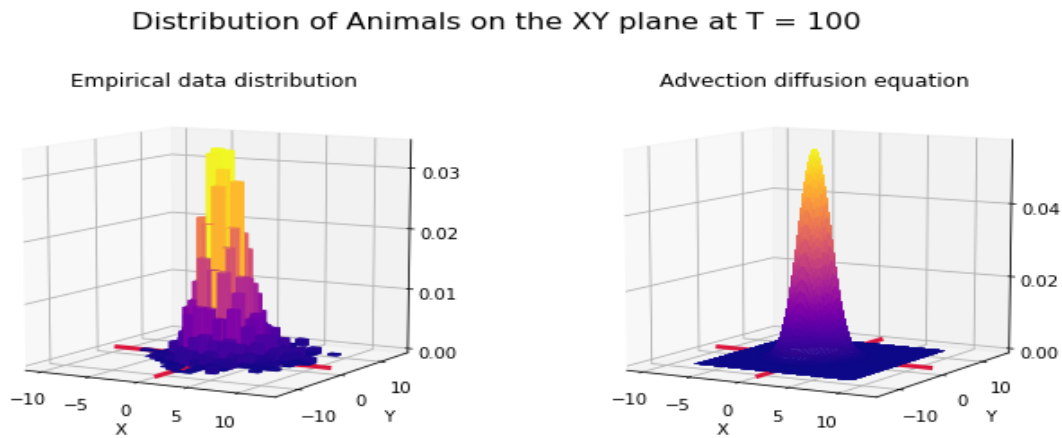


Figure 20: Wildebeest population evolution on the xy – plane

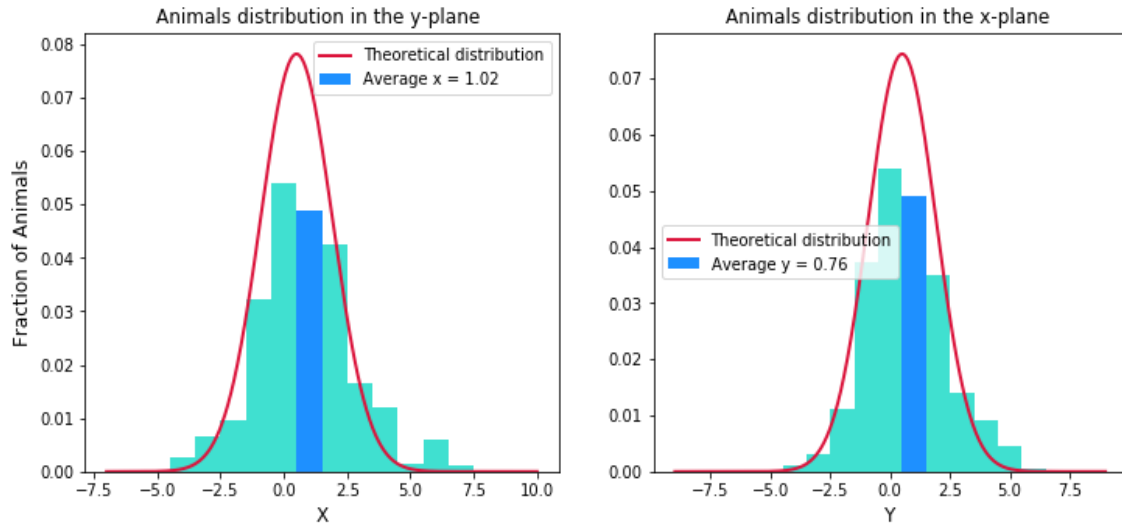


Figure 21: Zebra population distribution and average position of walkers on the xy -plane

The green bars are the actual (empirical) data, and the red line is the prediction based on the advection-diffusion equation.

Distribution of Animals on the XY plane at $T = 100$

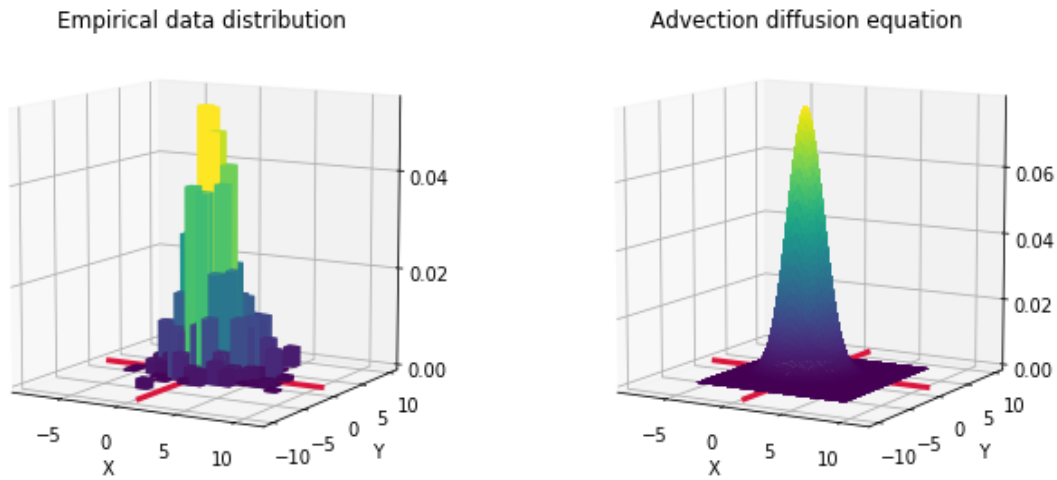


Figure 22: Zebra population evolution on the xy – plane

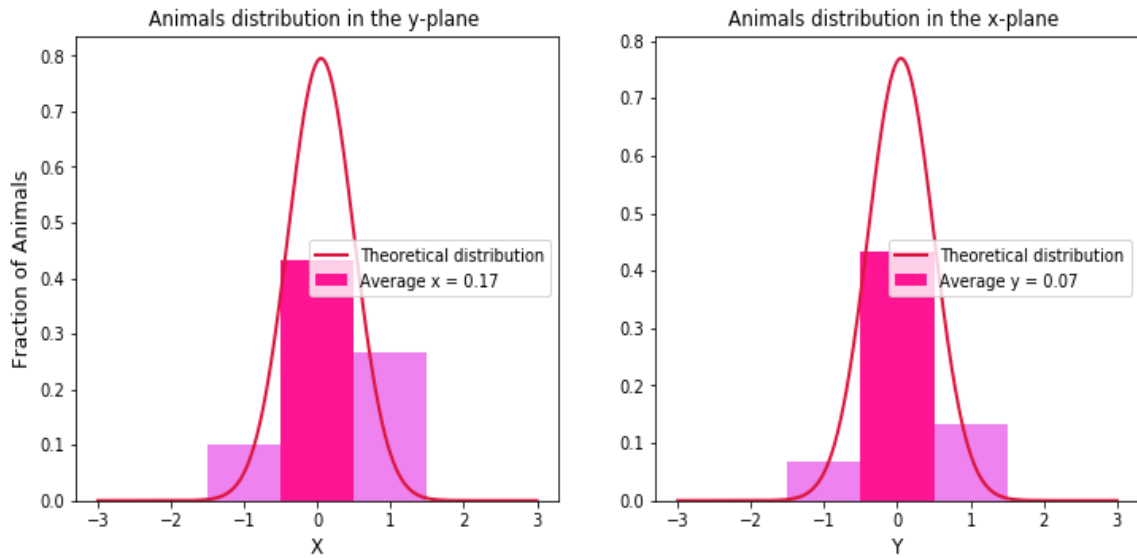


Figure 23: Lion population distribution and average position of walkers on the xy – plane. The violet bars are the actual (empirical) data and the red line is the prediction based on the advection-diffusion equation

Distribution of Animals on the XY plane at $T = 10$

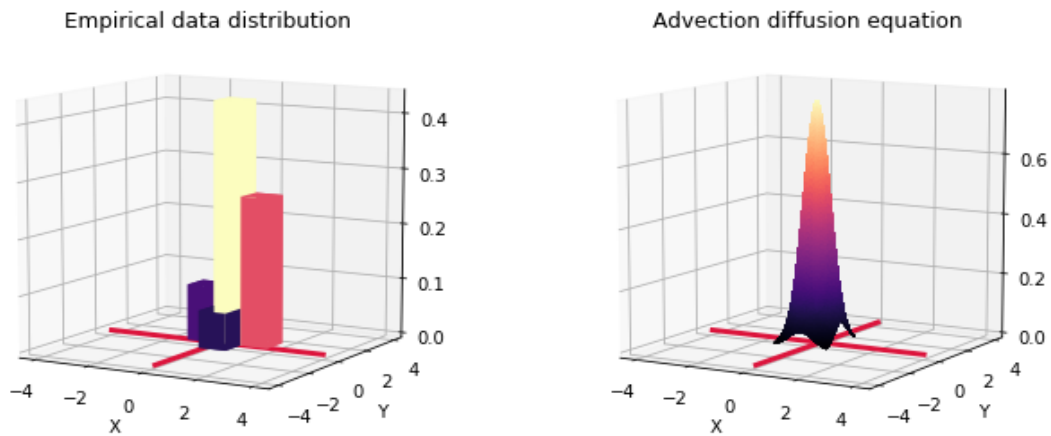


Figure 24: Lion population evolution on the xy – plane

The results show that wildebeest and zebra populations grow logistically to their carrying capacities (Fig. 26), and their movement via diffusion and advection follows a Gaussian distribution.

Drought has been reported in many kinds of literature in the Serengeti ecosystem to affect the survival of the migrating species (Sagamiko *et al.*, 2015). Therefore, the effect of drought on the survival of the interacting species was investigated. Wildebeest and zebra death rates are

normal for moderate drought and predation rates. They allow the species to grow to their carrying capacities (Fig. 26), but the population declines when the ecosystem experiences persistent drought (Fig. 27). This, in turn, affects the species' movement patterns (both advection and diffusion). During the drought seasons, there is increased diffusion (due to random utilization of the available resources) and decreased advection (reduced migration).

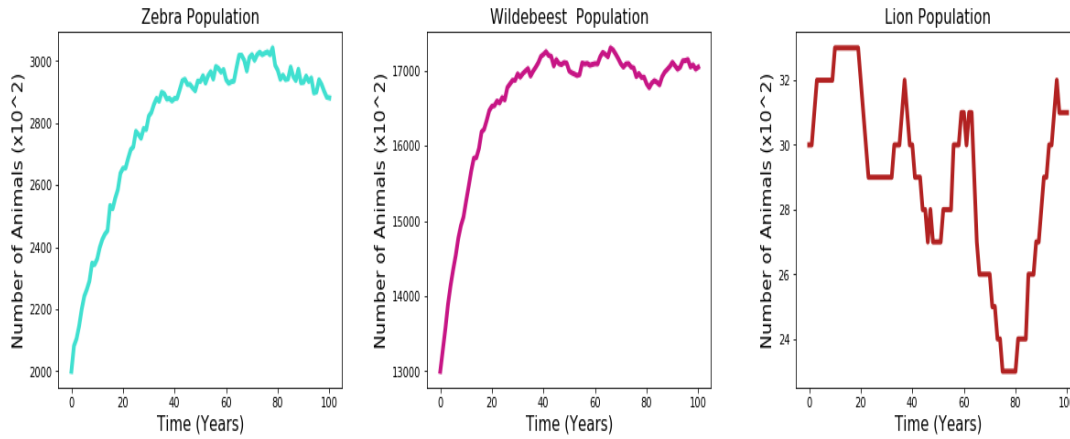


Figure 25: Populations growth of interacting species with time

Table 9: Wildebeest, zebra and lion population in absence of extreme drought

Time	Zebra Number	Wildebeest Number	Lion Number
0	200 000	1 300 000	3000
10	217 500	1 305 000	3034
20	265 400	1 652 400	3080
40	285 007	1 714 700	3013
60	293 900	1 709 100	3045
80	297 200	1 682 300	3105
100	300 000	1 802 300	3150

As the ecosystem experience extreme drought season, there is a decrease in population as shown in Fig. 27 and table 10.

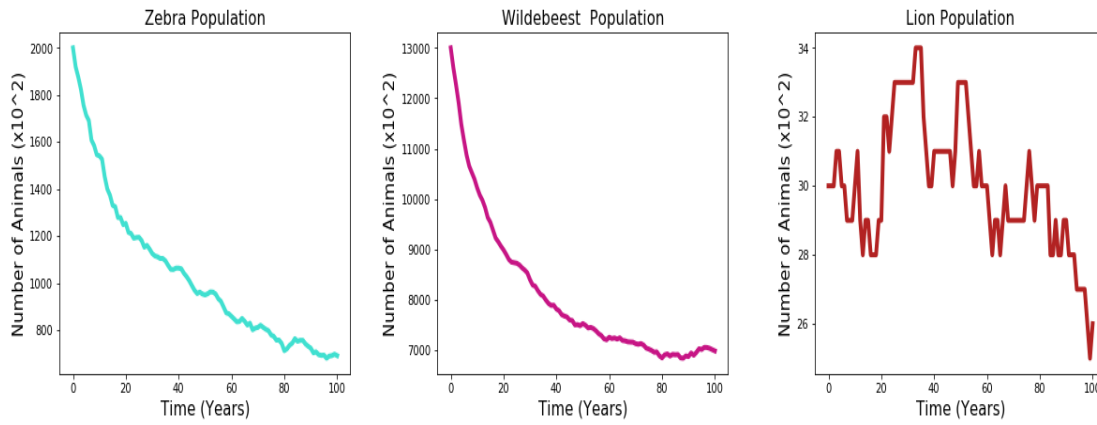


Figure 26: Populations growth of interacting species in the presence of drought with time

Table 10: Wildebeest, zebra and lion population in presence of extreme drought

Time	Zebra Number	Wildebeest Number	Lion Number
0	200 000	1 300 000	3000
10	196 000	1 264 700	2915
20	171 024	1 079 472	2980
40	111 507	704 523	2954
60	104 300	661 900	2876
80	71 300	685 200	2786
100	69 900	699 200	2621

4.2 Discussion

4.2.1 Foraging Processes of Wildebeest

A consistent movement of wildebeests towards specific directions was observed (Fig. 9 to 11). Paths containing a consistent movement in a preferred direction are termed as biased random walks (BRWs) (Fagan *et al.*, 2019). Therefore, the central question addressed the biased random walk in relation to foraging efficiency. The key idea of movement ecology is that animals move in response to internal needs such as hunger and thirst and external needs such as avoiding predators and responding to local landscape variables such as looking for favourable forage targets (Fryxell *et al.*, 2008). Forage resources in the Serengeti ecosystem are heterogeneously distributed in space. Wildebeests (in their herds) take advantage of this spatial distribution of resources by selecting patches of high resource abundance or travelling to locate them to inhabit. In the wet season, when resources are plentiful, wildebeests travel longer distances through both advection and diffusion (Table 5) compared to dry seasons (Table 4) (Tourney *et al.*, 2018;

Hopcraft *et al.*, 2014). During the dry season, forage resources are low; hence they spend more time in areas of low resource abundance to maximize their intake (Ferguson *et al.*, 2018). Therefore, the movement trajectories depicted in Fig. 8 to 11 result from wildebeests responding to their forage needs. As explained in different literature, this wildebeest movement pattern was expected.

Wildebeest in the Serengeti ecosystem are migratory species exhibiting consistent large-scale movements that lead to a net displacement in a unique direction, as evident in Fig. 8 to 11. This is a characteristic of animal migration (Sibert *et al.*, 1999). Wildebeests travel through directed movement parameters and spread across different habitats through diffusive parameters. Therefore, the great migration of wildebeests to different habitats results from directed movements (searching for forage resources) and spreads across habitats to utilize the resources through diffusive trends.

The extent to which wildebeests can be random or dispersive depends on habitat selection, socialization and avoidance of predators (Minasandra & Isvaran, 2020; Qi Ma, 2011; Fortin *et al.*, 2009). Furthermore, wildebeests are foraging animals and usually travel in groups making movement decisions that depend on forage availability and social interactions (Fortin *et al.*, 2009). Therefore, the stability and direction of their group largely rely on the knowledge about the quality and location of the food source, avoidance of predators and the ability of the informed individuals to influence group decisions to move in the desired direction (Couzin *et al.*, 2002).

Wildebeest herds respond to local rules of attraction, repulsion and alignment as they forage (Couzin *et al.*, 2000). With repulsion, animals tend to move away from each other to avoid collisions, come close to their neighbours due to attraction, and navigate from place to place due to alignment forces (Couzin *et al.*, 2000).

As observed in this study, during the dry seasons or when resources are scarce, wildebeests move farther away from each other (increased repulsion) to reduce competition while grazing (Ferguson *et al.*, 2018). In such cases, they spread making short movements to maximize their daily grass intake.

Predation pressure is another factor that affects the foraging efficiency of wildebeests. This occurs when an animal chooses between a patch with good forage while reducing predation risks (Hopcraft, 2010). The fear of predation determines the patch an animal may select and how long it may stay in that area (Ferguson *et al.*, 2018).

Further, the habitat's nature also determines how advective and diffusive factors determine wildebeest forage. Usually, wildebeests avoid dangerous habitats such as river banks or water sources where predators may hide (Mduma, 1994). Instead, wildebeests cross such habitats through drift (higher advective) forces, and they avoid spreading to reduce the chances that prevent herd formation. Therefore, the dispersive and advective components of wildebeest movements help them search for and acquire forage and resources and avoid predators.

Other authors have reported similar observations on foraging efficiency for various herbivore species. For instance, Anderson *et al.* (2016) reported on the spatial distribution of African savannah herbivores. The authors argue that wildebeests were positively associated with normalized difference vegetation index (NDVI) while foraging. This shows that wildebeests make movement decisions depending on the environmental gradients and responses to conspecifics.

The data was analysed for dry (June-October) and wet (November-May) seasons. The locations of species in the respective maps (Fig. 6 to 11) can be used to justify the foraging process in different seasons. The movement parameters (diffusion and advection) can also be used to justify the foraging process seasonally. The concentration of animals on the maps for different seasons could justify the foraging concentration. Generally, the findings of this study about the movement patterns and foraging needs of wildebeest, zebra, and lion are similar to the previous findings for non-mathematics studies like Mduma *et al.* (1996, 1999), Hopcraft *et al.* (2010, 2014), Holdo (2011), Anderson *et al.* (2016), Dutta (2019), Minasandra and Isvaran (2020) and Fagan *et al.* (2019).

It can be concluded that the advection-diffusion equation can explain wildebeest foraging behaviour. The random walk trajectories respond to different habitats, seasons and responses to conspecifics (Hopcraft, 2010). Such changes in behaviour could result from changes in the animals' nutritional requirements at different times of the year as a result of meeting other needs such as calving (Hopcraft, 2010).

4.2.2 Mutualism between Wildebeests and Zebras

The mutualistic behaviour between wildebeests and zebras was modelled, and the major factors limiting their abundance (predation and drought) were determined. It was observed that wildebeest and zebra populations' growth approached their respective carrying capacities, and the absence of one species does not affect the existence of the other. In the absence of predation

and drought, the ecosystem is likely overpopulated by wildebeests and zebras, with wildebeests being the dominant species. This would increase inter-species competition between the two species for the forage resources.

For a balanced ecosystem, predation is important to have all the species survive. The predation and drought rates varied over different scales. It was observed that lion predation (with moderate periodic drought seasons) would allow the ecosystem to have a balanced population of wildebeests and zebras (Fig. 14 and Table 7). Therefore, diffusion and advection processes depicted by wildebeest and zebra from place to place are a response to predation risks and looking for better forage that results in a balanced population. If the Serengeti ecosystem experiences the current trends of predation and periodic drought seasons (Fig. 14), there would be a slow population growth of wildebeests and zebras.

Furthermore, increased drought causes grass scarcity. The drought parameter was varied, and as drought intensity increases, the population of wildebeests and zebras decreases by a large amount (Fig. 15 and Table 8). Therefore, the ecosystem will likely collapse if the drought persists for a few years. Apart from foraging and avoiding predators, these species can potentially make foraging decisions at various spatial scales, including mate search, territorial marking, and looking for calving areas (Mduma, 1996; Hopcraft, 2010).

4.2.3 Wildebeest, Zebra and Lion Prey-Predator Interactions

Wildebeests and zebras aggregate to form herds and move towards a desirable target (Okubo & Levin, 2001). Being selective herbivores, wildebeests and zebras usually travel in herds to seek better forage, avoid predators and interact socially (Fortin, 2009). The variation in wildebeests and zebras in group sizes may be due to environmental conditions, such as the location of food and water and predators (Bonabeau, 1999). During migration, wildebeests and zebras develop migratory corridors containing information on rainfall and the normalized difference vegetation index (NDVI) of plant productivity, suggesting that seasonal fluctuations can account for the migration (Anderson *et al.*, 2016). Forage availability is the main driver of spatial distribution patterns of wildebeests and zebras (Boone, 2018). These prey species choose areas with good forage quality and nutrient content and are free from predators. Therefore, the diffusion and migration of wildebeests, and zebras followed by predators (lions) from place to place within the Serengeti ecosystem result from a search for better forage. These results suggest that good

forage quality, abundance, and avoidance of predators affect the wildebeests and zebras' choices of habitats and their navigation between them.

The lion population is still small in the Serengeti ecosystem (around 3000) (Hopcraft, 2010). This predator population seems stable even during dry seasons (Mduma *et al.*, 1994). However, its distribution does not show a convincing growth (Fig. 24 and 25). More factors need to be studied to identify the reasons for the slow growth of the lion population in the Serengeti ecosystem. When resources are scarce during the dry season, wildebeests and zebras spread more (high diffusion) and are easily caught by lions compared to the wet season (Okubo & Levin, 2001). Hence from Fig. 7 and 8, areas with dense wildebeests suggest the availability of better forage compared to areas with few wildebeests. Further, the Serengeti lions usually hide in places with high prey catchability, such as woody vegetation and near water sources (Anderson *et al.*, 2016).

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study used mathematical models to study wildebeest foraging processes and how it interacts with zebra and lion in the Serengeti ecosystem. The model for wildebeest foraging processes was formulated from the concepts of random walks and jump probabilities. The model analysis showed that the movement patterns of wildebeests could be described by the advection-diffusion equation when they respond to environmental variables such as water and grass.

The second mathematical model was formulated to include another herbivore species, the zebra. The model was further improved by adding one predator (lion) to form the third mathematical model. While these herbivore species grow to their carrying capacities, drought and predation from lions were found to be the major factors that affect their abundances. The dynamical behaviour of migrating wildebeests and zebras followed by lions as predators in the Serengeti ecosystem shows a stable state in the absence of extreme drought. However, in the presence of moderate drought seasons, there is a decreased growth rate of these species, while extreme drought periods drive the species to extinction. Since forage is the main driver of wildebeest and zebra migration, both species evolve migratory pathways containing information on the availability of good forage and areas free from predators. Therefore, the search for better forage results in advection and diffusion of wildebeests and zebras followed by predators from place to place within the Serengeti ecosystem.

Theoretical predictions point out that partial differential equations (PDEs) are useful for examining the interaction between habitat geometry and competitive coexistence (Holmes *et al.*, 1994). The PDEs models developed in the current study have successfully captured the theoretical predictions that movement and diffusion of the two species populations are motivated by a search for better forage and avoidance of predators. The survival of the prey species is primarily affected by the climate (availability of nutritious grass) and predation, particularly from lions.

5.2 Recommendations

Based on the finding described in this study, the study recommends that conservationists and ecologists in different ecosystems gain knowledge of animal behaviour. Thus, this study likely

increases awareness of wildlife behaviour by understanding animals' dynamics, such as animal movement and foraging patterns and the predator-prey relations of these species.

Also, this study serves as reference data for conducting new research or testing the validity of other findings. There are convincing improvements in the management of the protected areas as there is an overall decrease in human activities around the Serengeti ecosystem. However, despite these improvements, the policy-makers (Ministry of Tourism and Natural Resources) are advised to communicate ways of introducing programs that will allow local communities to manage wildlife resources around their villages. This could reduce poaching and other human activity pressure on the protected areas. In addition, wildlife protection awareness campaigns should complement this.

There is a need to conduct special ground surveys to assess the stock of wildebeest and zebra species rather than rely only on an aerial survey.

REFERENCES

- Ahmad, R., & Budin, H. (2012). Stability Analysis of Mutualism Population Model with Time Delay. *International Journal of Mathematical and Computational Sciences*, 6(2), 151-155.
- Akugizibwe, E. (2010). *Modeling and analysis of a two prey-one predator system with harvesting, Holling Type II and ratio-dependent responses* [Masters' thesis, Makerere University]. <https://www.google.com>
- Anderson, T. M., White, S., Davis, B., Erhardt, R., Palmer, M., Swanson, A., Kosmala, M., & Packer, C. (2016). The spatial distribution of African savannah herbivores: Species associations and habitat occupancy in a landscape context. *Philosophical Transactions of the Royal Society: Biological Sciences*, 371(1703), 20150314. <https://doi.org/10.1098/rstb.2015.0314>.
- Auger-Méthé, M., Derocher, A. E., DeMars, C. A., Plank, M. J., Codling, E. A., & Lewis, M. A. (2016). Evaluating random search strategies in three Mammals from distinct feeding guilds. *Journal of Animal Ecology*, 85(5), 1411-1421. <https://doi.org/10.1111/1365-2656>.
- Bonabeau, E., Dagon, L., & Freon, P. (1999). Scaling in animal group-size distributions. *Ecology, Physics*, 96, 4472-4477.
- Boone, R. B., Thirgood, S. J., & Hopcraft, J. G. C. (2006). Serengeti Wildebeest migratory patterns modeled from rainfall and new vegetation growth. *Ecology*, 87(8), 1989-1994.
- Codling, E. A., Plank, M. J., & Benhamou, S. (2008). Random walk models in Biology. *Journal of the Royal Society Interface*, 5, 813-834. <https://doi.org/10.1098/rsif.2008.0014>.
- Davies, N. B., Krebs, J. R., & West, S. A. (2012). *An Introduction to Behavioural Ecology*. A John Wiley & Sons, Ltd., Publication. <https://www.google.com>

- Duncan, P., Foose, T., Gordon, T. J., Gordon, I. J., Gakahu, C. G., & Lloyd, M. (1990). Comparative nutrient extraction from forages by grazing bovids and equids: A test of the nutritional model of equid/bovid competition and coexistence. *Springer-Verlag*, 84, 411-418. <https://doi.org/10.1007/BF00329768>.
- Dutta, K. (2019). Superdiffusive Searching Skill in Animal Foraging. *Discontinuity, Nonlinearity and Complexity*, 8(1), 49-55. <https://doi.org/10.5890/DNC.2019.03.005>
- Fagan, W. F., Hoffman, T., Dahiya, D., Gurare, E., Cantrell, R. S., & Cosner, C. (2019). Improved foraging by switching between diffusion and advection: benefits from movement that depends on spatial context. *Theoretical Ecology*, 13, 127–136. <https://doi.org/10.1007/s1200-019-00434-w>
- Fay, T. H., & Greeff, J. C. (2006). Lion, wildebeest and zebra: A predator-prey model. *Ecological Modelling*, 196(1–2), 237–244. <https://doi.org/10.1016/j.ecolmodel.2006.02.026>.
- Fortin, D., Fortin, M. E., Beyer, H. L., Duchesne, T., Courant, S., & Dancose, K. (2009). Group-Size-Mediated Habitat Selection and Group Fusion- Fission Dynamics of Bison under Predation Risk. *Ecology*, 90(9), 2480–2490.
- Fryxell, J. M., Hazell, M., Borger, L., Dalziel, B. D., Haydon, D. T., Morales, J. M., McIntosh, T., & Rosatte, R. C. (2008). Multiple movement models by large herbivores at multiple spatiotemporal scales. *Proceedings of the National Academy of Sciences*, 10(49), 19114-19119. <https://doi.org/10.1073/pnas.0801737105>
- Grange, S., Duncan, P., Gaillard, J. M., Sinclair, A. R. E., Gogan, P. J. P., Packer, C., Hofer, H., & East, M. (2004). What limits the Serengeti zebra population? *Oecologia*, 140(3), 523–532. <https://doi.org/10.1007/s00442-004-1567-6>.

- Gueron, S., Levin, S. A., & Rubenstein, D. I. (1996). The dynamics of herds: From individuals to aggregations. *Journal of Theoretical Biology*, 182(1), 85–98. <https://doi.org/10.1006/jtbi.1996.0144>.
- Hagen, M., Kissling, W. D., Rasmussen, C., De Aguiar, M. A. M., Brown, L. E., Carstensen, D. W., Do-Santos, I. A., Dupont, Y. L., Edwards, F. K., Genini, J., Guimaraes, P. R., Jenkins, G. B., Jordano, P., Kaiser-Bunbury, C. N., Ledger, M., Maia, K. P., Marquitti, F. M. D., McLaughlin, O., Morellato, L. P. C., & O'Gorman, E. J. (2012). Biodiversity, Species Interactions and Ecological Networks in a Fragmented World. *Advances in Ecological Research*, 46, 89–120. <https://doi.org/10.1016/B978-0-12-396992-7.00002-2>
- Hogan, B. G., Hildenbrandt, H., Scott-Samuel, N. E., Cuthill, I. C., & Hemelrijk, C. K. (2017). The confusion effect when attacking simulated three-dimensional starling flocks. *Royal Society Open Science*, 4(1), 1–9. <https://doi.org/10.1098/rsos.160564>.
- Holdo, R. M., Fryxell, J. M., Sinclair, A. R. E., Dobson, A., & Holt, R. D. (2011). Predicted impact of barriers to migration on the Serengeti wildebeest population. *PLoS One*, 6(1), 1-7. <https://doi.org/10.1371/journal.pone.0016370>.
- Holmes, E. E., Lewis, M. A., Banks, J. E., & Veit, R. R. (1994). Partial differential equations in ecology: Spatial interactions and population dynamics. *Ecology*, 75(1), 17–29.
- Hopcraft, J. G. C. (2010). *Ecological implications of food and predation risk for herbivores in the Serengeti*. Groningen: s.n. <https://www.google.com>
- Hopcraft, J. G. C., Morales, J. M., Beyer, H. L., Borner, M., Mwangomo, E., Sinclair, A. R. E., Olf, H., & Haydon, D. T. (2014). Competition, predation and migration: Individual choice patterns of Serengeti migratnts catured by hierarchical models. *Ecological Monographs*, 84(3), 355-372.

- Huth, A., & Wissel, C. (1992). The simulation of the movement of fish schools. *Journal of Theoretical Biology*, 156, 365–385.
- Ikanda, D., & Packer, C. (2008). Ritual vs. retaliatory killing of African lions in the Ngorongoro Conservation Area, Tanzania. *Endangered Species Research*, 6(1), 67–74. <https://doi.org/10.3354/esr00120>.
- Kar, T. K., & Chaudhuri, K. S. (2002). Harvesting in a two prey- one predator fishery: A Bio economic model. *Anziam Journal*, 45, 443–456.
- Kideghesho, J. R. (2010). ‘Serengeti shall not die’: Transforming an ambition into a reality. *Tropical Conservation Science*, 3, 228–248.
- Kittle, M. A., Bukombe, J. K., Sinclair, A. R. E., Mduma, S. A. R., & Fryxell, J. M. (2016). Landscape-level movement patterns by lions in western Serengeti: Comparing the influence of interspecific competitors, habitat attributes and prey availability. *Journal of Movement Ecology*, 4(17), 1-18. DOI 10.1016/j.mcm.2009.12.010.
- Kurowski, L., Krause, A. L., Mizuguchi, H., Grindrod, P., & Gorder, R. A. (2017). Two-species Migration and Clustering in Two-Dimensional Domains. *Society for Mathematical Biology*, 79, 2303-2333. DOI 10.1007/s11538-017-0331-0.
- Lee, S. (2007). *Analysis of Dynamics in Two-prey One-predator model: Effects of remained Carcass*. <https://www.google.com>
- Lian, X., Yue, Y., & Wang, H. (2012). Pattern Formation in a Cross-Diffusive Ratio Dependent Prey-Predator Model. *Applied Mathematics and Computation*, 331, 307-318. Doi: 10.1155/2012/814069.
- Liu, P. (2010). Analysis of a predator prey model with both diffusion and migration. *Mathematical and Computer Modelling*, 51, 1064–1070.

- Massawe, E. T., Maliti, H. T., Bakari, S. K., & Kija, H. (2010). *Aerial Census in the Serengeti Ecosystem*. <https://www.google.com>
- Mduma, S. A. R. (1996). *Serengeti Wildebeest Population Dynamics: Regulation, Limitation and implication for Harvesting* [PhD Thesis, University of British Columbia]. <https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0099202>.
- Mduma, S. A. R., Sinclair, A. R. E., & Hilborn, R. (1999). Food Regulates the Serengeti Wildebeest: A 40 year Record. *Journal of Animal Ecology*, 68(6), 1101–1122.
- Menard, C., Duncan, P., Fleurance, G., Georges, J. Y., & Lila, M. (2002). Comparative foraging and nutrition of horses and cattle in European wetlands. *Journal of Applied Ecology*, 39(1), 120–133. <https://doi.org/10.1046/j.1365-2664.2002.00693.x>.
- Murthy, B., Srinivas, M., & Narasimhulu, Y. (2017). A Three Species Prey-Predator Holling Type-II Non Autonomous Discrete Model. *Indian Journal of Science and Technology*, 10(22), 1–94. DOI: 10.17485/ijst/2017/v10i22/108930.
- Okubo, A., & Levin, S. A. (2001). *Diffusion and Ecological Problems: Modern perspectives*. Springer. <https://www.google.com>
- Plank, M. J., & James, A. (2008). Optimal Foraging: Levy Pattern or Process? *Journal of the Royal Society*, 5, 1077–1086. <https://doi.org/10.1098/rsif.2008.0006>.
- Pyke, G. (2019). *Animal movements: An optimal foraging approach*. <https://www.google.com>
- Qiao, M., Liu, A., & Forsys, U. (2014). Qualitative Analysis for a Reaction-Diffusion Predator-Prey Model with Disease in the Prey Species. *Journal of Applied Mathematics*, 2014, 1-9. <https://doi.org/http://dx.doi.org/10.1155/2014/236208>.
- Raichlen, D. A., Wood, B. M., Gordon, A. D., Mabulla, A. Z. P., Marlowe, F. W., & Pontzer, H. (2014). Evidence of Levy walk foraging patterns in human hunter-gatherers. *Pnas*,

111(2), 728-733. [https://doi.org/ 10.1073/ pnas.1318616111](https://doi.org/10.1073/pnas.1318616111).

Reynolds, A., Ceccon, E., Baldauf, C., Medeiros, T. K., & Miramontes, O. (2018). Levy foraging patterns of rural humans. *PLoS One*, 13(6), 1-16. [https://doi.org/ 10.1371/journal.pone.0199099](https://doi.org/10.1371/journal.pone.0199099).

Reynolds, C. W. (1987). Flocks, herds and schools: a distributed behavioral model. *Computers & Graphics*, 21, 25–34.

Rice, J. A., Miller, T. J., Rose, K. A., Crowder, L. B., Marschall, E. A., Trebitz, A. S., & DeAngelis, D. L. (2011). Growth Rate Variation and Larval Survival: Inferences from an Individual-Based Size-Dependent Predation Model. *Canadian Journal of Fisheries and Aquatic Sciences*, 73(1), 125–139. [https:// doi. org/ https:// doi. org/10. 1139/cjfas-2015-0102](https://doi.org/10.1139/cjfas-2015-0102).

Rosenthal, S. B., Twomey, C. R., Hartnett, A. T., Wu, H. S., & Couzin, I. D. (2015). Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion. *Proceedings of the National Academy of Sciences*, 112(15), 4690–4695. [https://doi.org/10. 1073/pnas.1420068112](https://doi.org/10.1073/pnas.1420068112).

Sagamiko, T. D., Shaban, N., Nahonyo, C. L., & Makinde, O. D. (2015). Optimal Control of a Threatened Wildebeest-Lion Prey-Predator System in the Serengeti Ecosystem. *Applied and Computational Mathematic*, 4(4), 296–312. [https:// doi. org/ 10. 11648/j. acm. 20150404.18](https://doi.org/10.11648/j.acm.20150404.18).

Schaller, G. B. (1977). *The Serengeti lion: A study of predator-prey relations*. University of Chicago, Chicago. <https://www.google.com>.

Sibert, J. R., Hampton, J., Fournier, D. A., & Bills, P. J. (1999). An advection–diffusion–reaction model for the estimation of fish movement parameters from tagging data,

- with application to skipjack tuna (*Katsuwonus pelamis*). *Canadian Journal of Fisheries and Aquatic Sciences*, 56, 925–938.
- Sinclair, A. R. E. (1977). The African Buffalo: A study of Resource Limitation of Population. *Journal of Mammalogy*, 60(2), 438–439. <https://doi.org/10.2307/1379830>.
- Sinclair, A. R. E., Packer, C., Mduma, S.A R., & Fryxell, J. M. (2008). *Serengeti III: Human Impacts on Ecosystems Dynamics*. The University of Chicago press. <https://doi.org/10.1093/icb/icp004>
- Takeuch, Y. (1996). *Global dynamical properties of Lotka-Volterra systems*. World scientific publishing Co. Ltd, Singapore.
- Taylor-Brown, A., Booth, R., Gillett, A., Mealy, E., Ogbourne, S. M., Polkinghorne, A. & Conroy, G. C. (2019). The impact of human activities on Australian wildlife. *PLoS One*, 14(1), 1-28. <https://doi.org/10.1371/journal.pone.0206958>.
- Torney, C. J., Hopcraft, J. G. C., Morrison, T. A., Couzin, I. D., & Levin, S. A. (2018). From single steps to mass migration: The problem of scale in the movement ecology of the Serengeti wildebeest. *Philosophical Transactions of the Royal Society: Biological Sciences*, 373(1746), 1-9.
- Torney, C. J., Lloyd-Jones, D. J., Chevallier, M., Moyer, D. C., Maliti, H. T., Mwita, M., Kohi, E. M., & Hopcraft, J. G. C. (2019). A comparison of deep learning and citizen science techniques for counting wildlife in aerial survey images. *Methods in Ecology and Evolution*, 10(6), 779-787. <https://doi.org/10.1111/2041-210X.13165>.
- Upadhyay, R. K., & Dubey, B. (2004). Persistence and Extinction of One-Prey and Two-Predators System. *Non Linear Modeling and Control*, 9(4), 307–329.
- Wang, C., Chang, L., & Liu, H. (2016). Spatial Patterns of a Predator-prey System of Leslie Type with Time Delay. *PloS One*, 11(3), e0150503. Doi:10.10.1371/journal.pone.

- Wang, Y. (2009). Asymptoticbehaviour of solutions of Lotka-Volterra mutualism reaction-diffusion system with time delays. *Computerrs and Mathematics with Applications PloS One*, 58, 597-604.
- Yamada, H., Nakagaki, T., Baker, R. E., & Maini, P. K. (2007). Dispersion relation in oscillatory reaction-diffusion systems with self-consistent flow in true slime mold, *Journal of Mathematical Biology*, 54 (2007), 745–760.

APPENDICES

Appendix 1: Analysis for full wet season

```
1 # Analysis for full wet season
2 import random
3 import numpy as np
4 import pandas as pd
5 from math import floor, ceil, isnan
6
7 import matplotlib.pyplot as plt
8 from matplotlib import cm
9 from mpl_toolkits.mplot3d.axes3d import Axes3D
10
11
12 class Population:
13     def __init__(self, filename):
14
15         """ Initilaize Population instance with default values.
16         Calls read_data() function that read data from the file. """
17
18         self.N_walkers = 0
19
20         self.__time = []
21         self.__time_str = []
22         self.__evolution_history = []
23
24         self.delta_xy = None
25         self.delta_t = None
26
27         self.read_data(filename)
28
29         self.diapason = None
30         self.N_points = None
31
32         self.__walkers_position = []
33         self.__walkers_distribution = []
34
35         self.p = 0 # Left
36         self.q = 0 # right
37         self.w = 0 # up
38         self.z = 0 # down
39
40         self.x0 = None
41         self.y0 = None
42         self.current_t = 0
43
44
45     def read_data(self, filename):
46
47         """ Open file 'filename' and read data from this file.
48         Calculate average delta x, delta y and delta t. Put data related to each animal into separate arrays.
49         Find the unique animals id and years. """
50
51         with open(filename, 'r') as file:
52             data = pd.read_csv(filename)
53             self.id = data.iloc[:, 0].values
54             xy = data.iloc[:, 1:3].values
55             time = data.iloc[:, 3].values
56             delta_t = data.iloc[:, 7].values
57
58         # calculate delta t
59         delta_t = [float(delta_t[i]) for i in range(len(delta_t))]
60
61         total_delta = 0
62         I = 0
63         for i in range(len(delta_t)):
64             if not isnan(delta_t[i]):
65                 total_delta += delta_t[i]
66                 I += 1
67         self.delta_t = total_delta / I
68
69         # separate data for different animals
70         last_id = -999
71         for i in range(len(self.id)):
72             if self.id[i] != last_id:
73                 self.N_walkers += 1
74                 last_id = self.id[i]
75                 self.__evolution_history.append([])
76                 self.__time_str.append([])
77                 self.__time.append([0])
78             else:
79                 self.__time[-1].append(self.__time[-1][-1] + delta_t[i-1])
80
81                 self.__evolution_history[-1].append(xy[i])
82                 self.__time_str[-1].append(time[i])
83
84         self.id = np.unique(self.id)
85
86         self.years = [int(time[i].split('/')[2][:4]) for i in range(len(time))]
87         self.years = np.unique(self.years)
```

```

90 # calculate delta x, delta y
91 total_steps = 0
92 total_X = 0
93 total_Y = 0
94
95 for animal in range(self.N_walkers):
96     total_X += sum([abs(self.__evolution_history[animal][i][0] - self.__evolution_history[animal][i-1][0]) for i in range(1, len(self.__evolution_history[animal]))])
97     total_Y += sum([abs(self.__evolution_history[animal][i][1] - self.__evolution_history[animal][i-1][1]) for i in range(1, len(self.__evolution_history[animal]))])
98     total_steps += len(self.__evolution_history[animal])
99
100 self.delta_xy = [total_X / total_steps, total_Y / total_steps]
101
102
103 def __point_index(self, i):
104     """ Utility. Return x, y index of point where walker with the i index is placed """
105
106     return int(round((self.__walkers_position[i, 0] - self.diapason[0][0]) / self.delta_xy[0])), int(round((self.__walkers_position[i, 1] - self.diapason[0][1]) / self.delta_xy[1]))
107
108
109 def xy_index(self, x = 0, y = 0, axis = None):
110     """ Utility. Return x, y index of the point with any given coordinates x, y,
111     or only x (axis = 0) or y (axis = 1) coordinate.
112     If these coordinated are out of diapason occupied by walkers, return None """
113
114     if axis == 0:
115         y = self.diapason[0][1]
116     if axis == 1:
117         x = self.diapason[0][0]
118
119     if x < self.diapason[0][0] - self.delta_xy[0] or x > self.diapason[1][0] + self.delta_xy[0] or y < self.diapason[0][1] - self.delta_xy[1] or y > self.diapason[1][1] + self.delta_xy[1]:
120         return None
121
122     return int(round((x - self.diapason[0][0]) / self.delta_xy[0])), int(round((y - self.diapason[0][1]) / self.delta_xy[1]))
123
124
125 def evaluate_distribution(self, year):
126     """ Utility. Calculate number of walkers in each delta_x * delta_y interval
127     and x,y - diapason that animals occupy. Consider only the part of data for the given to the function year.
128     If there are less than 10 data point return False, otherwise return True -- calculation will be made for this year's
129     Calculate p q w z probabilities. """
130
131     self.__walkers_position = []
132     self.__norm_animal = []
133
134     # calculate walkers position for given year
135     # calculate probabilities p, q, w, z
136     total_P = 0
137     total_Q = 0
138     total_W = 0
139     total_Z = 0
140     total_steps = 0
141
142     for animal in range(self.N_walkers):
143         years = [int(date.split('/')[2][:4]) for date in self.__time_str[animal]]
144
145         for j in range(len(years)):
146             if years[j] == year:
147                 y_first = j
148                 for j in range(y_first, len(years)):
149                     if years[j] != year:
150                         y_last = j
151                         break
152             else:
153                 y_last = len(years)
154
155             self.__walkers_position += self.__evolution_history[animal][y_first : y_last]
156             self.__norm_animal.append(len(self.__evolution_history[animal][y_first : y_last]))
157             for i in range(y_first + 1, y_last):
158                 if self.__evolution_history[animal][i][0] - self.__evolution_history[animal][i - 1][0] > 0:
159                     total_Q += 1
160                 elif self.__evolution_history[animal][i][0] - self.__evolution_history[animal][i - 1][0] < 0:
161                     total_P += 1
162                 if self.__evolution_history[animal][i][1] - self.__evolution_history[animal][i - 1][1] > 0:
163                     total_W += 1
164                 elif self.__evolution_history[animal][i][1] - self.__evolution_history[animal][i - 1][1] < 0:
165                     total_Z += 1
166             total_steps += y_last - y_first - 1
167
168             break
169
170     self.current_t = 365*24
171
172     self.p = total_P / total_steps # p + q <= 1 and w + z <= 1 separately
173     self.q = total_Q / total_steps # because animal make moves along both x and y direction during each step of time
174     self.w = total_W / total_steps
175     self.z = total_Z / total_steps
176

```

```

182         if len(self.__walkers_position) < 10:
183             return False
184
185         self.__walkers_position = np.array(self.__walkers_position)
186
187         self.diapason = [[int(floor(min(self.__walkers_position[:, 0])/self.delta_xy[0])) * self.delta_xy[0], int(floor(min(
188             [int(ceil(max(self.__walkers_position[:, 0]) / self.delta_xy[0])) * self.delta_xy[0], int(ceil(max(self.__walkers_
189
190         self.N_points = (int(round((self.diapason[1][0] - self.diapason[0][0])/self.delta_xy[0])) + 1,\
191             int(round((self.diapason[1][1] - self.diapason[0][1])/self.delta_xy[1])) + 1)
192
193         self.__walkers_distribution = np.zeros(shape = self.N_points)
194
195         i = 0
196         for animal in range(len(self.__norm_animal)):
197             for j in range(self.__norm_animal[animal]):
198                 self.__walkers_distribution[self.__point_index(i)] += 1 / self.__norm_animal[animal] / self.N_walkers
199                 i += 1
200
201         return True
202
203
204     def __axes(self, axis = 2):
205
206         """Utility. Return array of x or y values in diapason that is occupied by animals
207         with the step of delta_x or delta_y; if axis = 0 it returns x-axis,
208         axis = 1 -- returns y-axis, otherwise return (x,y) coordinates of all points within the diapason"""
209
210         if axis == 0 or axis == 1:
211             return [ self.diapason[0][axis] + i * self.delta_xy[axis] for i in range(self.N_points[axis])]
212         else:
213             x_all = [self.diapason[0][0] + i * self.delta_xy[0] for i in range(self.N_points[0]) for j in range(self.N_points[1])]
214             y_all = [self.diapason[0][1] + j * self.delta_xy[1] for i in range(self.N_points[0]) for j in range(self.N_points[1])]
215             return x_all, y_all
216
217
218     def __average_bin(self):
219
220         """ Utility. Return (x, y) indices of (delta_x, delta_y) 2D interval
221         that contains average position of animals."""
222
223         average = self.average_position()
224         axis = self.__axes(0), self.__axes(1)
225
226         average_bin = [0, 0]
227         for j in [0, 1]:
228             for i in range(self.N_points[j]):
229                 if average[j] <= axis[j][i]:
230                     break
231             if axis[j][i] - average[j] <= average[j] - axis[j][i-1]:
232                 average_bin[j] = i
233             else:
234                 average_bin[j] = i-1
235
236         return average_bin
237
238
239     def get_distribution_2D(self):
240
241         """ Returns distribution of animals on XY plane as a function of x, y.
242         This function distribution_2D() works for any x, y (within and out the diapason occupied by animals)"""
243
244         def disitrbution_2D(x, y):
245             indices = self.xy_index(x, y)
246             if indices == None:
247                 return 0
248             return self.__walkers_distribution[indices]
249
250         return disitrbution_2D
251
252
253     def distribution_1D(self, x, axis):
254
255         """ Returns 1D distribution of animals along x (axis = 0) or y (axis = 1) axis.
256         Sum up number of animals with along othe direction."""
257
258         indices = self.xy_index(x, axis = axis)
259         if indices == None:
260             return 0
261         if axis == 0:
262             return sum(self.__walkers_distribution[indices[0], :])
263         if axis == 1:
264             return sum(self.__walkers_distribution[:, indices[1]])
265
266
267     def average_position(self):
268
269         """ Retuns avarage position of the animals along x and y axis """
270

```



```

271     sum = [0, 0]
272     #for i in range(self.N_walkers):
273     for i in range(len(self.__walkers_position)):
274         for j in [0, 1]:
275             sum[j] += self.__walkers_position[i, j]
276
277     for j in [0, 1]:
278         #sum[j] *= self.delta_xy[j] / self.N_walkers
279         sum[j] /= len(self.__walkers_position)
280     return sum
281
282
283 def calculate_adv_diff_parameters(self):
284
285     """ Calculates advection-diffusion equation parameters """
286
287     self.D_x = (self.p + self.q) * self.delta_xy[0] ** 2 / (2 * self.delta_t)
288     self.D_y = (self.w + self.z) * self.delta_xy[1] ** 2 / (2 * self.delta_t)
289     self.u_x = (self.p - self.q) * self.delta_xy[0] / self.delta_t
290     self.u_y = (self.w - self.z) * self.delta_xy[1] / self.delta_t
291
292     self.x0 = self.average_position()[0] - self.u_x * self.current_t
293     self.y0 = self.average_position()[1] - self.u_y * self.current_t
294
295     print(' : \t D_x = {:.2f} \t D_y = {:.2f} \t u_x = {:.2f} \t u_y = {:.2f}'.format(self.D_x, self.D_y, self.u_x, self
296
297
298 def advection_diffusion(self, x, axis = 0):
299
300     """ Return advection-diffusion equation solution along axis x (0) or y(1). """
301
302     if axis == 0:
303         return np.exp(-(x - self.u_x * self.current_t - self.x0)**2 / (4 * self.D_x * self.current_t)) / (4 * np.pi * np.
304     else:
305         return np.exp(-(x - self.u_y * self.current_t - self.y0)**2 / (4 * self.D_y * self.current_t)) / (4 * np.pi * np.
306
307
308 def plot_distribution2D(self, save_figure = False, figname = "distribution2D"):
309
310     """ Plot distribution of the animals on XY plane as 3D histogram """
311
312     fig = plt.figure()
313     fig.suptitle('Distribution of the Walkers on the XY plane', fontsize = 16)
314
315     ax = fig.add_subplot(1, 1, 1, projection='3d')
316     ax.set_title('Fraction of Walkers')
317     distribution = self.get_distribution_2D()
318     x, y = self.__axes(2)
319     z = [distribution(x[i], y[i]) for i in range(len(x))]
320
321     while True:
322         for i in range(len(z)):
323             if i >= len(z):
324                 break
325             if z[i] == 0:
326                 del z[i], x[i], y[i]
327             else:
328                 break
329
330     cmap = cm.get_cmap('viridis')
331     max_height = max(z)
332     colormap = [cmap(k / max_height) for k in z]
333
334     z_bottom = np.zeros_like(z)
335     ax.bar3d(x, y, z_bottom, self.delta_xy[0], self.delta_xy[1], z, color = colormap, antialiased = True, shade = False)
336
337     ax.set_xlabel ("X")
338     ax.set_ylabel ("Y")
339     ax.view_init(elev=10.)
340     plt.locator_params(nbins = 5)
341
342     if save_figure:
343         fig.savefig(figname + '.png')
344
345     plt.close()
346
347
348 def plot_distributions1D(self, save_figure = False, figname = 'distribtution'):
349
350     """ Plot the animals's distribution 2D histograms along x and y axis
351     with the sum of animals number along other axis. """
352
353     x = self.__axes(0)
354     z1 = [self.distribution_1D(x[i], axis = 0) for i in range(len(x))]
355
356     y = self.__axes(1)
357     z2 = [self.distribution_1D(y[i], axis = 1) for i in range(len(y))]
358

```

```

359 fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (12, 5))
360
361 ax1.bar(x, z1, width = self.delta_xy[0], color = 'dodgerblue', label = 'Animals distribution')
362 ax2.bar(y, z2, width = self.delta_xy[1], color = 'dodgerblue', label = 'Animals distribution')
363
364 x_av = self.average_position()[0]
365 y_av = self.average_position()[1]
366 av_xbin = x[self.__average_bin()[0]]
367 av_ybin = y[self.__average_bin()[1]]
368 z_av1 = z1[self.xy_index(av_xbin, axis = 0)[0]]
369 z_av2 = z2[self.xy_index(av_ybin, axis = 1)[1]]
370 ax1.bar(av_xbin, z_av1, width = self.delta_xy[0], label = 'Average x = {:.2f}'.format(x_av), color = 'blue')
371 ax2.bar(av_ybin, z_av2, width = self.delta_xy[1], label = 'Average y = {:.2f}'.format(y_av), color = 'blue')
372
373 norm1 = self.delta_xy[0]
374 norm2 = self.delta_xy[1]
375
376 x_std = np.linspace(min(x) - 2*self.delta_xy[0], max(x) + 2*self.delta_xy[0], 10**3)
377 y_std = np.linspace(min(y) - 2*self.delta_xy[1], max(y) + 2*self.delta_xy[1], 10**3)
378
379 try:
380     z1_std = [norm1 * self.advection_diffusion(_x, axis = 0) for _x in x_std]
381     z2_std = [norm2 * self.advection_diffusion(_y, axis = 1) for _y in y_std]
382 except:
383     print("WARNING: invalid values encountered in advection_diffusion()")
384     z1_std = [0 for _ in x_std]
385     z2_std = [0 for _ in y_std]
386
387 ax1.plot(x_std, z1_std, linewidth = 2, label = 'Advection-diffusion \nequation', color = 'crimson')
388 ax2.plot(y_std, z2_std, linewidth = 2, label = 'Advection-diffusion \nequation', color = 'crimson')
389
390 ax1.set_title('Animals distribution along X axis')
391 ax2.set_title('Animals distribution along Y axis')
392
393 ax1.set_xlabel('X', fontsize = 12)
394 ax2.set_xlabel('Y', fontsize = 12)
395 ax1.set_ylabel('Fraction of Animals', fontsize = 12)
396
397 plt.locator_params(axis="x", nbins=5)
398 plt.locator_params(axis="y", nbins=5)
399
400 ax1.legend()
401 ax2.legend()
402
403 if save_figure:
404     fig.savefig(filename + '.png')
405
406 plt.close()
407
408 return 0
409
410
411 def plot_evolution(self, animals_plot = None, save_figure = False, filename = "walkers_evolution"):
412
413     """ Plot animal migration path as function of time, x and y coordinate separately."""
414
415     if animals_plot == None or animals_plot > self.N_walkers:
416         animals_plot = self.N_walkers
417
418     for animal in range(animals_plot):
419
420         fig, axs = plt.subplots(2, figsize = (6, 7))
421         fig.suptitle('Migration of Animal {} x and y Coordinates '.format(self.id[animal]))
422
423         evolution = np.array(self.__evolution_history[animal])
424         axs[0].plot(self.__time[animal], evolution[:, 0], linewidth = 2)
425         axs[1].plot(self.__time[animal], evolution[:, 1], linewidth = 2)
426
427         axs[0].set_xlabel('Time (hours)')
428         axs[1].set_xlabel('Time (hours)')
429
430         axs[0].grid()
431         axs[1].grid()
432
433         if save_figure:
434             fig.savefig(filename + '_' + self.id[animal] + '.png')
435
436         plt.close()
437
438
439 def plot_evolution_XY(self, animals_plot = None, save_figure = False, filename = "walkers_evolutionXY"):
440
441     """ Plot migration path of animals on the XY plane during all steps of time. """
442
443     if animals_plot == None or animals_plot > self.N_walkers:
444         animals_plot = self.N_walkers
445
446     fig, ax = plt.subplots()
447

```

```

448     for i in range(animals_plot):
449         evolution = np.array(self.__evolution_history[i])
450         ax.plot(evolution[:, 0], evolution[:, 1], linewidth = 2)
451
452     ax.set_xlabel('X', fontsize = 12)
453     ax.set_ylabel('Y', fontsize = 12)
454
455     ax.set_title("Animals' movement Paths", fontsize = 14)
456     ax.grid()
457
458     if save_figure:
459         fig.savefig(filename + '.png')
460
461     plt.show()
462
463
464
465 def main():
466
467     """ Example of working with Population class """
468
469     population = Population('Full_wet_season.csv') # create population instance, the data is readed from file 'animals.csv'
470
471     population.plot_evolution(save_figure = True, filename = "Animals_migration") # plot migration path of all animals separately
472     population.plot_evolution_XY(animals_plot = 5, save_figure = True, filename = "Animals_5") # plot migration paths on XY plane
473     population.plot_evolution_XY(save_figure = True, filename = "Animals_18")
474     population.plot_evolution()
475
476     # calculate and plot distribution for each year
477     for year in population.years:
478         if population.evaluate_distribution(year):
479             print(year, end = ' ')
480             population.calculate_adv_diff_parameters() # find advection-diffusion parameters
481             population.plot_distributions1D(save_figure = True, filename = 'Distribution_{}'.format(year))
482             population.plot_distribution2D(save_figure = True, filename = 'Distribution2D_{}'.format(year))
483
484
485 if __name__ == "__main__":
486     main()
487

```

Appendix 2: Mutual Relationship between Wildebeests and Zebras during Migration

```

1 import random
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from matplotlib import cm
6 from mpl_toolkits.mplot3d.axes3d import Axes3D
7
8
9
10
11 class Animals:
12
13     """ Class contains all options that are general for all interacting species: Zebra and Wildebeest """
14
15     def __init__(self, N_start, Dx, Dy, Cx, Cy, d, r, K, b = 0, delta_x = 1, delta_y = 1, delta_t = 1, initial_x = 0, initial_y = 0):
16
17         """ Initialize instance with given initial number of animals N_start;
18         diffusion parameters Dx, Dy;
19         advection parameters Cx, Cy;
20         mortality rate d, birth rate r.
21         Calculates corresponding probabilities to make step in different direction p,q,w,z. """
22
23         self.N_walkers = [N_start]
24
25         self.Dx = Dx
26         self.Dy = Dy
27         self.Cx = Cx
28         self.Cy = Cy
29
30         self.d = d
31         self.r = r
32         self.K = K
33         self.b = b
34
35         self.partner = None
36
37         self.delta_xy = [delta_x, delta_y]
38
39         self.delta_t = delta_t
40         self.time = 0
41         self.N_steps = 0
42
43         self.N_points = (1, 1)
44         self.diapason = [[int(initial_x/delta_x), int(initial_y/delta_y)], [int(initial_x/delta_x), int(initial_y/delta_y)]]
45
46         self.p = self.delta_t * (2 * self.Dx / self.delta_xy[0] + self.Cx) / self.delta_xy[0] / 2
47         self.q = self.delta_t * (2 * self.Dx / self.delta_xy[0] - self.Cx) / self.delta_xy[0] / 2
48         self.w = self.delta_t * (2 * self.Dy / self.delta_xy[1] + self.Cy) / self.delta_xy[1] / 2
49         self.z = self.delta_t * (2 * self.Dy / self.delta_xy[1] - self.Cy) / self.delta_xy[1] / 2
50
51         if self.p < 0 or self.q < 0 or self.w < 0 or self.z < 0 or self.p + self.q + self.w + self.z > 1:
52             raise Exception("invalid probabilities p, q, w, z were calculated according to the given advection-diffusion")
53
54         # privat attributes:
55         self.__zero = [initial_x, initial_y]
56         self.__walkers_position = [[int(initial_x/delta_x), int(initial_y/delta_y)] for _ in range(self.N_walkers[0])]
57         self.__walkers_distribution = [[self.N_walkers[0]]]
58
59     def set_interaction(self, partner, main_class = False):
60
61         """ Set partner instance, define if this instance will be main_class,
62         main_class will run evolution for all interacting species. """
63
64         self.partner = partner
65         self.main_class = main_class
66
67     def __one_step(self):
68
69         """ Model one delta_t period for animals population.
70         On the each time step every animal in population can die, add new animal in their current space position,
71         interact with partner specie or make a step in randomly choosen direction. """
72
73         i = 0
74         walkers_next_step = self.N_walkers[-1]
75
76         while i < walkers_next_step:
77
78             # mortality
79             if random.choices([True, False], [self.d * self.delta_t, 1 - self.d * self.delta_t])[0]:
80                 del self.__walkers_position[i]
81                 walkers_next_step -= 1
82
83             elif random.choices([True, False], [self.r * self.N_walkers[-1] * self.delta_t / self.K, 1 - self.r * self.N_walkers[-1] * self.delta_t / self.K])[0]:
84                 del self.__walkers_position[i]
85                 walkers_next_step -= 1
86
87             else:
88                 # birth
89                 if random.choices([True, False], [self.r * self.delta_t, 1 - self.r * self.delta_t])[0]:
90                     self.__walkers_position.append(self.__walkers_position[i])
91                     walkers_next_step += 1

```

```

92         # interaction
93         elif self.b > 0 and self.partner != None and random.choices([True, False], [self.b * self.delta_t * self.par
94             self.__walkers_position.append(self.__walkers_position[i])
95             walkers_next_step += 1
96
97     else:
98         # random walk (diffusion + advection)
99         direction = random.choices(['l', 'r', 'u', 'd', '0'], [self.p, self.q, self.w, self.z, 1 - self.p - self.
100
101         if direction == 'l':
102             self.__walkers_position[i][0] += 1
103         elif direction == 'r':
104             self.__walkers_position[i][0] -= 1
105         elif direction == 'u':
106             self.__walkers_position[i][1] += 1
107         elif direction == 'd':
108             self.__walkers_position[i][1] -= 1
109         i += 1
110
111     self.N_walkers.append(walkers_next_step)
112
113
114 def evolve(self, T):
115     """ Model full animals population evolutions, run one_step() utility T/delta_t times.
116     After that run __evaluate_distribution() utility
117     to calculate the number of walkers in each x, y interval."""
118
119     self.time += T
120     self.N_steps = int(T/self.delta_t)
121
122     if self.main_class:
123         self.partner.time += T
124         self.partner.N_steps = int(T/self.delta_t)
125
126     for n in range(self.N_steps):
127         self.__one_step()
128
129         if self.main_class:
130             self.partner.__one_step()
131
132     self.__evaluate_distribution()
133     if self.main_class:
134         self.partner.__evaluate_distribution()
135
136
137 def __evaluate_distribution(self):
138     """ Utility. Calculate special distribution of animals,
139     number of animals in each delta_x * delta_y interval. """
140
141     self.diapason = [[min(np.array(self.__walkers_position)[: , 0]), min(np.array(self.__walkers_position)[: , 1])], \
142         [max(np.array(self.__walkers_position)[: , 0]), max(np.array(self.__walkers_position)[: , 1])]]
143
144     self.N_points = (self.diapason[1][0] - self.diapason[0][0] + 1, self.diapason[1][1] - self.diapason[0][1] + 1)
145     self.__walkers_distribution = np.zeros(shape = self.N_points)
146
147     for i in range(self.N_walkers[-1]):
148         self.__walkers_distribution[self.__point_index(i)] += 1 / self.N_walkers[-1]
149
150
151 def __point_index(self, i):
152     """ Utility. Return x, y index of point where animal with the i index is placed """
153
154     return self.__walkers_position[i][0] - self.diapason[0][0], self.__walkers_position[i][1] - self.diapason[0][1]
155
156
157 def xy_index(self, x, y):
158     """ Utility. Return x, y index of the point with any given coordinates x, y.
159     If these coordinated are out of diapason occupied by animals, return None """
160
161     if x < self.diapason[0][0] * self.delta_xy[0] or x > self.diapason[1][0] * self.delta_xy[0] or y < self.diapason[0][1]
162         return None
163
164     return int(round(x / self.delta_xy[0] - self.diapason[0][0])), int(round(y / self.delta_xy[1] - self.diapason[0][1])
165
166
167 def __axes(self, self, axis = 2):
168     """Utility. Return array of x or y values in diapason that is occupied by animals
169     with the step of delta_x or delta_y; if axis = 0 it returns x-axis,
170     axis = 1 -- returns y-axis, otherwise return (x,y) coordinates of all points within the diapason"""
171
172     if axis == 0 or axis == 1:
173         return [(self.diapason[0][axis] + i) * self.delta_xy[axis] for i in range(self.N_points[axis])]
174
175     else:
176         x_all = [(self.diapason[0][0] + i) * self.delta_xy[0] for i in range(self.N_points[0]) for j in range(self.N_poi
177         y_all = [(self.diapason[0][1] + j) * self.delta_xy[1] for i in range(self.N_points[0]) for j in range(self.N_poi
178         return x_all, y_all
179

```

```

186
187 def __average_bin(self):
188     """ Utility. Return (x, y) indices of (delta_x, delta_y) 2D interval that contains average animals's position """
189
190     average = self.average_position()
191     axis = self.__axes(0), self.__axes(1)
192
193     average_bin = [0, 0]
194     for j in [0, 1]:
195         for i in range(self.N_points[j]):
196             if average[j] <= axis[j][i]:
197                 break
198             if axis[j][i] - average[j] <= average[j] - axis[j][i-1]:
199                 average_bin[j] = i
200             else:
201                 average_bin[j] = i-1
202
203     return average_bin
204
205 def get_distribution_2D(self):
206
207     """ Returns distribution of animals on XY plane as a function of x, y.
208     This function distribution_2D() works for any x, y (within and out the animals diapason) """
209
210     def disitrbution_2D(x, y):
211         indices = self.xy_index(x, y)
212         if indices == None:
213             return 0
214         return self.__walkers_distribution[indices[0]][indices[1]]
215
216     return disitrbution_2D
217
218 def get_walkers_position(self):
219
220     """ Return physical (i.e. taking into account delta_x, delta_y values) x, y positions of each animal 2D array """
221
222     walkers_physical_position = [[elem[0] * self.delta_xy[0], elem * self.delta_xy[1]] for elem in self.__walkers_position]
223     return walkers_physical_position
224
225 def average_position(self):
226
227     """ Retuns average position of the animals on x and y axis """
228
229     sum = [0, 0]
230     for i in range(self.N_walkers[-1]):
231         for j in [0, 1]:
232             sum[j] += self.__walkers_position[i][j]
233
234     for j in [0, 1]:
235         sum[j] /= self.N_walkers[-1]
236     return sum
237
238 def advection_diffusion(self, x, y, u_x = None, u_y = None, D_x = None, D_y = None):
239
240     """ Calculates advection-diffusion function value for given x and y """
241
242     if not (D_x and D_y and u_x and u_y):
243         D_x = self.Dx
244         D_y = self.Dy
245         u_x = self.Cx
246         u_y = self.Cy
247
248     if self.time <= 0 or np.min(D_x) <= 0 or np.min(D_y) <= 0:
249         raise ValueError
250
251     return np.exp(-(x - u_x * self.time - self.__zero[0])**2 / (4 * D_x * self.time) - (y - u_y * self.time - self.__zero[1])**2 / (4 * D_y * self.time))
252
253 def plot_distribution2D(self, show_average = True, save_figure = True, filename = "distribution2D", set_parameters = False):
254
255     """ Plot distribution of the animals on XY plane as 3D histogram
256     and the solution of advection-diffusion equation as a surface """
257
258     if self.time == 0:
259         print("WARNING: no step is made to print evolution of distribution. Please evolve Population first.")
260         return 0
261
262     def add_average(ax):
263         x_axis = np.linspace(min(x) - 3*self.delta_xy[0], max(x) + 3*self.delta_xy[0], 10**2)
264         y_axis = np.linspace(min(y) - 3*self.delta_xy[1], max(y) + 3*self.delta_xy[1], 10**2)
265
266         x_av = [self.average_position()[0] for _ in y_axis]
267         y_av = [self.average_position()[1] for _ in x_axis]
268
269         ax.plot(x_axis, y_av, zs=0, zdir='z', linewidth = 3, color = 'crimson')
270         ax.plot(x_av, y_axis, zs=0, zdir='z', linewidth = 3, color = 'crimson')
271
272     def fig_normalize(ax):
273

```

```

278     def fig_normalize(ax):
279         ax.set_xlabel("x")
280         ax.set_ylabel("y")
281         ax.view_init(elev=10.)
282         plt.locator_params(nbins = 5)
283
284     fig = plt.figure(figsize = (11, 5))
285     fig.suptitle('Distribution of Animals on the XY plane at T = {}'.format(self.time), fontsize = 16)
286
287     ax = fig.add_subplot(1, 2.5, 1, projection='3d')
288     ax.set_title('Empirical data')
289     distribution = self.get_distribution_2D()
290     x, y = self._axes(2)
291     z = [distribution(x[i], y[i]) for i in range(len(x))]
292
293     while True:
294         for i in range(len(z)):
295             if i >= len(z):
296                 break
297             if z[i] == 0:
298                 del z[i], x[i], y[i]
299             else:
300                 break
301
302     # choose different color theme for main_class and its partners
303     if self.main_class:
304         cmap = cm.get_cmap('viridis')
305         color = cm.viridis
306     else:
307         cmap = cm.get_cmap('plasma')
308         color = cm.plasma
309
310     max_height = max(z)
311     colormap = [cmap(k / max_height) for k in z]
312
313     z_bottom = np.zeros_like(z)
314     ax.bar3d(x, y, z_bottom, self.delta_xy[0], self.delta_xy[1], z, color = colormap, antialiased = True, shade = False)
315
316     if show_average:
317         add_average(ax)
318
319     fig_normalize(ax)
320
321     ax = fig.add_subplot(1, 2.5, 2, projection='3d')
322     ax.set_title('Advection diffusion equation')
323
324     x_std = np.linspace(min(x), max(x), 10**2)
325     y_std = np.linspace(min(y), max(y), 10**2)
326     x_std, y_std = np.meshgrid(x_std, y_std)
327
328     norm = self.delta_xy[0] * self.delta_xy[1]
329     try:
330         if not set_parameters:
331             z_std = norm * self.advection_diffusion(x_std, y_std)
332         else:
333             z_std = norm * self.advection_diffusion(x_std, y_std, mu_x, mu_y, D_x, D_y)
334     except:
335         print("WARNING: invalid values encountered in {}".format(advection_diffusion()))
336         z_std = np.zeros_like(x_std)
337
338     ax.plot_surface(x_std, y_std, z_std, rstride=1, cstride=1, linewidth=0, cmap=color, antialiased = False, alpha = 0.7)
339
340     if show_average:
341         add_average(ax)
342
343     fig_normalize(ax)
344
345     if save_figure:
346         fig.savefig(filename + '.png')
347
348     plt.show()
349
350
351     def plot_distributions1D(self, show_average = True, save_figure = True, filename = "distribution", set_parameters = False):
352         """ Plot the walker's distribution 2D histograms along x and y axis with the other coordinate
353         equal to its average value. """
354
355         if self.time == 0:
356             print("WARNING: no step is made to print evolution of distribution. Please evolve Population first.")
357             return 0
358
359         # choose different colors for main animals class and its partners
360         if self.main_class:
361             color_dist = 'turquoise'
362             color_av = 'dodgerblue'
363         else:
364             color_dist = 'coral'
365             color_av = 'mediumvioletred'
366
367

```



```

369     x = self.__axes(0)
370     y_av = self.average_position()[1]
371     z1 = [self.get_distribution_2D()(x[i], y_av) for i in range(len(x))]
372
373     y = self.__axes(1)
374     x_av = self.average_position()[0]
375     z2 = [self.get_distribution_2D()(x_av, y[i]) for i in range(len(y))]
376
377     norm = self.delta_xy[0] * self.delta_xy[1]
378
379     x_std = np.linspace(min(x) - 2*self.delta_xy[0], max(x) + 2*self.delta_xy[0], 10**3)
380     y_std = np.linspace(min(y) - 2*self.delta_xy[1], max(y) + 2*self.delta_xy[1], 10**3)
381
382     try:
383         if not set_parameters:
384             z1_std = [norm * self.advection_diffusion(_x, y_av) for _x in x_std]
385             z2_std = [norm * self.advection_diffusion(x_av, _y) for _y in y_std]
386         else:
387             z1_std = [norm * self.advection_diffusion(_x, y_av, mu_x, mu_y, D_x, D_y) for _x in x_std]
388             z2_std = [norm * self.advection_diffusion(x_av, _y, mu_x, mu_y, D_x, D_y) for _y in y_std]
389     except:
390         print("WARNING: invalid values encountered in {}".format(advection_diffusion))
391         z1_std = [0 for _ in x_std]
392         z2_std = [0 for _ in y_std]
393
394     fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (12, 5))
395
396     ax1.bar(x, z1, width = self.delta_xy[0], color = color_dist)
397     ax2.bar(y, z2, width = self.delta_xy[1], color = color_dist)
398
399     ax1.plot(x_std, z1_std, linewidth = 2, label = 'Advection diffusion equation', color = 'crimson')
400     ax2.plot(y_std, z2_std, linewidth = 2, label = 'Advection diffusion equation', color = 'crimson')
401
402     if show_average:
403         av_xbin = x[self.__average_bin()[0]]
404         av_ybin = y[self.__average_bin()[1]]
405         z_av = self.get_distribution_2D()(av_xbin, av_ybin)
406         ax1.bar(av_xbin, z_av, width = self.delta_xy[0], label = 'Average x = {:.2f}'.format(x_av), color = color_av)
407         ax2.bar(av_ybin, z_av, width = self.delta_xy[1], label = 'Average y = {:.2f}'.format(y_av), color = color_av)
408
409     ax1.set_title('Average animals distribution on the y-plane')
410     ax2.set_title('Average animals distribution on the x-plane')
411
412     ax1.set_xlabel('X', fontsize = 12)
413     ax2.set_xlabel('Y', fontsize = 12)
414     ax1.set_ylabel('Fraction of animals', fontsize = 12)
415
416     ax1.legend()
417     ax2.legend()
418
419     if save_figure:
420         fig.savefig(filename + '.png')
421
422     plt.show()
423
424
425     def save_num_animals(self):
426
427         """ Save evolution of number of population in txt file """
428
429         t = [[i] for i in range(self.time + 1)]
430         n1 = []
431         n2 = []
432         for i in range(len(self.N_walkers)):
433             n1.append([self.N_walkers[i]])
434             n2.append([self.partner.N_walkers[i]])
435         table = np.concatenate([t, n1, n2], axis = 1)
436
437         df = pd.DataFrame(table)
438         df.columns = ['Time', 'N Zebra', 'N Wildebeest']
439
440         df.to_csv('Two_pecies_interaction.txt', index = False)
441
442
443     def plot_num_animals(self, save_figure = True, filename = "animals_number", show_patners = True):
444
445         """ Plot evolution of number of animals in the population,
446         for this class instance and its partner animal. """
447
448         fig, ax = plt.subplots()
449
450         t = [i for i in range(self.time + 1)]
451
452         ax.plot(t, self.N_walkers, label = 'Zebra', color = 'turquoise')
453         ax.plot(t, self.partner.N_walkers, label = 'Wildebeest', color = 'mediumvioletred')
454
455         ax.set_xlabel('Time', fontsize = 12)
456         ax.set_ylabel('Number of Animals (x 10^2)')
457
458         ax.set_title('Populations of Interacting Species')
459         ax.legend()
460
461

```



```

461
462     if save_figure:
463         fig.savefig(filename + '.png')
464
465     plt.show()
466
467
468 def main():
469
470     # create instances of animal class zebra nad wildebeest
471     zebra = Animals(N_start = 2000, Dx = 0.01, Dy = 0.01, Cx = 0.005, Cy = 0.005, d = 0, r = 0.01, K = 4000, b = 2 / 10**8)
472     wildebeest = Animals(N_start = 13000, Dx = 0.01, Dy = 0.01, Cx = 0.005, Cy = 0.005, d = 0, r = 0.01, K = 24000, b = 15 /
473
474     # d=0, d=0.03, d=0.1
475     # set interaction between zebra and wildebeest
476     zebra.set_interaction(wildebeest, main_class = True) # zebra will be main interacting class
477     wildebeest.set_interaction(zebra)
478
479     # run evolution for interacting species (zebra is main interacting class so it runs evolution for wildebeest on each time
480     zebra.evolve(T = 100)
481
482     # plot spetial distribution of zebra
483     zebra.plot_distribution2D(filename = 'zebra_distribution2D')
484     zebra.plot_distributions1D(filename = 'zebra_distribution1D')
485
486     # plot spetial distribution of wildebeest
487     wildebeest.plot_distribution2D(filename = 'wildebeest_distribution2D')
488     wildebeest.plot_distributions1D(filename = 'wildebeest_distribution1D')
489
490     # plot how the number of zebra and wildebeest changes with time
491     zebra.plot_num_animals(filename = 'zebra_wildebesst_number')
492     zebra.save_num_animals()
493
494 if __name__ == "__main__":
495     main()

```

Appendix 3: Analysis of Prey-Predator Relations of the three Species

```

1  # Objective three
2  import random
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from matplotlib import cm
6  from mpl_toolkits.mplot3d.axes3d import Axes3D
7
8
9  class Animals:
10
11      """ Class contains all options that are general for all interacting species: Zebra and Wildebeest """
12
13      def __init__(self, N_start, Dx, Dy, Cx, Cy, d = 0, r = 0, K = 1, b = 0, delta_x = 1, delta_y = 1, delta_t = 1, initial_x
14
15          """ Initialize instance with given intial number of animals N_start;
16              diffusion parameters Dx, Dy;
17              advection parameters Cx, Cy;
18              mortality rate d, birth rate r.
19              Calculates corresponding probabilities to make step in different direction p,q,w,z. """
20
21          self.N_walkers = [N_start]
22
23          self.Dx = Dx
24          self.Dy = Dy
25          self.Cx = Cx
26          self.Cy = Cy
27
28          self.d = d
29          self.r = r
30          self.K = K
31          self.b = b
32
33          self.main_class = False
34          self.partner = None
35          self.predator = None
36
37          self.a = 0
38          self.h = 0
39          self.g = (0, 0)
40
41          self.delta_xy = [delta_x, delta_y]
42
43          self.delta_t = delta_t
44          self.time = 0
45          self.N_steps = 0
46
47          self.N_points = (1, 1)
48          self.diapason = [[int(initial_x/delta_x), int(initial_y/delta_y)], [int(initial_x/delta_x), int(initial_y/delta_y)]]
49
50          self.p = self.delta_t * (2 * self.Dx / self.delta_xy[0] + self.Cx) / self.delta_xy[0] / 2
51          self.q = self.delta_t * (2 * self.Dx / self.delta_xy[0] - self.Cx) / self.delta_xy[0] / 2
52          self.w = self.delta_t * (2 * self.Dy / self.delta_xy[1] + self.Cy) / self.delta_xy[1] / 2
53          self.z = self.delta_t * (2 * self.Dy / self.delta_xy[1] - self.Cy) / self.delta_xy[1] / 2
54
55          if self.p < 0 or self.q < 0 or self.w < 0 or self.z < 0 or self.p + self.q + self.w + self.z > 1:
56              raise Exception("invalid probabilities p, q, w, z were calculated according to the given advection-diffusion
57
58          self.zero = [initial_x, initial_y]
59          self.walkers_position = [[int(initial_x/delta_x), int(initial_y/delta_y)] for _ in range(self.N_walkers[0])]
60          self.walkers_distribution = [[self.N_walkers[0]]]
61
62
63      def evaluate_distribution(self):
64
65          """ Model full animals population evolutions, run one_step() utility T/delta_t times.
66              After that run __evaluate_distribution() utility
67              to calculate the number of walkers in each x, y interval.
68              ones_step utility is defined separately for Herbivore and Predator classes inherited for Animal class """
69
70          self.diapason = [[min(np.array(self.walkers_position)[:, 0]), min(np.array(self.walkers_position)[:, 1])], \
71                          [max(np.array(self.walkers_position)[:, 0]), max(np.array(self.walkers_position)[:, 1])]]
72
73          self.N_points = (self.diapason[1][0] - self.diapason[0][0] + 1, self.diapason[1][1] - self.diapason[0][1] + 1)
74          self.walkers_distribution = np.zeros(shape = self.N_points)
75
76          for i in range(self.N_walkers[-1]):
77              self.walkers_distribution[self.point_index(i)] += 1 / self.N_walkers[-1]
78
79
80      def point_index(self, i):
81
82          """ Utility. Return x, y index of point where animal with the i index is placed """
83
84          return self.walkers_position[i][0] - self.diapason[0][0], self.walkers_position[i][1] - self.diapason[0][1]
85
86
87      def xy_index(self, x, y):
88
89          """ Utility. Return x, y index of the point with any given coordinates x, y.
90              If these coordinated are out of diapason occupied by animals, return None """
91

```



```

233     else:
234         cmap = cm.get_cmap('plasma')
235         color = cm.plasma
236
237     max_height = max(z)
238     colormap = [cmap(k / max_height) for k in z]
239
240     z_bottom = np.zeros_like(z)
241     ax.bar3d(x, y, z_bottom, self.delta_xy[0], self.delta_xy[1], z, color = colormap, antialiased = True, shade = False)
242
243     if show_average:
244         add_average(ax)
245
246     fig_normalize(ax)
247
248     ax = fig.add_subplot(1, 2.5, 2, projection='3d')
249     ax.set_title('Advection diffusion equation')
250
251     x_std = np.linspace(min(x), max(x), 10**2)
252     y_std = np.linspace(min(y), max(y), 10**2)
253     x_std, y_std = np.meshgrid(x_std, y_std)
254
255     norm = self.delta_xy[0] * self.delta_xy[1]
256
257     if not set_parameters:
258         z_std = norm * self.advection_diffusion(x_std, y_std)
259     else:
260         z_std = norm * self.advection_diffusion(x_std, y_std, mu_x, mu_y, D_x, D_y)
261
262     ax.plot_surface(x_std, y_std, z_std, rstride=1, cstride=1, linewidth=0, cmap=color, antialiased = False, alpha = 0.7)
263
264     if show_average:
265         add_average(ax)
266
267     fig_normalize(ax)
268
269     if save_figure:
270         fig.savefig(filename + '.png')
271
272     plt.show()
273
274
275 def plot_distributions1D(self, show_average = True, save_figure = True, filename = "distribution", set_parameters = False)
276
277     """ Plot the walker's distribution 2D histograms along x and y axis with the other coordinate
278     equal to its average value. """
279
280
281     """ Plot distribution of the animals on XY plane as 3D histogram
282     and the solution of advection-diffusion equation as a surface """
283
284     if self.time == 0:
285         print("WARNING: no step is made to print evolution of distribution. Please evolve Population first.")
286         return 0
287
288     def add_average(ax):
289         x_axis = np.linspace(min(x) - 3*self.delta_xy[0], max(x) + 3*self.delta_xy[0], 10**2)
290         y_axis = np.linspace(min(y) - 3*self.delta_xy[1], max(y) + 3*self.delta_xy[1], 10**2)
291
292         x_av = [self.average_position()[0] for _ in y_axis]
293         y_av = [self.average_position()[1] for _ in x_axis]
294
295         ax.plot(x_axis, y_av, zs=0, zdir='z', linewidth = 3, color = 'crimson')
296         ax.plot(x_av, y_axis, zs=0, zdir='z', linewidth = 3, color = 'crimson')
297
298     def fig_normalize(ax):
299         ax.set_xlabel("x")
300         ax.set_ylabel("y")
301         ax.view_init(elev=10.)
302         plt.locator_params(nbins = 5)
303
304     fig = plt.figure(figsize = (11, 5))
305     fig.suptitle('Distribution of Animals on the XY plane at T = {}'.format(self.time), fontsize = 16)
306
307     ax = fig.add_subplot(1, 2.5, 1, projection='3d')
308     ax.set_title('Empirical data distribution')
309     distribution = self.get_distribution_2D()
310     x, y = self.axes(2)
311     z = [distribution(x[i], y[i]) for i in range(len(x))]
312
313     while True:
314         for i in range(len(z)):
315             if i >= len(z):
316                 break
317             if z[i] == 0:
318                 del z[i], x[i], y[i]
319             else:
320                 break
321
322     # choose different color theme for main_class, predator and herbivore partners
323     if self.main_class:
324         cmap = cm.get_cmap('viridis')
325         color = cm.viridis
326     elif self.__class__.__name__ == 'Predator':
327         cmap = cm.get_cmap('magma')
328         color = cm.magma

```

```

279
280     if self.time == 0:
281         print("WARNING: no step is made to print evolution of distribution. Please evolve Population first.")
282         return 0
283
284     # choose different colors for main class its herbivore partner and predator
285     if self.main_class:
286         color_dist = 'turquoise'
287         color_av = 'dodgerblue'
288     elif self.__class__.__name__ == 'Predator':
289         color_dist = 'violet'
290         color_av = 'deeppink'
291     else:
292         color_dist = 'coral'
293         color_av = 'mediumvioletred'
294
295
296     x = self.axes(0)
297     y_av = self.average_position()[1]
298     z1 = [self.get_distribution_2D()(x[i], y_av) for i in range(len(x))]
299
300     y = self.axes(1)
301     x_av = self.average_position()[0]
302     z2 = [self.get_distribution_2D()(x_av, y[i]) for i in range(len(y))]
303
304     norm = self.delta_xy[0]* self.delta_xy[1]
305
306     x_std = np.linspace(min(x) - 2*self.delta_xy[0], max(x) + 2*self.delta_xy[0], 10**3)
307     y_std = np.linspace(min(y) - 2*self.delta_xy[1], max(y) + 2*self.delta_xy[1], 10**3)
308
309     try:
310         if not set_parameters:
311             z1_std = [norm * self.advection_diffusion(_x, y_av) for _x in x_std]
312             z2_std = [norm * self.advection_diffusion(x_av, _y) for _y in y_std]
313         else:
314             z1_std = [norm * self.advection_diffusion(_x, y_av, mu_x, mu_y, D_x, D_y) for _x in x_std]
315             z2_std = [norm * self.advection_diffusion(x_av, _y, mu_x, mu_y, D_x, D_y) for _y in y_std]
316     except:
317         print("WARNING: invalid values encountered in {}".format(advection_diffusion))
318         z1_std = [0 for _ in x_std]
319         z2_std = [0 for _ in y_std]
320
321     fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (12, 5))
322
323     ax1.bar(x, z1, width = self.delta_xy[0], color = color_dist)
324     ax2.bar(y, z2, width = self.delta_xy[1], color = color_dist)
325
326     ax1.plot(x_std, z1_std, linewidth = 2, label = 'Theoretical distribution', color = 'crimson')
327     ax2.plot(y_std, z2_std, linewidth = 2, label = 'Theoretical distribution', color = 'crimson')
328
329     if show_average:
330         av_xbin = x[self.average_bin()[0]]
331         av_ybin = y[self.average_bin()[1]]
332         z_av = self.get_distribution_2D()(av_xbin, av_ybin)
333         ax1.bar(av_xbin, z_av, width = self.delta_xy[0], label = 'Average x = {:.2f}'.format(x_av), color = color_av)
334         ax2.bar(av_ybin, z_av, width = self.delta_xy[1], label = 'Average y = {:.2f}'.format(y_av), color = color_av)
335
336     ax1.set_title('Animals distribution in the y-plane ')
337     ax2.set_title('Animals distribution in the x-plane ')
338
339     ax1.set_xlabel('X', fontsize = 12)
340     ax2.set_xlabel('Y', fontsize = 12)
341     ax1.set_ylabel('Fraction of Animals', fontsize = 12)
342
343     ax1.legend()
344     ax2.legend()
345
346     if save_figure:
347         fig.savefig(filename + '.png')
348
349     plt.show()
350
351
352 def plot_num_animals(self, save_figure = True, filename = "animals_number", show_patterns = True):
353
354     """ Plot evolution of number of animals in the population,
355     for this class instance and its partner animals (one herbivore and on one predator). """
356
357     fig, ax = plt.subplots()
358
359     t = [i for i in range(self.time + 1)]
360
361     ax.plot(t, self.N_walkers, label = 'Zebra', color = 'turquoise')
362     ax.plot(t, self.partner.N_walkers, label = 'Wildebeest', color = 'mediumvioletred')
363
364     ax.set_xlabel('Time', fontsize = 12)
365     ax.set_ylabel('Nuber of Animals')
366
367     ax.set_title('Populations of Interacting Species')
368     ax.legend()
369
370     if save_figure:
371         fig.savefig(filename + '.png')
372

```

```

372     plt.show()
373
374
375
376
377 class Herbivore(Animals):
378     """ Class contains all functionality for Zebra and Wildebeest species interacting one with another
379     and with one predator instance -- Lions """
380
381
382
383     def set_interaction(self, partner, predator, a, h, main_class = False):
384         """ Set partner and predator instances and parameters of interaction with predator a, h. """
385
386         self.partner = partner
387         self.main_class = main_class
388
389         self.predator = predator
390         self.a = a
391         self.h = h
392
393
394
395     def evolve(self, T):
396         """ Model full animals population evolutions, run one_step() utility T/delta_t times.
397         After that run __evaluate_distribution() utility
398         to calculate the number of walkers in each x, y interval.
399         If it is main_class also run evolution for partner herbivore and predator. """
400
401         self.time += T
402         self.N_steps = int(T/self.delta_t)
403
404         if self.main_class:
405             self.partner.time += T
406             self.partner.N_steps = int(T/self.delta_t)
407             self.predator.time += T
408             self.predator.N_steps = int(T/self.delta_t)
409
410         for n in range(self.N_steps):
411             self.one_step()
412
413             if self.main_class:
414                 self.partner.one_step()
415                 self.predator.one_step()
416
417         self.evaluate_distribution()
418
419         if self.main_class:
420             self.partner.evaluate_distribution()
421             self.predator.evaluate_distribution()
422
423
424     def one_step(self):
425         """ Model one delta_t period for herbivore population.
426         On the each time step every animal in population can die, add new animal in their current space position,
427         interact with herbivore partner, be eaten by predator or make a step in randomly chosen direction. """
428
429         i = 0
430         walkers_next_step = self.N_walkers[-1]
431
432         while i < walkers_next_step:
433
434             # mortality
435             if random.choices([True, False], [self.d * self.delta_t, 1 - self.d * self.delta_t])[0]:
436                 del self.walkers_position[i]
437                 walkers_next_step -= 1
438
439             elif random.choices([True, False], [self.r * self.N_walkers[-1] * self.delta_t / self.K, 1 - self.r * self.N_wa
440                 del self.walkers_position[i]
441                 walkers_next_step -= 1
442
443             # interaction with predator
444             elif random.choices([True, False], [self.h * self.delta_t * self.predator.N_walkers[-1] / (1 + self.a * self.N_wa
445                 del self.walkers_position[i]
446                 walkers_next_step -= 1
447
448             else:
449                 # birth
450                 if random.choices([True, False], [self.r * self.delta_t, 1 - self.r * self.delta_t])[0]:
451                     self.walkers_position.append(self.walkers_position[i])
452                     walkers_next_step += 1
453
454                 # interaction
455                 elif self.b > 0 and self.partner != None and random.choices([True, False], [self.b * self.delta_t * self.part
456                     self.walkers_position.append(self.walkers_position[i])
457                     walkers_next_step += 1
458
459                 else:
460                     # random walk (diffusion + advection)
461                     direction = random.choices(['l', 'r', 'u', 'd', '0'], [self.p, self.q, self.w, self.z, 1 - self.p - self.
462                     if direction == 'l':
463                         self.walkers_position[i][0] -= 1
464
465

```

```

464         if direction == 'l':
465             self.walkers_position[i][0] += 1
466         elif direction == 'r':
467             self.walkers_position[i][0] -= 1
468         elif direction == 'u':
469             self.walkers_position[i][1] += 1
470         elif direction == 'd':
471             self.walkers_position[i][1] -= 1
472         i += 1
473
474     self.N_walkers.append(walkers_next_step)
475
476
477     def plot_num_animals(self, save_figure = True, filename = "animals_number", show_patners = True):
478
479         """ Plot evolution of number each species animals together on the same plot. """
480
481         fig, ax = plt.subplots()
482
483         t = [i for i in range(self.time + 1)]
484
485         ax.plot(t, self.N_walkers, label = 'Zebra', color = 'turquoise')
486         ax.plot(t, self.partner.N_walkers, label = 'Wildebeest', color = 'mediumvioletred')
487         ax.plot(t, self.predator.N_walkers, label = 'Lion', color = 'firebrick')
488
489         ax.set_xlabel('Time', fontsize = 12)
490         ax.set_ylabel('Number of Animals (x10^2)')
491
492         ax.set_title('Populations of Interacting Species')
493         ax.legend()
494
495         if save_figure:
496             fig.savefig(filename + '.png')
497
498         plt.show()
499
500     def save_num_animals(self):
501
502         """ Save evolution of number of population in txt file """
503
504         t = [[i] for i in range(self.time + 1)]
505         n1 = []
506         n2 = []
507         n3 = []
508
509         for i in range(len(self.N_walkers)):
510             n1.append([self.N_walkers[i]])
511             n2.append([self.partner.N_walkers[i]])
512             n3.append([self.predator.N_walkers[i]])
513         table = np.concatenate([t, n1, n2, n3], axis = 1)
514
515         df = pd.DataFrame(table)
516         df.columns = ['Time', 'N Zebra', 'N Wildebeest', 'N Lions']
517
518         df.to_csv('Three_pecies_interaction.txt', index = False)
519
520
521     class Predator(Animals):
522
523         """ Class models predator animal, its interaction with two herbivore partners,
524         mortality and random walk. """
525
526         def set_interaction(self, partner1, partner2, g1, g2):
527
528             """ Set two herbivore partners and parameters of interaction g1, g2. """
529
530             self.partners = (partner1, partner2)
531             self.g = (g1, g2)
532
533
534         def one_step(self):
535
536             """ Model one delta_t period for predator population.
537             On the each time step every predator animal in population can die, add new animal through interaction with herbivore
538             or make a step in randomly choosen direction. """
539
540             j = 0
541             walkers_next_step = self.N_walkers[-1]
542
543             while j < walkers_next_step:
544
545                 # mortality
546                 if random.choices([True, False], [self.d * self.delta_t, 1 - self.d * self.delta_t])[0]:
547                     del self.walkers_position[j]
548                     walkers_next_step -= 1
549
550                 else:
551                     # interaction with herbivore
552                     for i in range(len(self.partners)):
553                         partner = self.partners[i]
554                         probability = self.g[i] * self.delta_t * partner.h * partner.N_walkers[-1] / (1 + partner.a * partner.N_w
555

```



```

557         if random.choices([True, False], [probability, 1 - probability])[0]:
558             self.walkers_position.append(self.walkers_position[i])
559             walkers_next_step += 1
560             j += 1
561             break
562         else:
563             # random walk (diffusion + advection)
564             direction = random.choices(['l', 'r', 'u', 'd', '0'], [self.p, self.q, self.w, self.z, 1 - self.p - self.q - self.w - self.z])[0]
565
566             if direction == 'l':
567                 self.walkers_position[j][0] += 1
568             elif direction == 'r':
569                 self.walkers_position[j][0] -= 1
570             elif direction == 'u':
571                 self.walkers_position[j][1] += 1
572             elif direction == 'd':
573                 self.walkers_position[j][1] -= 1
574
575             j += 1
576
577         self.N_walkers.append(walkers_next_step)
578
579
580
581 def main():
582
583     # creates zebra and wildebeest instances of Herbivore class and lion instance of Predator class
584     zebra = Herbivore(N_start = 2000, Dx = 0.01, Dy = 0.01, Cx = 0.005, Cy = 0.005, d = 0.02, r = 0.04, K = 3000, b = 2 / 10)
585     wildebeest = Herbivore(N_start = 13000, Dx = 0.01, Dy = 0.01, Cx = 0.005, Cy = 0.005, d = 0.02, r = 0.04, K = 18000, b = 2 / 10)
586     lion = Predator(N_start = 30, Dx = 0.01, Dy = 0.01, Cx = 0.005, Cy = 0.005, d = 0.01)
587     #d=0.02
588
589     # set interaction between zebra and wildebeest, and between zebra(wildebeest) and lion
590     zebra.set_interaction(partner = wildebeest, predator = lion, a = 1, h = 0.01, main_class = True)
591     wildebeest.set_interaction(partner = zebra, predator = lion, a = 1, h = 0.01)
592     lion.set_interaction(zebra, wildebeest, 0.25, 0.25)
593     # zebra will be main class
594
595
596     zebra.evolve(100) # run evolution for zebra and its partners wildebeests and lion
597
598     # plot special distributions of zebras, wildebeests and lions after all evolution steps.
599     zebra.plot_distribution2D(filename = 'zebra_model2_2D')
600     zebra.plot_distributions1D(filename = 'zebra_model2_1D')
601
602
603     wildebeest.plot_distribution2D(filename = 'wildebeest_model2_2D')
604     wildebeest.plot_distributions1D(filename = 'wildebeest_model2_1D')
605
606     lion.plot_distribution2D(filename = 'lion_model2_2D')
607     lion.plot_distributions1D(filename = 'lion_model2_1D')
608
609     zebra.plot_num_animals(filename = 'zebra_wildebeest_lions_number')
610     zebra.save_num_animals()
611
612 if __name__ == "__main__":
613     main()

```


RESEARCH OUTPUTS

(i) Publications

Kisoma, L. N., & Kuznetsov, D. (2020). Modelling Predator Prey with Diffusion for Migrating Wildebeest and Lion in the Presence of Drought. *Journal of Mathematics and Informatics*, 18, 2349-0632. Doi: 10.22457/jmi. v18a8164.

Kisoma, L. N., Kuznetsov, D., Torney, C., Treydte, C. A. (2020). An investigation of power law distribution in wildebeest (*Connochaetes taurinus*) herds in Serengeti National Park, Tanzania. *Commun. Math. Biol. Neurosci.* 2020:66 doi.org/10.28919/cmbn/4943.

(ii) Poster Presentation