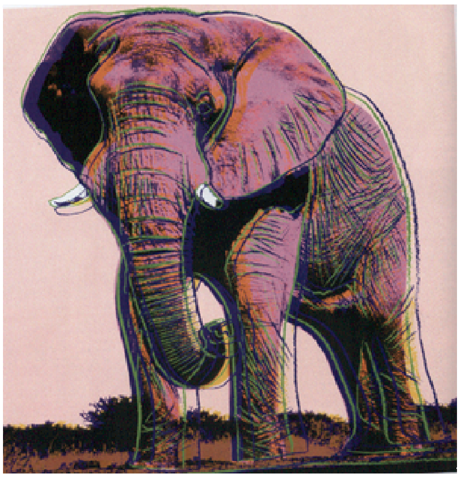
Amarula Elephant Research Programme
School of Life and Environmental Sciences
University of Natal
DURBAN 4001

Modelling Population Dynamics of Elephants



by Samy Gallego

June 2003

Acknowledgements

I would like to thank sincerely the following people who helped me during my internship:

Mr Bruce Page for proposing me this project, accepting to be my supervisor and helping me to realize it.

Prof Rob Slotow for giving me some bursaries and lending me one of his computer in his wonderfully air-conditioned laboratory.

Mr Hugh Murell for teaching me Mathematica.

Mr Philippe Bellan for helping me to go in the University of Natal and accepting to be my supervisor.

Mr Bradley Carter from Space for Elephants, for showing me how tracking elephants in Pongola game reserve.

Miss Marina Alais for testing an early version of my program.

Miss Thenjiwe Ndlovu for giving me her good mood everyday.

Miss Vaneshree Govender for answering my stupid questions about english language.

Miss Sandra Spagnol for proof reading my report and helping me to wake up every morning.

All the elephants of South Africa that I encountered for not charging me and my little blue Volkswagen golf.

Contents

List of Figures

List of Tables

Introduction

As an internship for my french engineering diploma of Applied Mathematics and modelling, I had a five month project with the Amarula Elephant Research Programme concerning modelling of elephant population dynamics.

In South Africa (Southern Africa in general), elephants are a key component of game reserves wishing to attract tourists. The problem is that they are very difficult to manage because populations are growing exponentially in most game reserves. The habitat capacity is bounded so elephants have to be removed every year, otherwise they destroy the vegetation (see the current passionate debate involving elephants in Botswana in Appendix E "To cull or not to cull").

During the internship, I built a software called "Simulele" using Matlab and Fortran. This software permits one to simulate the growth of a population knowing the initial state and it can calculate the number of calves (under the age of 1) that have to be removed so that the population is stationary. I also tried different models under Mathematica to simulate elephant-tree ecosystems in a two dimensional lattice.

This document is a report of this project and has several purposes. It is divided into two parts describing the two main phases of the work done during the internship.

The first part is dedicated to population dynamics and to the creation of the software "simulele". First, it gives some general information about African Elephants and their situation in South Africa. Then this document explains the different models used in the software and how they are programmed. It can interest the reader who wants to understand what is behind the graphical interface so that he can modify the programs and add other functionalities. Finally, the first part include an application on the St Lucia elephant population in order to show how the software works.

In the second part, the report explores elephant-tree ecosystems. It starts from the well-known Caughley's model based on a system of two differential equations and gives some leads to add space variables to it. Finally, it describes a cellular automaton model which is probably the most promising to generate population oscillations observed by Caughley.

Part I

Population dynamics

Chapter 1

Some facts about the african elephants

In this chapter, the reader can find some very general information about elephants, but he can also find some clues on which parameters to use in the program simulele.

1.1 General description

The elephant is the largest living land mammal, found in tropical regions of Africa and Asia. Elephants have massive bodies and heads, thick, pillarlike legs, and broad, short padded feet, with toes bearing heavy, hooflike nails. The gray skin is loose, tough, thick, and nearly hairless. The slender tail ends in a tuft of hair. The upper lip and nose are elongated into a flexible trunk, or proboscis, reaching nearly to the ground; this sensitive appendage is used for picking up food, feeding from trees and other sources, and drawing up water. The large, thin, floppy ears provide an extensive cooling surface; the animal flaps its ears vigorously when it is overheated. The upper incisor teeth are elongated into tusks highly valued for their ivory which the animal uses for digging up roots and tubers. A gland between the eye and the ear periodically produces an oily substance called musth; during these periods the animal is in an excitable, dangerous condition, also called musth, meaning madness. Such a condition occurs more often in males than in females and is thought to be a state of sexual excitement.

Elephants are browsing animals, feeding on fruits, leaves, shoots, and tall grasses; they consume hundreds of pounds of food a day and drink up to 50 gal (190 liters) of water. They travel in groups called family units of between 15 and 30 individuals, composed of a matriarch (female of 35-65 years old) and her daughters and their offspring. Up to five family units associate in bond groups temporarily. Males[1] older than 13 years are generally solitary or

[1] Males are usually called bulls, and females are usually called cows.

live in small groups. A single calf is born after a gestation of 22 months and is nursed approximately for 5 years. Elephants reach maturity at between 11 and 20 years of age; their lifespan is usually 60 or 70 years. Elephants walk at a pace of about 4 mi (6.4 km) per hr, but can charge at speeds of 30 mi (48 km) per hr. They can not jump (it is said that elephants are the only terrestrial mammals which can not jump) and so cannot pass barriers too wide or too high to step over.

The African elephant, Loxodonta africana, is found in Africa at the south of the Sahara. African bull elephants may reach a shoulder height of 13 ft (4 m) and weigh 6 to 8 tons. Their tusks are more than 10 ft (3 m) long and weigh up to 200 lbs (90 kg) each. Females are somewhat smaller and have more slender tusks. African elephants have enormous ears, measuring up to 42 in. (107 cm) in diameter. The long, conspicuously wrinkled trunk terminates in two fleshy, fingerlike protuberances, used for handling objects.

1.2 Measuring elephant ages

1.2.1 Shoulder-height equations

Once alive, it is quite difficult to measure the age of an elephant. Von Bertalanffy (1938) deduced a growth equation for vertebrates which gives the relationship between the shoulder-height (h) and the age (x). The parameters of the Von Bertalanffy equation were calculated by Laws, Parker and Johnstone (1975) for both sexes. For the male a second equation was fitted for ages above twenty years. These equations are:

$$\begin{aligned}
\text{male (1-20 years): } h(x) &= 265(1 - e^{-0.114(x+3.95)})\ cm && (1.1)\\
\text{male (20-60 years): } h(x) &= 307(1 - e^{-0.166(x-10.48)})\ cm && (1.2)\\
\text{female (2-60 years): } h(x) &= 252(1 - e^{-0.099(x+6.00)})\ cm && (1.3)
\end{aligned}$$

If we want to deduce the age from the height, we should use these equations:

$$\begin{aligned}
\text{male (96-245 cm): } x(h) &= -\frac{log(1-\frac{h}{265})}{0.114} - 3.95\ years && (1.4)\\
\text{male (245-307 cm): } x(h) &= -\frac{log(1-\frac{h}{307})}{0.166} + 10.48\ years && (1.5)\\
\text{female (113-252 cm): } x(h) &= -\frac{log(1-\frac{h}{252})}{0.099} - 6.00\ years && (1.6)
\end{aligned}$$

1.2.2 Other methods

Once dead or asleep, the accepted criterion for ageing African elephant is the degree of molar replacement and wear. However, the methods of establishing

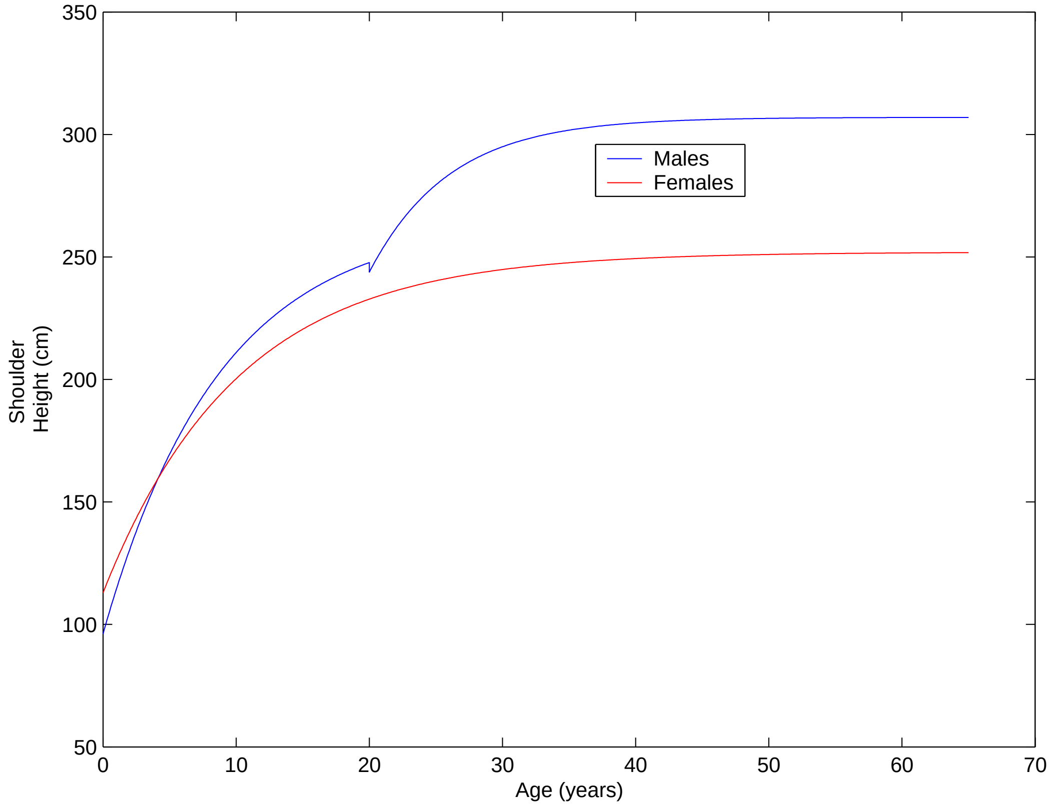


Figure 1.1: Growth curve for male and female elephants (after Laws, Parker and Johnstone (1975))

absolute ages, and the ages allocated to particular molar situations vary. Three methods have been developed:

- Molar usage-interval determinations of Johnson & Buss (1965), and Krumrey & Buss (1968)
- The "FM" (foramen mentale) technique and Age Estimation Reference Chart of Sikes (1966,1968)
- The pictorial age scale of Laws (1966)

1.3 Mortality

Man is the only predator that has a limiting effect on elephant populations. Other predators include lions, wild dogs, hyenas and crocodiles, which usually prey on isolated calves or less than two years.

Mortality[2] depends strongly on age so that we can speak of age specific mortality. If we plot the mortality against the age, a J-shape is usually obtained.

Table 1.1 and Figure 1.2 show the mortality rates for the elephants of the Tsavo National Park (Kenya) before 1970-71 (after Laws (1969b)) and during the drought period of 1970-71 (after Corfield (1973)).

Age	mortality before 1970-71		mortality during 1970-71	
	Female	Male	Female	Male
0-1	36.0	36.0	18.2	18.2
2-5	10.2	10.2	37.6	37.6
6-10	2.4	2.4	17.1	17.1
11-15	2.4	2.4	7.8	7.7
16-20	2.1	2.1	6.5	1.9
21-25	2.0	1.6	12.1	3.3
26-30	2.5	7.0	29.3	3.7
31-35	2.5	7.5	28.4	6.5
36-40	2.6	6.9	61.3	12.1
41-45	3.0	8.8	33.4	6.1
46-50	8.1	27.4	66.3	26.7
51-55	18.4	19.0	83.7	31.3
56-60	29.8	100.0	63.2	100.0

Table 1.1: Mortality rates for the Tsavo National Park (Kenya), expressed as a percentage per year per annual age group (after [17] Laws (1969b) and [3] Corfield (1973)

[2] Survivorship is equal to one minus the mortality rate.

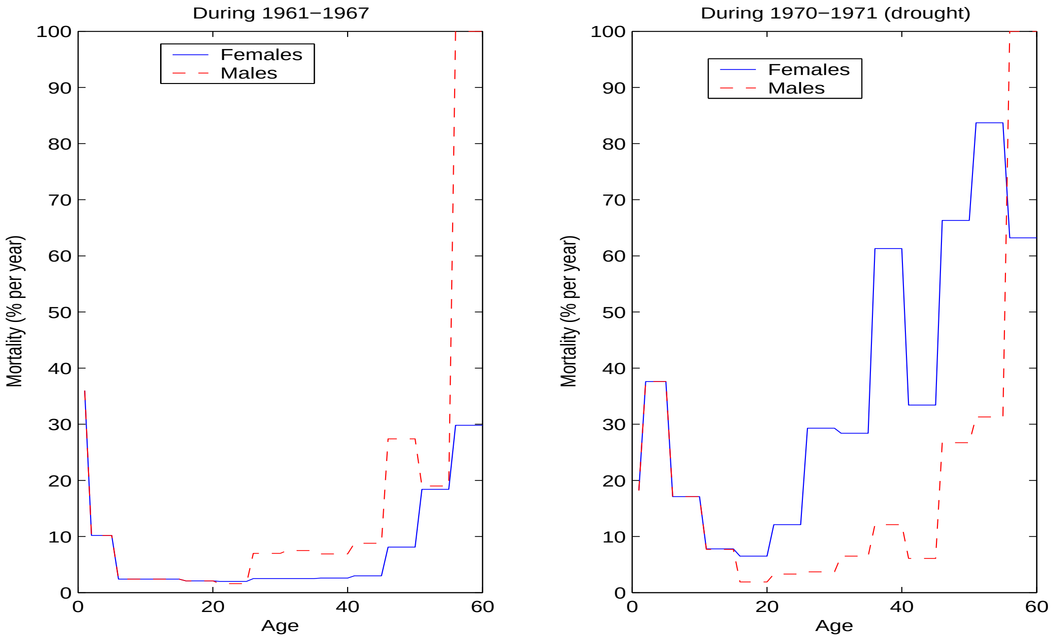


Figure 1.2: Mortality rates for the Tsavo National Park (Kenya), expressed as a percentage per year per annual age group (after [17] Laws (1969b) and [3] Corfield (1973))

We clearly see that the mortality is higher during the drought. As agreed by a lot of elephant experts (Laws, Parker and Johstone [18], Jachmann [10]), the survival of certain age groups is sensitive to environmental conditions among which rain is of paramount importance. Dry year mortality of the youngest and oldest age groups may be greater than in wet year, especially if the number of elephants is close to the carrying capacity of the park.

What is more difficult to interpret is the difference between the sex-specific mortality and all our models, we will assume the mortality to be the same for both sexes.

J. Hanks and J.E.A. McIntoch [9] built an elephant population model in 1973 (they did not give details of the model) in which they considered three hypothetical mortality patterns, a low mortality population (L_{mort}), a medium mortality population (M_{mort}), and one with a high mortality (H_{mort}) as shown in in Table 1.3.

Age	L_{mort}	M_{mort}	H_{mort}
1-4 years	5	10	20
5-45 years	1	1.5	4
46-55 years	4	5	50
56-60 years	50	50	100

Table 1.2: Three hypothetical mortality patterns used in the models of J. Hanks and J.E.A McIntosh [9]

1.4 Fecundity

1.4.1 Age of sexual maturity

The age of first ovulation ranges from population to population, from as young as seven to as old as twenty two years.

Table 1.4 shows Ages of sexual maturity for females of 7 different populations of elephant.

In the model of Hanks and McIntosh [9], they considered three different ages of sexual maturity : 12 years in good conditions, 17 years in medium conditions and 22 years in poor conditions.

1.4.2 Menopause

It is very difficult to find data about ages of menopause for the female elephants. It is commonly believed that menopause occurs around 50 years.

In the model of Hanks and McIntosh [9], they considered three different ages of menopause : 55 years in good conditions, 50 years in medium conditions and 45 years in poor conditions.

Sample	Mean Age of sexual maturity	Confidence limits
Mkomazi	12.24	11.33-13.15
Mkomazi East	12.18	10.92-13.44
Tsavo National Park	11.73	10.8-12.66
Murchison Falls National Park N.	16.28	15.48-17.08
Murchison Falls National Park S.	17.82	16.96-18.68
Budongo Forest Reserve	22.38	19.86-24.90
Luangwa Valley National Park	14.0	-

Table 1.3: Age of sexual maturity in seven different elephant populations (after [10])

1.4.3 Calving Interval

Most often in the literature, the fecundity is expressed in term of calving interval[3]. The calving interval (CI) is the time interval between the birth of two calves for the same female. It is the sum of the duration of anoestrous (DA) and the duration of gestation (DG).

$$CI = DA + DG \tag{1.7}$$

The duration of anoestrous can be estimated by the formula (after Hanks 1972):

$$\begin{aligned} DA &= \frac{DG \times \text{Number not pregnant}}{\text{Number pregnant}} \qquad (1.8) \\ &= \frac{DG \times (1 - \%\text{pregnant})}{\%\text{pregnant}} \qquad (1.9) \end{aligned}$$

So the mean calving interval for a population can be estimated by the formula:

$$CI = \frac{DG}{\%\text{pregnant}} \tag{1.10}$$

Where the duration of gestation time is 22 months after Kenneth and Ritchie (1953) and Lang (1967).

Most estimates of calving intervals have been calculated from the placental scars of culled elephants. Laws and Parker (1968) that twinning occurred in less than 1% of conceptions, which is negligible. Table 1.4 shows calving intervals for 7 different populations of elephant, and we can see that the boundaries are 2.6 years and 13.5 years.

To finish, fecundity can be considered to be age independent between the age of sexual maturity and the age of menopause, as observed by Laws, Parker and Johnstone (1975).

[3] In the Section 3.3.2 page 48, I explain how you can calculate the fecundity from the calving interval.

Sample	Mean Calving Interval	Confidence limits
Mkomazi	2.9	2.6-3.4
Mkomazi East	4.2	3.1-5.0
Tsavo National Park	6.8	5.1-10.3
Murchison Falls National Park N.	9.1	7.5-11.5
Murchison Falls National Park S.	5.6	4.8-6.8
Budongo Forest Reserve	7.7	5.4-13.5
Luangwa Valley National Park	4.0	-
Kasungu National Park	3.9	2.2-5.3

Table 1.4: Calving interval in 7 different elephant populations (after [10])

1.4.4 Parameters which influence age of sexual maturity, age of menopause and calving interval

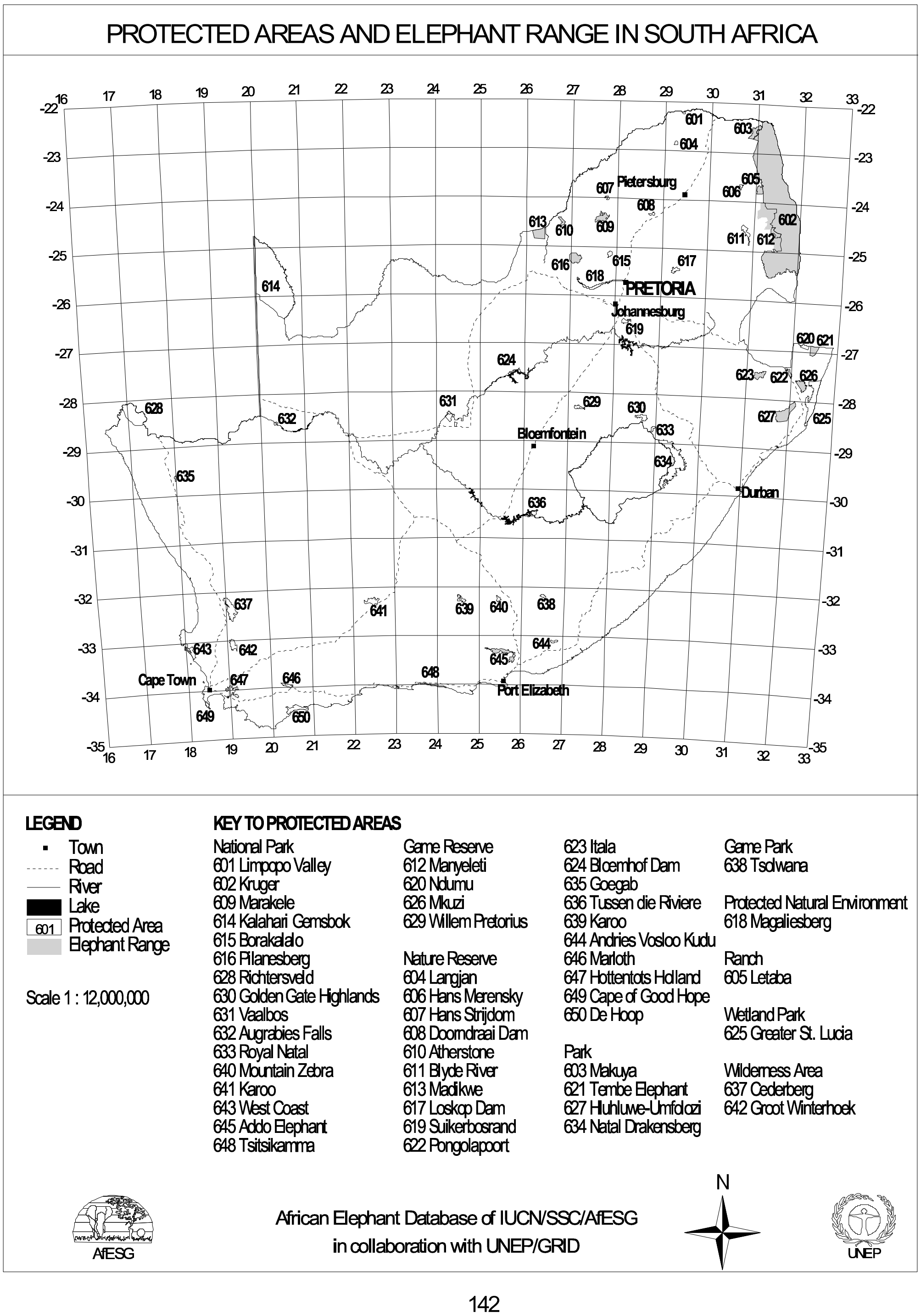
Once again, both Laws (1969) and Hanks (1979) report an increase in the calving interval and the age of sexual maturity with a decrease in environmental conditions among which rain is of paramount importance. Concerning the age of menopause, it decreases with the environmental conditions.

1.5 Elephant range in South Africa

The IUCN Species Survival Commission provides the most complete source of continental information on the numbers and distribution of the African elephant. Unfortunately the data for 2002 are not yet available but the data for 1998 give a good idea of the elephant range. All the data are summarized in [1] *African Elephant Database 1998*. Because most of South Africa's elephants live within protected areas, the populations are relatively well known and the range estimates are accurate.

The largest portion of elephant range falls within and around Kruger National Park. The area includes some range outside the park, adjacent to its southwestern borders. The remaining elephant populations are distributed in small parks and private reserves scattered across the country.

South Africa's elephant population recovered from a low of 120 animals in 1920 (Hall-Martin, 1992) to more than 12000 today. About 75% of South Africa's elephants are foun in the Kruger National Park, where an aerial total count is carried out annually from helicopters (Whyte and Wood, 1996,1997). The other two major elephant habitats are Addo Elephant National Park and Hluhluwe-Umfolozi Park (see the population estimate and the map in Figure 1.4). There are 20 protected areas and many private reserves (Figure 1.3) with elephant populations for which updated estimates have been provided in 1997 or 1998. South Africa is unique because all the estimates are obtained from either total counts (from the ground or air) or from individual registration.



PROTECTED AREAS AND ELEPHANT RANGE IN SOUTH AFRICA

LEGEND

- ▪ Town
- ----- Road
- —— River
- ■ Lake
- 601 Protected Area
- ▨ Elephant Range

Scale 1 : 12,000,000

KEY TO PROTECTED AREAS

National Park
601 Limpopo Valley
602 Kruger
609 Marakele
614 Kalahari Gemsbok
615 Borakalalo
616 Pilanesberg
628 Richtersveld
630 Golden Gate Highlands
631 Vaalbos
632 Augrabies Falls
633 Royal Natal
640 Mountain Zebra
641 Karoo
643 West Coast
645 Addo Elephant
648 Tsitsikamma

Game Reserve
612 Manyeleti
620 Ndumu
626 Mkuzi
629 Willem Pretorius

Nature Reserve
604 Langjan
606 Hans Merensky
607 Hans Strijdom
608 Doorndraai Dam
610 Atherstone
611 Blyde River
613 Madikwe
617 Loskop Dam
619 Suikerbosrand
622 Pongolapoort
623 Itala
624 Bloemhof Dam
635 Goegab
636 Tussen die Riviere
639 Karoo
644 Andries Vosloo Kudu
646 Marloth
647 Hottentots Holland
649 Cape of Good Hope
650 De Hoop

Park
603 Makuya
621 Tembe Elephant
627 Hluhluwe-Umfolozi
634 Natal Drakensberg

Game Park
638 Tsolwana

Protected Natural Environment
618 Magaliesberg

Ranch
605 Letaba

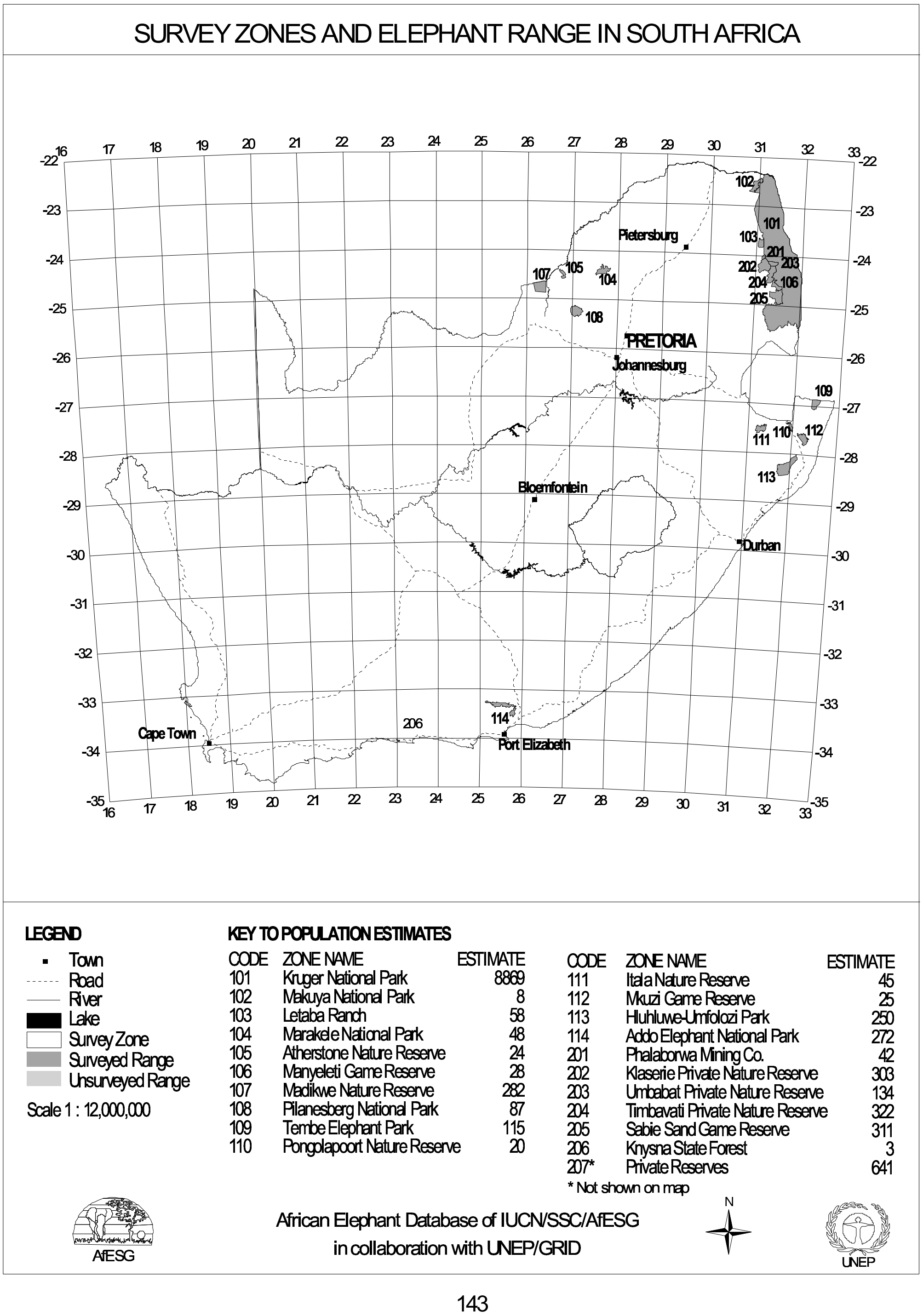
Wetland Park
625 Greater St. Lucia

Wilderness Area
637 Cederberg
642 Groot Winterhoek

African Elephant Database of IUCN/SSC/AfESG
in collaboration with UNEP/GRID

142

Figure 1.3: Protected areas in South Africa (found in [1] *African Elephant Database 1998*, page 142)



SURVEY ZONES AND ELEPHANT RANGE IN SOUTH AFRICA

LEGEND

■	Town
------	Road
——	River
■	Lake
□	Survey Zone
▨	Surveyed Range
░	Unsurveyed Range

Scale 1 : 12,000,000

KEY TO POPULATION ESTIMATES

CODE	ZONE NAME	ESTIMATE	CODE	ZONE NAME	ESTIMATE
101	Kruger National Park	8869	111	Itala Nature Reserve	45
102	Makuya National Park	8	112	Mkuzi Game Reserve	25
103	Letaba Ranch	58	113	Hluhluwe-Umfolozi Park	250
104	Marakele National Park	48	114	Addo Elephant National Park	272
105	Atherstone Nature Reserve	24	201	Phalaborwa Mining Co.	42
106	Manyeleti Game Reserve	28	202	Klaserie Private Nature Reserve	303
107	Madikwe Nature Reserve	282	203	Umbabat Private Nature Reserve	134
108	Pilanesberg National Park	87	204	Timbavati Private Nature Reserve	322
109	Tembe Elephant Park	115	205	Sabie Sand Game Reserve	311
110	Pongolapoort Nature Reserve	20	206	Knysna State Forest	3
			207*	Private Reserves	641

* Not shown on map

AfESG

African Elephant Database of IUCN/SSC/AfESG
in collaboration with UNEP/GRID

N

UNEP

143

Figure 1.4: Population estimate in South Africa (found in [1] *African Elephant Database 1998*, page 143)

1.6 Parks carrying capacity

1.6.1 Theoretical concept

As explained in [2], the idealized concept of carrying capacity is one where a population can be supported by the environment over a long period of time. This concept is illustrated by use of the logistic equation (see the s-shape model in Section 2.1.3 page 25) where the asymptotic or equilibrium value is defined as the carrying capacity (denoted K), which is the equilibrium level at which some limiting factors prevent further population growth from occuring. However, in reality, populations never reach a steady asymptote but will show oscillations, regulated by negative feedback, around a mean value. These oscillations can either be short term (seasonal) or long term (climatic cycles) or both, depending on the nature of the factor limiting population growth.

1.6.2 Large herbivore biomass

It is very difficult to find in the literature some methods to calculate the carrying capacity of a park for elephants. However, some work has been done to calculate the large herbivore biomass(LHB in kg/km^2), depending on mean annual precipitations(AP in $mm/year$). Coe et al. (1976) suggest a possible linear relationship for areas below 700 mm by year:

$$LHB = 8.684(\pm 2.28)AP - 1205.9(\pm 156.6) \tag{1.11}$$

Bell (1982) and East (1984) assessed the importance of different soil types. If we concentrate on arid savannas species (which dominate herbivore biomass[4]), the over all regression equations proposed by East are:
For high nutrient soils:

$$Log_{10}(LHB) = -2.0(\pm 0.8) + 2.03(\pm 0.29)Log_{10}(AP) \tag{1.12}$$

For low nutrient soils:

$$Log_{10}(LHB) = -2.5(\pm 0.8) + 2.10(\pm 0.31)Log_{10}(AP) \tag{1.13}$$

For high nutrient soils, the regression equation is correct for areas between 100 mm and 1100 mm but it is not the case for low nutrient soils where the regression equation is correct only for areas below 750 mm. For areas above 750 mm, the large herbivore biomass seems to decrease (due to the high density of trees). I calculated the regression equation which is (for eastern and southern Africa):

[4] Especially in arid/euthropic savannas where they comprise well over 90% of the total biomass on average.

$$Log_{10}(LHB) = -2.0(\pm 0.8) + 2.0(\pm 0.31)Log_{10}(AP) \tag{1.14}$$

Figure 1.5 shows the regression on the high nutrient soil data. Figure 1.6 shows the regression on the low nutrient soil data. Finally, Figure 1.7 displays the graph of all the equations mentioned above.

If the soil types are unknown, it is better to use equation (1.11)[5], whereas if the ratio "High nutrient soils : Low nutrient soils" is known, it is better to use a weighed mean of equations (1.12) and (1.13) (if we are in areas under 750 mm) or (1.14) (if we are above 750 mm).

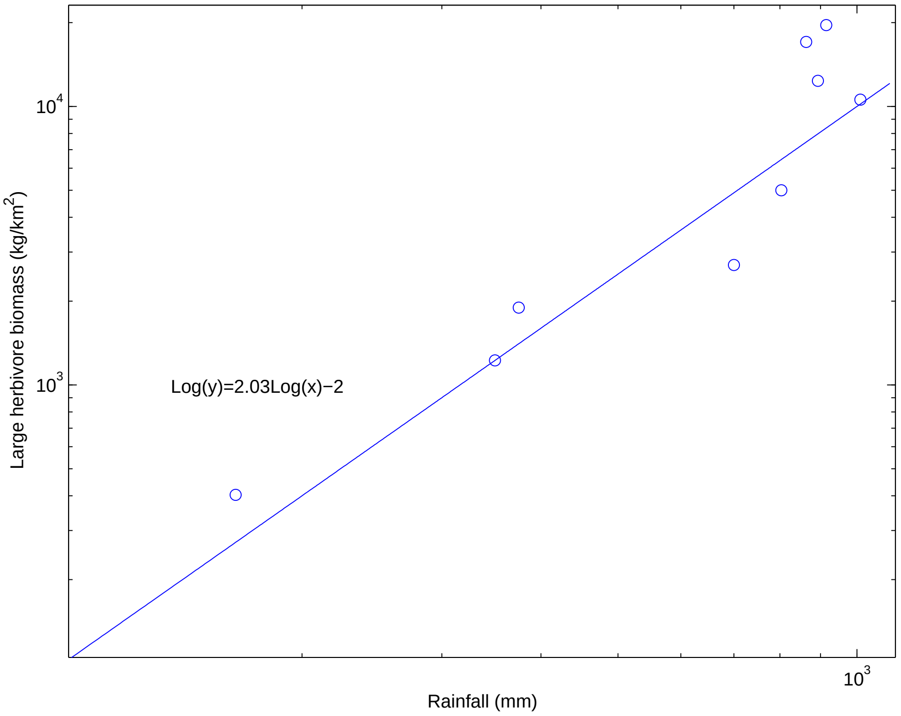


Figure 1.5: Large Herbivore Biomass depending on Annual Precipitations for high nutrient soils, logarithmic scales (data from [5] East)

According to Mr B.R. Page and data from North Eastern Tuli Block (Crispian Chad Cheney (1998)), in most game reserves, elephants constitute the dominant large herbivores, which range between 45% and 70% of the community biomass (60% seems to be a good average). So it is possible to approximate the elephant biomass by multiplying LHB by 0.6.

[5] Only in areas under 700 mm!

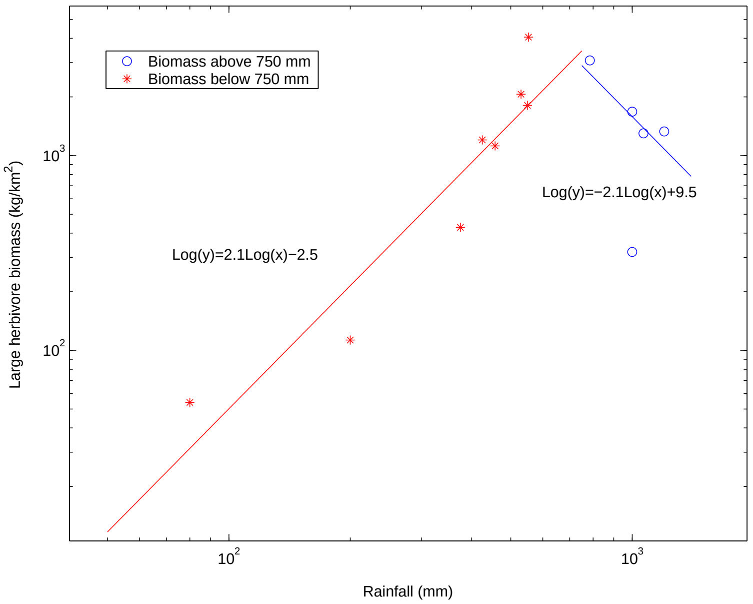


Figure 1.6: Large Herbivore Biomass depending on Annual Precipitations for low nutrient soils, logarithmic scales (data from [5] East)

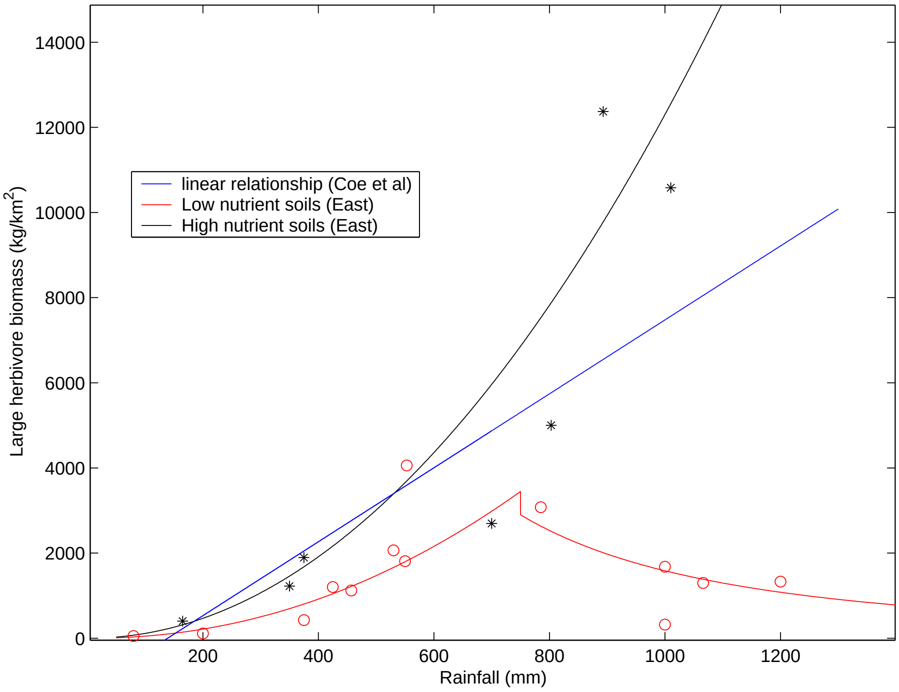


Figure 1.7: Large Herbivore Biomass depending on Annual Precipitations

1.6.3 Body weight

Figure 1.8 plots the body weight-growth curves ($w(x)$, w is the weight in kg, x is the age in years) obtained from these equations ([19] Laws, Parker and Johnstone (1975)):

$$\begin{aligned} \text{male (1-20 years): } w(x) &= 3112(1-e^{-0.114(x+3.95)})^3\ kg \quad (1.15)\\ \text{male (20-60 years): } w(x) &= 4742(1-e^{-0.166(x-10.48)})^3\ kg \quad (1.16)\\ \text{female (2-60 years): } w(x) &= 2744(1-e^{-0.099(x+6.00)})^3\ kg \quad (1.17)\end{aligned}$$

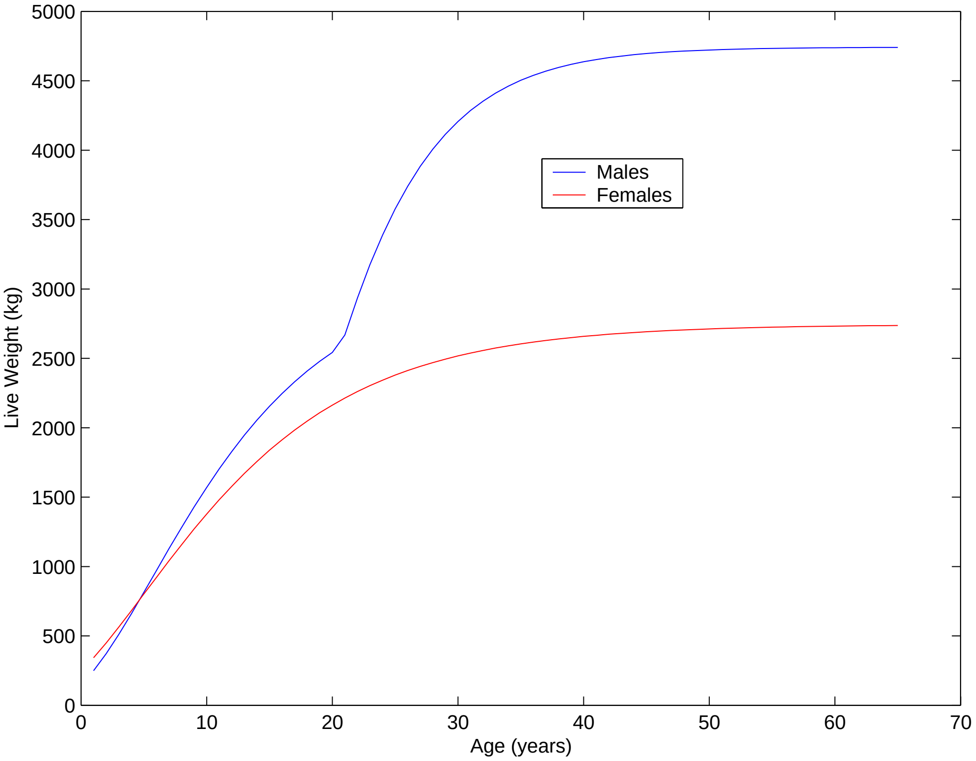


Figure 1.8: Body weight-growth curves

So if we know the age distribution of a given population (percentage of elephants in each age and sex group), it is possible to approximate the mean weight. If we denote $ad_m(x)$ the age distribution for males at age x and $ad_f(x)$ the age distribution for females at age x, we can then calculate the mean weight for males ($\overline{w_m}$) and females ($\overline{w_f}$) using these formulas:

$$\begin{aligned}
\overline{w_m} &= \sum_{x=1}^{20} ad_m(x) \times 3112(1 - e^{-0.114(x+3.95)})^3 \\
&\quad + \sum_{x=21}^{k} ad_m(x) \times 4742(1 - e^{-0.166(x-10.48)})^3 \qquad (1.18) \\
\overline{w_f} &= \sum_{x=1}^{k} ad_f(x) \times 2744(1 - e^{-0.099(x+6.00)})^3 \qquad (1.19)
\end{aligned}$$

where k is the maximum age reached by an elephant of the population (as denoted in Chapter 2).

And so if we denote s_m the sex ratio of males and s_f the sex ratio of females of the elephant population, we can then deduce the mean weight of an elephant ($\overline{w}$):

$$\overline{w} = s_m \times \overline{w_m} + s_f \times \overline{w_f} \qquad (1.20)$$

1.6.4 Carrying capacity calculus

With all the equations of the last subsections, it is now possible to calculate the carrying capacity (CC in number of elephants) of a park, given the mean annual precipitations (in mm), the age distribution (no unit), the sex ratio (no unit), the "high nutrient soils : low nutrient soils" ratio (no unit) , and the area (km^2) of the park:

$$CC = 0.6\frac{LHB}{\overline{w}}area \qquad (1.21)$$

where LHB and $\overline{w}$ are calculated as explained in the last two subsections.

1.7 The Amarula Elephant Research Programme

The African elephant is a symbol of Africa, evoking images of a life uninfluenced by modernization and industrialization. The South African producer of Amarula cream has been intimately linked to the African elephant. Its label has the elephant as its main icon and the Marula fruit, from which Amarula is derived, is favoured by elephant in the region.

Amarula has recently announced its sponsorship of more than 3 000 000 Rands (around 300 000 Euros) over the next five years for the Amarula Elephant Research programme, based in the School of Life and Environmental Sciences at the University of Natal, Durban. This contribution will build on an existing relationship with the University of Natal that goes back to 1998.

The Programme enables a team of South African and international academics to work in close partnership with Government Conservation Agencies and Private Game Reserves and focuses on solving key elephant conservation questions:

- Stopping Elephants Killing Rhino

  At Pilansberg National Park and Hluhluwe-Umfolozi Park, elephants have killed over 100 rhino since 1992. The reason is that the elephants were introduced from Kruger Park in the 1980's as orphans from the culling operations there, and youngsters matured in the absence of older male elephants. These youngsters have been entering musth, which is a heightened and aggressive hormonal state, much younger than normal. While in musth, they were killing rhino. This problem was solved in both reserves through the introduction of older male elephants, who suppressed the musth behaviour of the delinquent youngsters.

- Greater St. Lucia Wetland Park Elephant Introduction

  St. Lucia has been declared a World Heritage Site, and has been selected as a focal area for ecotourism development as part of the Lubombo Spatial Development Initiative. Elephants are a key component of this ecosystem, and are being introduced to St. Lucia as part of a programme of restocking animals that previously occurred there. The Programme is monitoring and researching the animals following their release.

- Stress in Elephants

  A new technique allows to measure the stress levels of elephants by collecting dung samples, and then analyzing the hormones within the dung. The Programme collects dung from all of the project sites, and analysis allows to assess the influence of the environment on the elephant. Particularly, the programme focus on the influence of management actions on the welfare of the elephant. Such information will allow more humane conservation of elephants in small reserves in the future.

- Conservation of Elephants

  Centuries ago, the African elephant (Loxodonta africana) inhabited most of the African continent. Today, elephants are found only south of the Sahara Desert and their range is fragmented and discontinuous. As a result of this range loss and fragmentation, along with poaching of elephants for ivory, elephant numbers declined across the continent. By the 1970s, the decline had provoked serious concern about the long-term survival of the species.

  In south Africa, the problem is completely different. Elephants are becoming increasingly restricted to small, fenced reserves. Because the poaching has been nearly stopped, the number of elephants is growing exponentially in these reserves and because of their large size and social nature, elephants pose unique problems to reserve managers. Elephants consume a large amount of vegetation, and also knock over very large trees. The small reserves have only a limited amount of vegetation, and conservationists fear that the elephant will irreversibly change the habitat by destroying all the trees. Even in bigger reserves such as the Kruger National Park, population growth are actively limited through fencing and culling (including translocation) campaigns. An alternative to culling could be contraception and some studies are being led in this domain (See the current passionate debate involving elephants in Botswana in Appendix E).

  A primary goal of the Programme is to work with reserve managers and ecologists to generate elephant management plans for each reserve. These plans, based on data collected, will provide managers with guidelines to conserve elephants in a large number of small reserves in a responsible and sustainable manner.

Chapter 2

Different models

This chapter describes the four different models used in the program simulele. There is two models in continuous time (the exponential model and the s-shape model) and two models in discrete time (the Leslie Matrix model and the Age and State model). These models are classed from the simplest one to the most complex one.

2.1 General models in continuous time

2.1.1 Differential equation

The total number of elephants (n) in a fixed region of space can only change for four reasons:

- births (b),
- deaths (d),
- immigration (i),
- emigration (e).

In a small time interval Δt the change in population (Δn) can thus always be written as

$$\Delta n = (b - d + i - e) \cdot \Delta t \tag{2.1}$$

where $b\Delta t$, $d\Delta t$ are respectively the total numbers of births and deaths during Δt, while $i\Delta t$ and $e\Delta t$ are the total number of individuals entering and leaving the region during the same time interval.

Since we are interested by elephant populations in game reserves (fenced areas), we can eliminate immigration and emigration.

As we have seen in Chapter 1, elephants have no serious predators so deaths do not depend on another population species.

To simplify, suppose n is a function of t and b and d depend only on $n(t-\Delta t)$,

We obtain this equation:

$$\Delta n = (b(n(t - \Delta t)) - d(n(t - \Delta t))) \cdot \Delta t \tag{2.2}$$

We obtain a differential equation by assuming that we may consistently let ΔN and Δt become very small. We then write

$$\frac{dn}{dt} = b(n(t)) - d(n(t)) \tag{2.3}$$

2.1.2 Exponential model

The simplest case that we can imagine is when $b(n(t)) - d(n(t))$ depends linearly on $n(t)$. Suppose then that

$$b(n(t)) - d(n(t)) = r \cdot n(t) \tag{2.4}$$

where r is the growth rate. The solution of the differential equation is

$$n(t) = n_0 \cdot e^{rt} \tag{2.5}$$

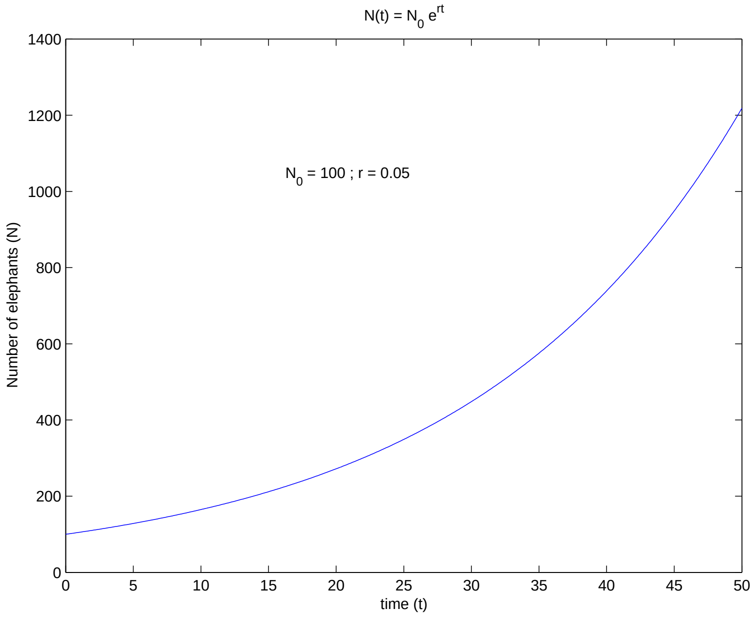


Figure 2.1: Exponential model

But in a game reserve, the maximum size of a population is finite because resources are finite. So this model is not relevant when the size of the population exceeds this maximum value.

2.1.3 S-shape model

If the maximum size of the population (or carrying capacity) is K, we can construct this differential equation known as the logistic equation:

$$\frac{dn}{dt} = rn(1 - \frac{n}{K}) \tag{2.6}$$

And the solution can be found as follows:

$$\begin{aligned}
\frac{dn}{n(K-n)} &= \frac{r}{K}dt \\
\int_{n_0}^{n}(\frac{1}{n} + \frac{1}{K-n})dn &= \int_0^t rdt \\
[ln(n) - ln(K-n)]_{n_0}^{n} &= rt \\
\frac{n(K-n_0)}{n_0(K-n)} &= e^{rt} \\
n(t) &= \frac{K}{1+(\frac{K-n_0}{n_0})e^{-rt}} \qquad (2.7)
\end{aligned}$$

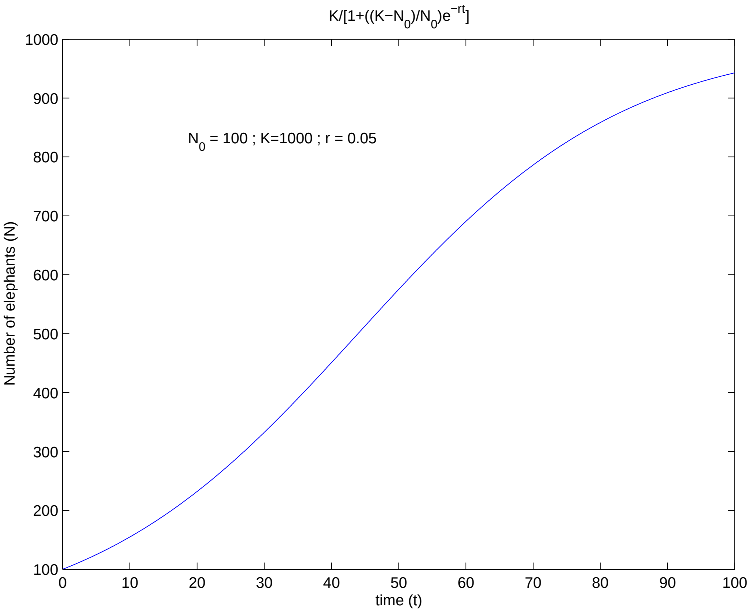


Figure 2.2: s-shape model

2.1.4 Advantages and disadvantages

These models are very easy to program but they are extreme idealizations and they have no sense if the age-structure of the population does not remain constant. In most cases that interest us, the age-structures are not constants because we are looking at very young populations (such as St Lucia or the Pilanesberg) and we need to take the age structure it to account. It can be done using discrete models such as the Leslie Matrix one.

2.2 Leslie Matrix model

2.2.1 Life Tables

As we have seen in Chapter 1, elephant populations are aged structured, it means that the probability of surviving and the fecundity depend essentially on age. They depend also on the environment (fecundity and rainfall can be related for example) but the lack of data does not permit to include these parameters in the model for the moment.

We can summarize the probability of surviving and the fecundity in a table called Life Table, or l_x-Table.

x: Age category ($x = 1, 1, ..., k$). These values can be in any convenient units (we will work in years), but must be chosen so that no observed lifespan extends past category k.

p_x: Probability of surviving the interval $[x, x+1]$. We have $p_k = 0$.

b_x: Fecundity (or average number of offspring produced by an individual in age category x while in that age category).

l_x: "Survivorship", the proportion of individuals reaching age category $x+1$.

$$l_x = p_1 p_2 p_3 ... p_x = \prod_{i=0}^{x-1} p_i \tag{2.8}$$

We have $l_k = 0$.

(Table 2.2.1 on next page is an example of a Life Table)

2.2.2 Immediate information extracted from a life table

We can extract some interesting information from a Life Table.

Net reproductive rate per generation

We can notice that $l_x b_x$ is the average number of offsprings produced by an individual within age category x weighted by the probability of surviving to the beginning of the next age category.
Therefore, the sum

$$R_0 = \sum_{x=1}^{k} l_x b_x \tag{2.9}$$

is the average lifetime number of offspring produced by a member of the study population, or net reproductive rate per generation.

Age category(years)	p_x	l_x	b_x
1	0.90	0.90	0
2	0.99	0.891	0
3	0.99	0.882	0
4	0.99	0.873	0
5	0.99	0.865	0
6	1	0.865	0
7	1	0.865	0
8	1	0.865	0
9	1	0.865	0
10	1	0.865	0.033
11	1	0.865	0.061
12	1	0.865	0.084
13	1	0.865	0.104
14	1	0.865	0.120
15	1	0.865	0.133
16	1	0.865	0.144
.	.	.	.
.	.	.	.
.	.	.	.
45	1	0.865	0.200
46	0.950	0.821	0.200
.	.	.	.
.	.	.	.
.	.	.	.
55	0.950	0.518	0.200
56	0.900	0.466	0
.	.	.	.
.	.	.	.
.	.	.	.
65	0	0	0

Table 2.1: Example of a Life Table for an elephant

Mean generation time of the population

We can notice that xl_xb_x is a column weighting the offspring counted in the previous column by their parents' age when they were born.
Therefore, the ratio

$$T = \frac{\sum xl_xb_x}{\sum l_xb_x} \tag{2.10}$$

is the mean generation time of the population.

Malthusian parameter

It measures the reproductive rate per unit time and can be calculated as:

$$r = \frac{ln(R_0)}{T} \tag{2.11}$$

For an exponentially increasing population, the population size $n(t)$ at time t is then given by:

$$n(t) = n_0e^{rt} \tag{2.12}$$

Annual growth rate

We denote it g and we can deduce it from the Malthusian parameter r[1]. If t is in years, we have

$$\begin{aligned} n(t) &= n(t-1) + g \cdot n(t-1) = n(t-1) \cdot (1+g) \qquad (2.13) \\ &= n_0(1+g)^t \\ &= n_0e^{ln(1+g)t} \\ &= n_0e^{rt} \end{aligned}$$

By identification, we can see that $r = ln(1+g)$ and by consequence

$$g = e^r - 1 \tag{2.14}$$

Notice that if r is very small, g and r are very close because $e^r \sim r + 1$.

2.2.3 The Leslie Matrix

We will make the assumption that the dynamics of the population are determined by females independent of the relative abundance of males (there will always be enough males to fertilize the females). So we start by modelling the female population alone.

[1] These are approximate estimations of r and g because we used a simplified assumption that generations are discrete. Accurate estimations of r and g will be discussed in Section 2.2.4 (see equation (2.30) and (2.29) page 31).

Let s_f and s_m denote the sex ratio at birth for females and males. Then the average numbers of females and males produced by an individual within age category x are

$$b_{fx} = b_x \cdot s_f \tag{2.15}$$

$$b_{mx} = b_x \cdot s_m \tag{2.16}$$

Let $n_{fx}(t)$ and $n_{mx}(t)$ denote the number of females and males of exactly age x at time t. Remember that $n_{fk} = n_{mk} = 0$. The female and male populations at time t can be represented by two vectors $N_f(t)$ and $N_m(t)$:

$$N_f(t) = \begin{pmatrix} n_{f1}(t) \\ n_{f2}(t) \\ \vdots \\ n_{fk}(t) \end{pmatrix} \quad ; \quad N_m(t) = \begin{pmatrix} n_{m1}(t) \\ n_{m2}(t) \\ \vdots \\ n_{mk}(t) \end{pmatrix} \tag{2.17}$$

We will also denote by $n_f(t)$ and $n_m(t)$ the total number of males and females at time t:

$$n_f(t) = \sum_x n_{fx}(t) \qquad ; \qquad n_m(t) = \sum_x n_{mx}(t) \tag{2.18}$$

Using the same notation as in the Life Table, we begin with two simple relationships:

$$n_{fx}(t) = n_{f(x-1)}(t-1)p_{x-1} \qquad where \;\; x > 1 \tag{2.19}$$

$$n_{f1}(t) = \sum_{x=1}^{k} n_{fx}(t-1)p_x b_{fx} \tag{2.20}$$

Equation (2.19) should be obvious.

Equation (2.20) gives the number of individuals in the first age class as the sum of newborn by parents of all ages.

These relationships can now be put into matrix form:

$$\begin{pmatrix} n_{f1}(t) \\ n_{f2}(t) \\ \vdots \\ \vdots \\ n_{fk}(t) \end{pmatrix} = \begin{pmatrix} p_1 b_{f1} & p_2 b_{f2} & p_3 b_{f3} & \dots & p_k b_{fk} \\ p_1 & 0 & 0 & \dots & 0 \\ 0 & p_2 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \vdots \\ 0 & 0 & \dots & p_{k-1} & 0 \end{pmatrix} \cdot \begin{pmatrix} n_{f1}(t-1) \\ n_{f2}(t-1) \\ \vdots \\ \vdots \\ n_{fk}(t-1) \end{pmatrix}$$

or more conveniently,

$$N_f(t) = L \cdot N_f(t-1) \tag{2.21}$$

The matrix L is the Leslie matrix. It belongs to a class of projection matrices, named for the fact that it projects the population vector at one time onto a new population vector in the next time unit.

We can then simply express $N_f(t)$ in function of the initial female population $N_f(0)$:

$$N_f(t) = L^t \cdot N_f(0) \tag{2.22}$$

2.2.4 Properties of the Leslie Matrix

Interesting generalizations can be made about the eigenvalues and eigenvectors of the Leslie matrix:

- The eigenvalues of L are given by the solutions to an equation of the form
$$1 = \sum_{x=1}^{k} \lambda^{-x} l_x b_{fx} \tag{2.23}$$
(See proof of theorem 1 in Appendix A)

- Equations of the form (2.23) have one and only one real positive root (denoted λ_1 and also known as the Perron root). All other roots are either negative or complex and therefore describe population oscillations. Moreover, we have $\lambda_1 > |\lambda_i|$ (see proof[2] of theorem 2 in Appendix A).

- The right eigenvector (let denote it U_1) of L associated to λ_1 can easily be calculated, writing
$$L \cdot U_1 = \lambda_1 U_1 \tag{2.24}$$
If we arbitrarily set the first coordinate $u_1(1)$ equal to one, we find $p_1 u_1(1) = \lambda_1 u_1(2)$, so that $u_1(1) = \frac{p_1}{\lambda_1}$. And by induction,
$$u_1(i) = p_1 p_2 ... p_{i-1} \lambda_1^{-i} \tag{2.25}$$

- The left eigenvector (let denote it V_1) of L associated to λ_1 can also easily be calculated, writing
$$V_1' \cdot L = \lambda_1 V_1' \tag{2.26}$$
If we arbitrarily set the first coordinate $v_1(1)$ equal to one, we find for each $i \in \{1 \, ... \, k-1\}$: $p_i b_{fi} + p_i v_1(i+1) = \lambda_1 v_1(i)$. So we have the following formula
$$v_1(i+1) = \frac{\lambda_1 v_1(i) - b_{fi}}{p_i} \tag{2.27}$$
with $v_1(1) = 1$.

[2] To prove this last property, we could use the well-known theorem of Perron-Frobenius which comes from the theory of matrices. This theorem is the theorem 3 of Appendix A but we will omit the proof which is quite technical.

- The results of the previous properties lead us to a stable age distribution.

  – If the Leslie Matrix is diagonalizable, the general solution to an equation such as equation (2.22) is then

$$n_{fi}(t) = \sum_j C_{ij} \lambda_j^t \tag{2.28}$$

where λ_j are the eigenvalues of L and C_{ij} are constants whose values depend on the initial population vector (See proof of theorem 4 in Appendix A).
But this says that once oscillations die out, we have

$$N_f(t) = C\lambda_1^t$$

So each age group grows at the same rate as every other, and we say that the population is in stable age distribution. Furthermore, the vector N_f for which the above is true,

$$N_f(t+1) = L \cdot N_f(t) = \lambda_1 N_f(t)$$

is the eigenvector of L associated with λ_1. Therefore the stable age distribution is given by the right eigenvector U_1 associated with the Perron root λ_1. Finally, when the population is in stable age distribution, so that

$$n_{fi}(t+1) = n_{fi}(t)\lambda_1$$

for all i, it follows that

$$n_f(t+1) = \sum_i n_{fi}(t+1) = \lambda_1 \sum_i n_{fi}(t) = \lambda_1 n_f(t)$$

We can see here that once the population is in a stable age distribution, the population number stay the same if and only if $\lambda_1 = 1$, we say that the population is stationary.

If we compare the last equation with equation 2.13, we see that $1 + g$ equals λ_1. So we can make a better estimation of g and r than in Section 2.2.2 (equation (2.14) and (2.11)). We have

$$g = \lambda_1 - 1 \tag{2.29}$$

and

$$r = ln(\lambda_1) \tag{2.30}$$

– If L is not diagonalizable, it is possible to prove that we obtain the same results even if it is more complicated. Moreover, in Appendix A theorem 5, we prove that

$$\frac{L^n}{\lambda_1^n} = U_1 V_1' + o(\lambda_1^n) \tag{2.31}$$

where U_1 and V_1 are the right and left eigenvectors of L. The following application of this result is called the Fundamental theorem of Demography (see Appendix A theorem 6). In the special case of the Leslie Population, we have

$$N_f(t) = L^t \cdot N_f(0) = \lambda_1^t V_1' \cdot N_f(0) \cdot U_1 + o(\lambda_1^t) \tag{2.32}$$

2.2.5 Introducing males into the Leslie Model

Introducing males into the Leslie Model is not very difficult, but I could not find any example in the literature. The difference with the female population is that the male population at time $t+1$ depends on the male population at time t, but depends also on the female population at time t (females will give birth to males calves). We have two relationships similar to equations (2.19) and (2.20):

$$\begin{aligned} n_{mx}(t) &= n_{m(x-1)}(t-1)p_{x-1} \qquad \textit{where} \;\; x > 1 && (2.33) \\ n_{m1}(t) &= \sum_{x=1}^{k} n_{fx}(t-1)p_x b_{mx} && (2.34) \end{aligned}$$

We can again put these relationships into matrix form:

$$\begin{pmatrix} n_{m1}(t) \\ n_{m2}(t) \\ \vdots \\ \vdots \\ n_{mk}(t) \end{pmatrix} = \begin{pmatrix} p_1 b_{m1} & p_2 b_{m2} & p_3 b_{m3} & \dots & p_k b_{mk} \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} n_{f1}(t-1) \\ n_{f2}(t-1) \\ \vdots \\ \vdots \\ n_{fk}(t-1) \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ p_1 & 0 & 0 & \dots & 0 \\ 0 & p_2 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \vdots \\ 0 & 0 & \dots & p_{k-1} & 0 \end{pmatrix} \cdot \begin{pmatrix} n_{m1}(t-1) \\ n_{m2}(t-1) \\ \vdots \\ \vdots \\ n_{mk}(t-1) \end{pmatrix} \tag{2.35}$$

If we use L_u to denote[3] the upper part (first row) of the Leslie Matrix L (the same as used in equation 2.21), and L_l to denote the lower part (all

[3] L_u and L_l are sometimes written as F and T for the Fertility matrix and the Transition matrix.

the other rows) of L we can rewrite the last equation as:

$$N_m(t) = \frac{s_m}{s_f} L_u \cdot N_f(t-1) + L_l \cdot N_m(t-1) \tag{2.36}$$

where s_m and s_f are the sex ratio at birth of males and females.

Using equation (2.22), we can express $N_m(t)$ as a function of the initial population of males $N_m(0)$ and females $N_f(0)$:

$$\begin{aligned} N_m(t) &= \frac{s_m}{s_f} L_u \cdot L^{t-1} \cdot N_f(0) + L_l \cdot N_m(t-1) \qquad (2.37) \\ &= \frac{s_m}{s_f} \sum_{i=1}^{t} L_l^{i-1} \cdot L_u \cdot L^{t-i} \cdot N_f(0) + L_l^t \cdot N_m(0) \qquad (2.38) \end{aligned}$$

It is easy to show that the growth rate and the stable age distribution of the male population are the same as for the female population.

At $t = 1$, using equations (2.34), (2.20), (2.15) and (2.16) we have

$$\begin{aligned} n_{m1}(1) &= \sum_{x=1}^{k} n_{fx}(t-1) p_x b_{mx} \\ &= \sum_{x=1}^{k} n_{fx}(t-1) p_x b_{fx} \frac{s_m}{s_f} \\ &= n_{f1}(1) \frac{s_m}{s_f} \end{aligned}$$

At $t = 2$, using (2.34), (2.20), (2.15), (2.16) and (2.33), we have

$$\begin{aligned} n_{m1}(2) &= n_{f1}(2) \frac{s_m}{s_f} \\ n_{m2}(2) &= n_{m1}(1) p_1 \\ &= n_{f1}(1) \frac{s_m}{s_f} p_1 \\ &= n_{f2}(2) \frac{s_m}{s_f} \end{aligned}$$

With the same reasoning, at $t = k$ we have

$$\begin{aligned} n_{m1}(k) &= n_{f1}(k) \frac{s_m}{s_f} \\ n_{m2}(k) &= n_{f2}(k) \frac{s_m}{s_f} \\ &\cdots \\ n_{mk}(k) &= n_{fk}(k) \frac{s_m}{s_f} \end{aligned}$$

So that

$$N_m(k) = N_f(k)\frac{s_m}{s_f}$$

And more generally for $t \geq k$, we have

$$\begin{aligned} N_m(t) &= N_f(t)\frac{s_m}{s_f} & (2.39) \\ &= L^t N_f(0)\frac{s_m}{s_f} & (2.40) \end{aligned}$$

Now, we understand why the growth rate and the stable age distribution are the same for male and female populations.

2.2.6 Advantages and disadvantages

As we have seen, this model is completely deterministic. It is easy to develop a theory based on the Leslie matrix but this model is still too simplistic. For example, it does not incorporate density feedback. So this model must be used when the population number is quite far from the carrying capacity of the game reserve. In a way, this model is the discrete version of the exponential model in continuous time.

Attempts to incorporate density feedback do a certain violence to the simple elegance of the Leslie matrix approach. Pencuick et Al. (1968) performed computer simulations using the Leslie matrix and defining

$$b_{fx} = \frac{B_{fx}a}{c + n_f}p_x = \frac{P_x}{1 + e^{n_f/d-f}}$$

where B_{fx} is fecundity, P_x survival in the absence of density feedback, and a, c, d and f are constants. Depending on the values of B_{fx}, P_x,a, c, d and f chosen, they obtained sigmoid growth with no oscillations,damped oscillations or stable oscillations. In itself this information is not terribly useful although if we knew the true form of feedback on b_{fx} and p_x, this simulation approach might be of considerable value.

We also would like the model to be less deterministic and take into account the environmental conditions such as the rainfall for example. It would also be better to take into account the pregnancy length and the duration of anoestrous. This cannot be done with Leslie matrices so we developed another model that we called Age and State model based on a Fortran program originally coded by Professor Botkin (University of California, Santa Barbara).

2.3 Age and State model

2.3.1 Main idea

The main idea of this model is to class the population in a table with different states. In the Leslie model, the population is described by two vectors $N_f(t)$ and $N_m(t)$. Here $N_m(t)$ is the same but $N_f(t)$ is split in different vectors corresponding to the state of the female: immaturity, maturity, pregnancy and anoestrous lactation.

2.3.2 Population table

The whole population $N(t)$ can be described by an allocatable[4] array like in Table 2.2, filled with integers:

Age	Males	I.F	M.F	P.F #1	...	P.F #i	L.F #1	...	L.F #j
1									
2									
$\vdots$									
k									

Table 2.2: Population table for the Age and State model

Legend:
I.F: Immature females
M.F: Mature females (not pregnant, not lactating)
P.F #1: Pregnant females in the first year of pregnancy
P.F #i: Pregnant females in the i^{th} year of pregnancy
L.F #1: Lactating females in the first year of lactation
L.F #j: Lactating females in the j^{th} year of lactation

According to the probabilities of surviving, and the fecundity (denoted p_x and b_x[5] in the life table, Section 2.2.1, page 26), this table is evolving each year. This model is no more deterministic, but stochastic because we use a random number generation to determine for each elephant if it will survive to the next year, and if it will change its state (for example, if a female will be mature or pregnant). This model is better than the Leslie Matrix model because the pregnancy length and the lactation length are taken into account, but it is still possible to improve it.

[4] The maximum age and the number of columns are not fixed.

[5] b_x has to be modified so that to take into account the delays due to pregnancy length and anoestrous lactation length which are not present in the Leslie Matrix model

2.3.3 Handling orphan calves

A sad but true reality concerns orphan calves. Most of the time, if a female which is lactating a very young calf dies, the calf will starve to death. So in this model, it is possible to kill orphan calves whose mothers were lactating.

2.3.4 Environmental conditions

We saw in Chapter 1 Section 1.3 and 1.4 that the mortality and the fecundity are sensitive to environmental conditions among which rain is of paramount importance. For a manager, it can be interesting to test different scenarios of hypothetical rainfall to see how it is affecting the population growth. Unfortunately, the amount of data is insufficient to build equations relating rainfall and mortality or fecundity. But it has been decided to try including environmental conditions in the model, and the ideal would be to improve the relationship with the mortality and fecundity in the future.

To make it simple, an excellent year (very high rainfall) can be represented by 1, a normal year (usual rainfall) by 0 and a catastrophic year (drought, epidemic) by -1, so that all the real numbers between -1 and 1 describe all possible environmental conditions. Best, normal and worst parameters (for survivorship and fecundity) corresponding to best, normal and worst conditions have to be chosen. It was decided that both best and worst parameters would be set as constants in the program (Table 3.1 page 64) whereas the user would set normal parameters. For any environmental conditions between -1 and 0 or 0 and 1, corresponding parameters can then be calculated using linear interpolation.

2.3.5 Density dependence

An attempt has been made to include a sort of density dependence into the Age and State model.

If the population size is under the carrying capacity of the park, we do not take into account years with bad environmental conditions and we use the normal parameters because we can consider that elephants will have enough food to live.

However, if the population size is above the carrying capacity, bad years have bad consequences whereas during good years, we use normal parameters.

In other words, if the population size is under the carrying capacity, the parameters will vary between the normal and the best ones, and if the population size is above the carrying capacity, the parameters will vary between the normal and the worst ones.

This process, corresponding to a biological reality, can also be improved in the future. It could be an interesting and useful project to relate precisely rainfall and carrying capacity with mortality and fecundity, but a large amount of data is needed.

2.3.6 Advantages and disadvantages

It is much more difficult to develop a theory (like the one with the Leslie Model) but it is quite intuitive that if we do not kill the orphan calves, if we do not use the environmental conditions and if we do not use the density dependence, then the mean number of elephants obtained running several times this model tends to be the same as the number of elephants obtained running the Leslie Model. This is so because we use exactly the same parameters of the life table and the only difference comes from the delay due to pregnancy length and anoestrous lactation length.

The advantage of this model is that it is much more realistic because it takes into account the environmental conditions and there is a density dependence, so that we can see some oscillations around the carrying capacity appearing. In a way, this model is the discrete version of the s shape model. Moreover, because of the delay due to pregnancy length, this model permits to observe phenomenon such as steps in the evolution of the population number, or group formation in the age distribution.

Chapter 3

Programming with Matlab and Fortran

This chapter describes how the models of the previous chapter were coded.

3.1 Needs of the Amarula Elephant Research team

The Amarula Elephant Research team needed a software that they could use from any computer so that the students and research members in the different game reserves could use it. Graphical outputs was also required so that the results could be analysed more easily and included in any report or thesis. I also decided to build a graphical user interface (GUI). By providing an interface between the user and the application's underlying code, GUIs enable the user to operate the application without knowing the commands that would be required by a command line interface. For this reason, applications that provide GUIs are easier to learn and use than those that are run from the command line.

So I had to build an executable program (*.exe) with a graphical user interface and graphical outputs.

3.2 Programming possibilities and choices

I worked on a computer of the behavioural ecology laboratory of the biology department. The operating system installed was Windows 98 and I could use Fortran PowerStation 4.0, Matlab 6.5 and Mathematica 4.2.

My knowledge of Fortran was limited but I knew that graphics were quite poor in comparison with Matlab. Moreover, I had already used Matlab to build a GUI for the "Laboratoire National des Champs Magnétiques Pulsés" of Toulouse (FRANCE) and so Matlab was much more familiar to me. Matlab seemed to be accurate for the Leslie Model which is very vectorized. That is why I decided to build my main program with Matlab and to compile it as an executable file (simulele.exe) as explained in section 3.3.3.

I had to upgrade and use the Fortran program running the Age and State model originally made by Professor Botkin so I used Fortran Powerstation to modify it and compile it as an executable called ModelingEle7.exe. So the program simulele.exe calls ModelingEle7.exe and data are exchanged by text files as explained in Section 3.5.

3.3 Programming with Matlab

3.3.1 Graphical user interface: simulele.fig

A graphical user interface (GUI) is a user interface built with graphical objects – the components of the GUI – such as buttons, text fields, sliders, and menus. In order to manipulate information contained in this components, Matlab use handles on them.

If the GUI is well-designed, it should be intuitively obvious to the user how its components function.

Guide

The graphical user interface was realized with GUIDE version 6, the MATLAB Graphical User Interface Development Environment. GUIDE provides a set of tools for creating GUIs. These tools greatly simplify the process of laying out and programming them. The Layout Editor enables to lay out a GUI quickly and easily by dragging components, such as push buttons, pop-up menus, or axes, from the component palette into the layout area. Once I designed my GUI and set each component's properties, using the tools in the Layout Editor, I programmed the GUI with the M-file Editor.

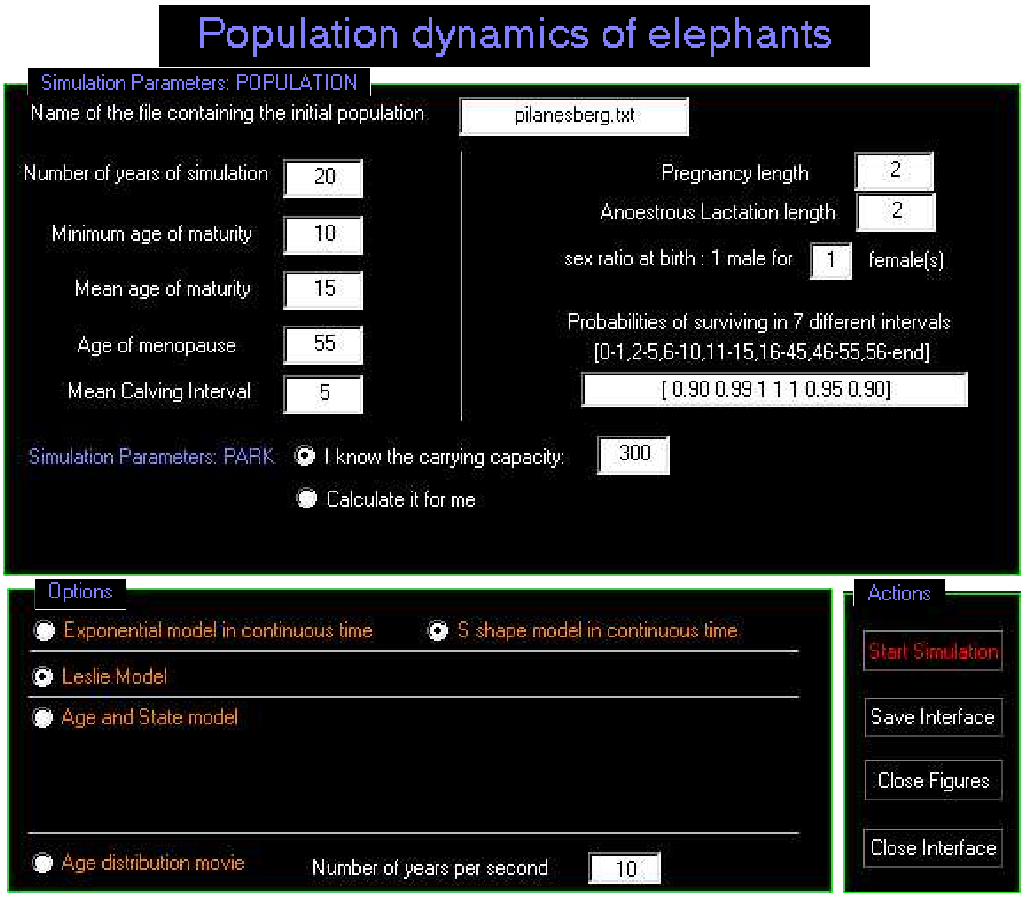
GUIDE stores GUIs in two files, which are generated the first time the GUI is saved or run:

1. FIG-file: a file with extension .fig that contains a complete description of the GUI figure layout and the components of the GUI: push buttons, menus, axes, and so on.

2. M-file: a file with extension .m containing the code that controls the GUI, including the callbacks for its components. This file is referred to as the GUI M-file. When the GUI is run for the first time from the Layout Editor, GUIDE generates the GUI M-file with blank stubs for each of the callbacks. The callbacks can then be programmed using the M-file editor.

The interface

I chose to construct three different frames in the interface in order to separate the Parameters, the Options and the Actions. These three frames are displayed on the same page so that the user can see all the information at the same time. Nevertheless, it was chosen to hide parameters that the user does not need to give according to the options chosen, in order to make the interface more readable. For example, Figure 3.1 page 41 shows the interface with only some options selected (so that some parameters are hidden) whereas Figure 3.2 page 42 shows the same interface with all the options selected (so that no parameters are hidden).



Population dynamics of elephants

Simulation Parameters: POPULATION

Name of the file containing the initial population: pilanesberg.txt

Number of years of simulation: 20

Minimum age of maturity: 10

Mean age of maturity: 15

Age of menopause: 55

Mean Calving Interval: 5

Pregnancy length: 2

Anoestrous Lactation length: 2

sex ratio at birth : 1 male for 1 female(s)

Probabilities of surviving in 7 different intervals [0-1,2-5,6-10,11-15,16-45,46-55,56-end]: [0.90 0.99 1 1 1 0.95 0.90]

Simulation Parameters: PARK

● I know the carrying capacity: 300

○ Calculate it for me

Options

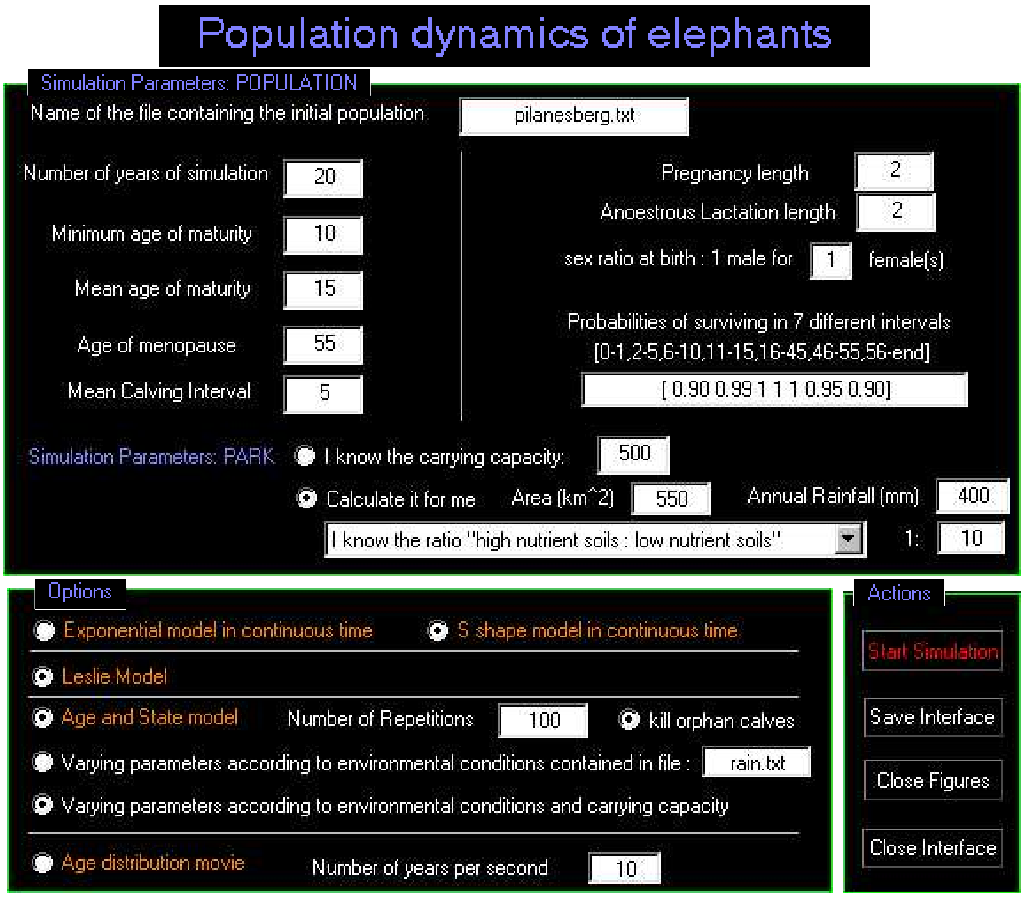
○ Exponential model in continuous time ● S shape model in continuous time

● Leslie Model

○ Age and State model

○ Age distribution movie — Number of years per second: 10

Figure 3.1: The GUI appearance with some parameters hidden



Population dynamics of elephants

Simulation Parameters: POPULATION

Name of the file containing the initial population: pilanesberg.txt

Number of years of simulation: 20

Minimum age of maturity: 10

Mean age of maturity: 15

Age of menopause: 55

Mean Calving Interval: 5

Pregnancy length: 2

Anoestrous Lactation length: 2

sex ratio at birth : 1 male for 1 female(s)

Probabilities of surviving in 7 different intervals [0-1,2-5,6-10,11-15,16-45,46-55,56-end]: [0.90 0.99 1 1 1 0.95 0.90]

Simulation Parameters: PARK

○ I know the carrying capacity: 500

● Calculate it for me Area (km^2) 550 Annual Rainfall (mm) 400

I know the ratio "high nutrient soils : low nutrient soils" 1: 10

Options

○ Exponential model in continuous time ● S shape model in continuous time

● Leslie Model

● Age and State model Number of Repetitions 100 ● kill orphan calves

○ Varying parameters according to environmental conditions contained in file : rain.txt

● Varying parameters according to environmental conditions and carrying capacity

○ Age distribution movie Number of years per second 10

Figure 3.2: The GUI appearance with all parameters shown

Parameters (population)

It has been chosen to use the same parameters for all the models discussed in Chapter 2. It implies that even to run the simplest model (exponential model in continuous time), the user needs to enter the parameters used for the most complicated one (Age and State model). The parameters needed to run the simulation are:

- Name of the file containing the initial population: this file must be a text file and has to be placed in the main folder (where the program simulele.exe is running). The initial population has to be a table similar to the Table 2.2 page 35. This table must contain integers only and **the number of rows of the table determines the maximum age that an elephant can reach** (denoted k in Chapter 2). The number of columns must also correspond to the parameters entered, so that it is equal to 3+Pregnancy length+Anoestrous Lactation length. The integers in this file must be space delimited or tab delimited but it is advised to use the excel file NewInitialPopulation.xls to build a new initial population, as explained in Section 3.6.1 page 67.

- Number of years of simulation: it is denoted *iter* and must be a positive integer.

- Minimum age of sexual maturity: it is denoted *min_age_sm* and must be a positive integer.

- Mean age of sexual maturity: it is denoted *mean_age_sm* and must be a positive integer.

- Age of menopause: it is denoted *meno_age* and must be a positive integer.

- Calving interval: it is denoted *mean_ci* and must be real, positive and greater than Pregnancy length + Anoestrous Lactation length.

- Pregnancy length: it is denoted *mxpreg* and must be a positive integer[1].

- Anoestrous Lactation length: it is denoted *mxlact* and must be a positive integer[2]. This is the time during which the female can not conceive after giving birth to the new calf. It sets the minimum calving interval to Pregnancy length + Anoestrous Lactation length. Moreover, calves whose mothers die during this period can die too in the Age and state model, according to the option chosen.

[1] advised value : 2. I put this in a parameter so that the model can be used for other grazers with different pregnancy lengths.

[2] advised value : 2 or 3 or 4.

- Sex ratio at birth: it is first expressed as the number of females for 1 male. It is denoted *sex_ratio_fem* and must be a positive real number.
- Probabilities of surviving in 7 different intervals: the age categories have been chosen after Bruce Page and and the probabilities correspond to the p_x used in Section 2.2.1. This parameter is denoted *probs* and must be a vector (matlab format) containing 7 real numbers between 0 and 1.

Parameters (park)

It is possible to set the carrying capacity of the park or to let it be calculated by the program, as explained in Section 1.6 page 15. In this case, the user has to provide :

- The area of the park (in km^2), denoted *area.* It must be a positive real number.
- The annual rainfall (in mm), denoted *rainfall.* It must be a positive real number. If the user does not know the soil types, this parameters should not exceed 700, otherwise the equations used to calculate the carrying capacity are no more accurate.
- The ratio "high nutrient soils:low nutrient soils". It is expressed as the number of low nutrient soils area units for one area unit of high nutrient soils. This number must be a positive real number. If this ratio is unknown, the program will use equation (1.11) in the calculus of the carrying capacity.

Options

It is possible to choose different options in order to plot different models: the exponential model in continuous time, the s-shape model in continuous time, the Leslie model and the Age and State model.

If the Age and State model is chosen, the program needs some more information. The user has to give the number of repetitions (denoted *nrep*) to run the program in order to calculate means and standard deviations. This number must be a positive integer[3].

The option "kill orphan calves" can be chosen by the user. In this case, the calves whose mothers were still lactating and died will also die the same year.

There is also two different options to include environmental conditions and density dependence in the age and state model, as explained in Chapter 2 Section 2.3.4 page 36. These options are: "Varying parameters according

[3] 50 or 100 seem sufficient to estimate a good mean in a reasonable time, but depending on the computer power, the user can try different values.

to environmental conditions contained in file..." and "Varying parameters according to environmental conditions and carrying capacity". In both cases, the user must specify the name of the file containing the environmental conditions (this text file must be placed in the main folder and contains real numbers between -1 and 1. This number must be space or tab delimited and the user can build this file using the excel file NewEnvironment.xls as explained in Section 3.6.2 page 67).

It is advised to use the two options "kill orphan calves" and "Varying parameters according to environmental conditions and carrying capacity" because they make the model more realistic. But if these options are not selected, it is possible to check that we obtain results very similar to the Leslie Matrix model.

To finish, it is possible to select the option "Age distribution movie" which permits to see the evolution, year by year, of the age distribution for the whole population. The stable age distribution (as explained in Section 2.2.4 page 31) is plotted on the same graph in another color so that it is possible to see how the age distribution evolves to the stable age distribution for each discrete model. A parameter giving the number of years per second can also be chosen.

Actions

By clicking on four buttons, you can choose four different actions: "start the simulation", "save the interface" (in order to use the same parameter the next time the program will be run), "close the figures" and "close the interface". To close the program properly, it is better to close the figures and then the interface with these buttons.

3.3.2 The program: simulele.m

The file contains two types of function:

1. Opening function: executes before the GUI becomes visible to the user.
2. Callbacks: execute each time the user activates the corresponding component of the GUI.

It is not interesting to describe the opening function because it is automatically created by guide. However, it can be interesting to understand how the callbacks work. Here is the structure of the main callback run when the user presses the button "start the simulation", which calls the callback associated to the tag *go*. Some pieces of the code are shown.

```
% --- Executes on button press in go.
function go_Callback(hObject, eventdata, handles)
% hObject    handle to go (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Collecting data contained in the graphical user interface

The first step is to collect all data entered by the user. Normally, all the properties of the objects contained in the interface are in a structure of handles called by default *handles*. But I had some problems with it (object sometimes disappeared from the structure) so I had to find a workaround. That is why I chose to use the function guihandles to collect the handles of the interface (called *figure1*) in each callback :

```
han=guihandles(handles.figure1);
```

If we want to access the handle of the object recognized with the tag *Name_file_input*, we just have to type *han.Name_file_input*.

The function *V = GET(H,'PropertyName')* returns the value of the specified property for the graphics object with handle H. For example, if the property that interests us is a string, and we want to put it in the variable *name*, we have to type:

```
name = get(han.Name_file_input,'String');
```

If the string is a number, we have to evaluate it before putting it in a variable:

```
mean_ci=eval(get(han.mean_ci_input,'String'));
```

This has to be done for all the parameters concerning the population. To get the values of the Radio Buttons (1 if the button is selected, 0 if not) or the Pop-Up Menus (the value contains the index of the selected item), we will use:

```
know=get(han.know_input,'value'); %=1 if the user knows
                                  %the carrying capacity
```

Reading the initial population and the environmental conditions

We use the function *fscanf* for this purpose. Here is an example for the initial population.

```
fid = fopen(name);
%We build the format
formatpop='';
for i=1:3+mxpreg+mxlact
    formatpop=strcat([formatpop,' %e ']);
end
```

```
popinit=fscanf(fid,formatpop,[3+mxpreg+mxlact inf]);
%now the initial population is contained in the matrix popinit
fclose(fid);
popinit=popinit';
mxage=size(popinit,1);  %maximum age
```

The format expressed as "%e" is very flexible so that it can read integers or real numbers of any length, delimited by any spaces or tabulations, that is why there is no constraint for the user when creating the text file. We use exactly the same code for reading the environmental conditions (if necessary), except that the format is the one of a vector and not of a matrix. We do not need the environmental conditions for the models run under Matlab but we need them for the Age and State model used under the fortran program ModelingEle7.exe. So we can write the input file for ModelingEle7.exe as follows:

```
fid=fopen('Fortran\conditions.txt','w');
formatcond=' ';
for i=1:length(cond)
   formatcond=strcat(formatcond,'%7.4f');
end
formatcond=strcat(formatcond,'\n');
fprintf(fid,formatcond,cond');
fclose(fid);
```

The advantage of doing so is that we now know the exact format of the file condition.txt and we can use the same in the fortran program[4]. If the fortran program read the initial population directly, the user would have much more constraints for the format because there is no equivalent of "%e" in fortran.

Building the vectors for the Leslie Matrix model

We have to sum the columns of the population table containing the females in order to run the Leslie Matrix model.

```
% x_fem(i) : population of female elephants at time t=0
x_fem=sum(popinit(1:mxage,2:3+mxpreg+mxlact),2);
x0_fem=x_fem;
N0_fem=sum(x_fem); %initial number of females
% x_mal(i) : population of male elephants at time t=0
x_mal=popinit(1:mxage,1);
N0_mal=sum(x_mal); %initial male population
N0_tot=N0_mal+N0_fem; %initial total population
```

[4] %7.4f is a floating number with width of field 7 and 4 digits following the decimal point.

Calculating fecundity

In Section 2.2.1, we can see that in order to construct the Leslie matrix in the Leslie model, we need the different probabilities of surviving denoted p_x and the fecundities at age x denoted b_x. But the user never give b_x. What he gives are the minimum age of sexual maturity, the mean age of sexual maturity and the mean calving interval. It is easy to deduce the fecundity if we notice that the discrete distributions which are behind each parameter (calving interval, age of sexual maturity) are geometric distributions and if we use conditional probabilities.

Conditional probability: The probability that an event A occurs on condition of an event B occurs is called a conditional probability and is denoted $P(A/B)$. It is equal to $P(A \cap B)/P(B)$ ($P(A \cap B$ is the probability that events A and B occur). So we obtain, multiplying each side by $P(B)$:

$$P(A \cap B) = P(A/B) \times P(B) \tag{3.1}$$

For example, we have

$$P(pregnant \cap mature) = P(pregnant/mature) \times P(mature) \tag{3.2}$$

But $P(pregnant \cap mature)$ is precisely the fecundity b_x. We now have to calculate $P(pregnant/mature)$ (that we will assume independent of the age x) and $P(mature)$ (which depends obviously on the age x).

Geometric distribution: As a reminder, the geometric distribution is a discrete probability distribution that describes the probability that a Bernoulli process (of parameter p) will have its first success on the n^{th} trial. Thus we have

$$P(X = n) = (1 - p)^{n-1}p \tag{3.3}$$

The mean of such a distribution is $1/p$.

Here the Bernoulli process can be the sequence of independent random variables Y_1, Y_2, Y_3,..., such as $Y_i = 1$ if the mature female is pregnant during the i^{th} year after being mature (or after being pregnant for the last time), $Y_i = 0$ otherwise. The parameter for this bernoulli process is $P(pregnant/mature)$ the probability of being pregnant on condition of being mature. We can see that the mean calving interval is the mean of this geometric distribution so $CI = 1/P(pregnant/mature)$ and we have $P(pregnant/mature) = 1/CI$.

But we can also consider another geometric distribution where the Bernoulli process is the sequence of independent random variables Z_1, Z_2, Z_3,..., such as $Z_i = 1$ if the immature female is being mature during the i^{th} year after the minimum age of sexual maturity, $Z_i = 0$ otherwise. Now the parameter for this Bernoulli process can be denoted $prmat$, and is equal to $1/(mean\_age\_sm - min\_age\_sm + 1)$, so that, for all $x \geq min\_age\_sm$:

$$\begin{aligned} P(\text{mature at age x}) &= P(X = x - min\_age\_sm + 1) \\ &= (1 - prmat)^{x - min\_age\_sm} \times prmat \end{aligned}$$

Finally we have our nice formula for b_x:

$$b_x = \frac{1}{CI} \times (1 - prmat)^{x - min\_age\_sm} \times prmat \qquad (3.4)$$

b_{fx} and b_{mx} are then calculated multiplying b_x by the sex ratios at birth, as we can see in the following code:

```
prmat=1/(mean_age_sm-min_age_sm+1); %probability of being
                                    %mature during the current year
K=[min_age_sm:mxage];
prmature=zeros(1,mxage);  %prmature(i) : probability of being mature at age i
prmature(min_age_sm:mxage)=cumsum(prmat*(1-prmat).^(K-min_age_sm));

cfecundity=1/mean_ci; %Conditional fecundity : probability of being
                      %fecund knowing that mature
cfec=zeros(1,mxage);  %cfec(i) : conditional probability : Fecundity
                      %knowing that the female is  mature at age i
cfec(min_age_sm:meno_age)=cfecundity;

fec=cfec.*prmature; % fec(i) : fecundity at age i
fec_fem=fec*sex_ratio_fem;
fec_mal=fec*(1-sex_ratio_fem);
```

Building the vector with all the probabilities of surviving

```
% prsur(i) : survival rate at age i
prsur=zeros(1,mxage);
prsur(1)=probs(1);
prsur(2:5)=probs(2);
prsur(6:10)=probs(3);
prsur(11:15)=probs(4);
prsur(16:45)=probs(5);
prsur(46:55)=probs(6);
prsur(56:mxage-1)=probs(7);
prsur(mxage)=0;

lx=cumprod(prsur); %survivorship
```

Building the Leslie Matrix and main loop

We can now create the Leslie Matrix and calculate the largest magnitude eigenvalue and the right eigenvector associated to it (equation (2.25) page 30). The stable age distribution vector (see page 31) is then calculated normalizing it as follows:

```
L=diag(prsur(1:mxage-1),-1);
L(1,:)=fec_fem.*prsur;
%largest magnitude eigenvalue of Leslie Matrix
lambda_1=max(abs(eig(L)));
%Right eigenvector associated to lambda_1
power=[1:mxage-1];
u_1=[1,(lambda_1.^(-power)).*lx(1:mxage-1)];
stableagedist=u_1/sum(u_1);  %stable age distribution vector
```

We can then enter the loop on the years:

```
for i=1:iter
    %males
    x_mal=[x_fem'.*prsur*fec_mal';x_mal(1:mxage-1).*prsur(1:mxage-1)'];
    N_mal(i)=sum(x_mal);

    %females
    x_fem=L*x_fem;
    N_fem(i)=sum(x_fem);

    %if the option Age distribution movie
    %has been chosen, we create the frames here
    ...

end
```

Calculating carrying capacity

If the user does not know the carrying capacity of the park, we have to calculate it for him. We use the formulas displayed in Section 1.6 page 15. We make use of the stable age distribution to calculate the elephant mean weight.

```
if menu_soils==1 %if the user know the nutrient ratio
    if rainfall<=750
    biomass=0.6*(
        nutrient_ratio*exp((-2+2.03*log10(rainfall))*log(10))...
        +(1-nutrient_ratio)*exp((-2.5+2.10*log10(rainfall))*log(10))...
               )*area;
    else
    biomass=0.6*(
        nutrient_ratio*exp((-2+2.03*log10(rainfall))*log(10))...
        +(1-nutrient_ratio)*exp((9.5-2.10*log10(rainfall))*log(10))...
               )*area;
    end
```

```
elseif menu_soils==2 %if the user does not know the nutrient ratio
        biomass=0.6*(8.684*rainfall-1205.9)*area;
end
ele_male_weight=sum(...
  stableagedist(1:20).*3112.*(1-exp(-0.114*([1:20]+3.95))).^3)+...
  sum(stableagedist(21:mxage).*4742.*(1-exp(-0.166*([21:mxage]-10.48))).^3);
ele_female_weight=sum(...
  stableagedist(1:mxage).*2744.*(1-exp(-0.099*([1:mxage]+6.0))).^3);
ele_weight=ele_male_weight*(1-sex_ratio_fem)+ele_female_weight*sex_ratio_fem;
carrying=round(biomass/ele_weight);
```

Running the fortran program ModelingEle7.exe

In order to run the fortran program which uses the Age and State model, we need to build the input data file containing the parameters and the population table. Once it has been run, we still have to read the results in order to plot them later.

```
%We build the input data file
fid=fopen(’Fortran\data.txt’,’w’);

fprintf(fid,’%4u%4u%4u%4u%4u%4u%4u%4u%4u%4u%4u\n’,nrep,iter,mxage,mxpreg,...
               mxlact,min_age_sm,mean_age_sm,meno_age,carrying,option,5);
fprintf(fid,’%7.4f%7.4f%7.4f%7.4f%7.4f%7.4f%7.4f\n’,probs);
fprintf(fid,’%8.5f%8.5f\n’,mean_ci,sex_ratio_fem);
formatpop=’ ’;
for i=1:size(popinit,2)
   formatpop=strcat(formatpop,’%2u’);
end
formatpop=strcat(formatpop,’\n’);
fprintf(fid,formatpop,popinit’);

fclose(fid);

%We run fortran program
!Fortran\ModelingEle7.exe

%We read results
fid = fopen(’Fortran\agetot.txt’);
formatscan=’’;
for i=1:iter
   formatscan=strcat([formatscan,’ %e ’]);
end
agetot=fscanf(fid,formatscan,[iter mxage]);
```

```
fclose(fid);
agetot=agetot';
```

Outputs

Graphical outputs: It is very easy to plot graphs under Matlab and it is not very useful to display the code for that. The only difficulty is that the user can choose to run the models he wants so the legend has to be adapted. We do so using a string *legendstr* containing the legend that we update each time a model is plotted. Figure 3.3 contains the graphical outputs that we obtain running the program with the parameters and options seen in Figure 3.2 page 42:

1. The first graph shows the environmental conditions (in that case it is a cosinus of amplitude 0.8 and of period 20).
2. The second graph describes the probability of being mature for a female depending on its age. This graph must help the user to see if he is satisfied with the parameters he entered (minimum age of sexual maturity and mean age of sexual maturity).
3. The third graph gives the fecundity and mortality of elephants depending on their ages.
4. The fourth graph shows the population growth of female elephants (total number of females for each year of the simulation).
5. The fifth graph shows the population growth of male elephants (total number of males for each year of the simulation).
6. The last graph shows the population growth of the whole population of elephants.

Figure 3.4 displays some frames of the age distribution movie (percentage of the elephant population in each age for each year and each model).

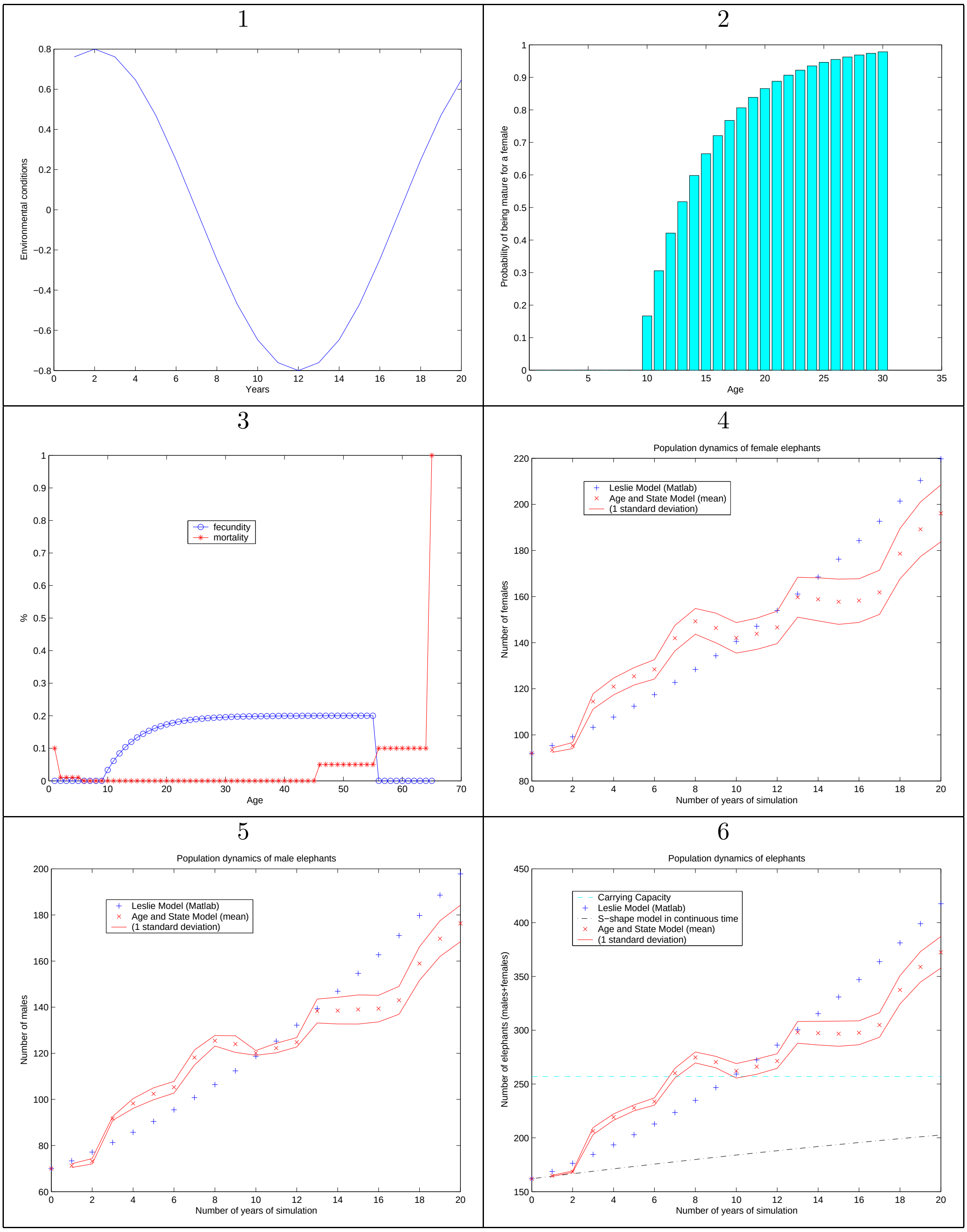


Figure 3.3: Graphical outputs

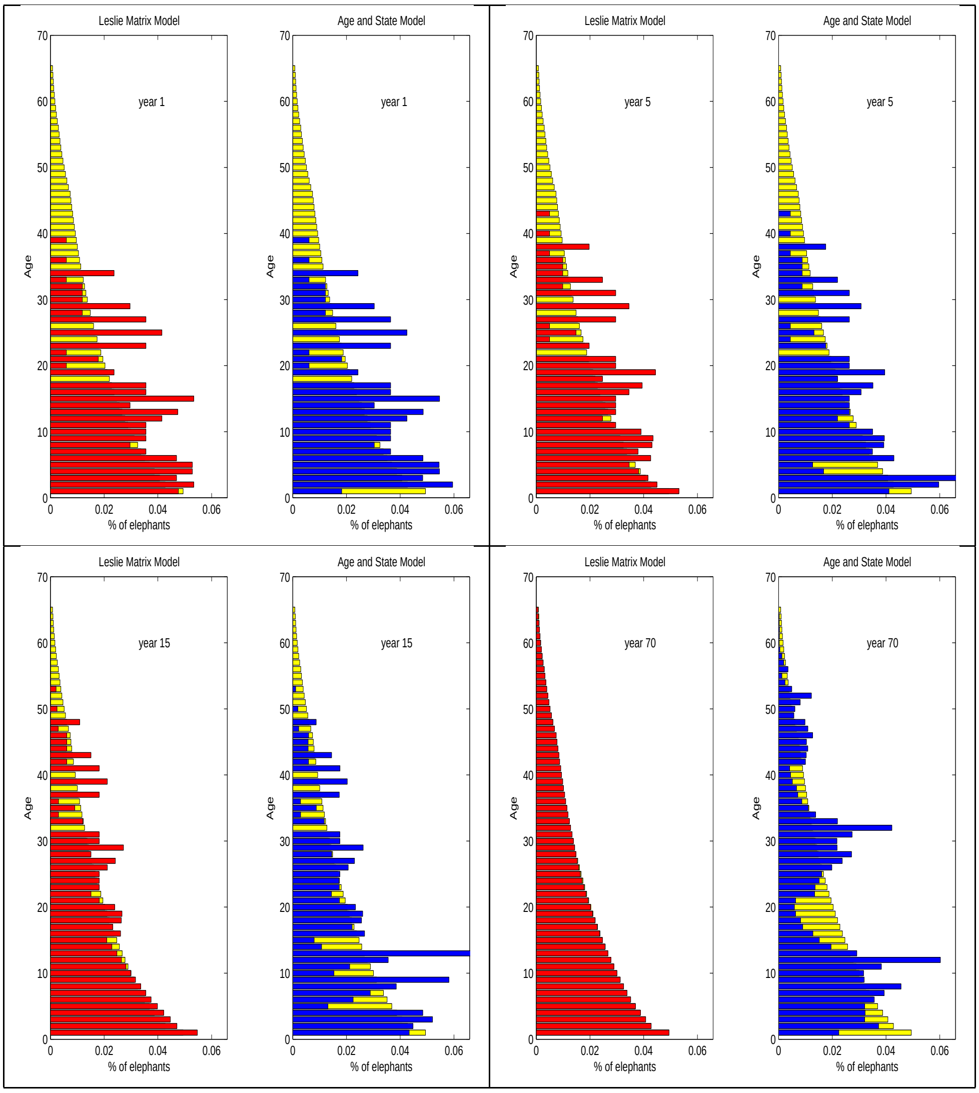


Figure 3.4: Some frames of the Age Distribution Movie

It was possible for me to write numerical outputs on the DOS window which is launched with the program simulele.exe. But this window does not have a scroll bar and so it is impossible to write too much information on it. So I chose to write information concerning stationary population in the DOS window and all the other information in a text file named LifeTable.txt and placed in the main directory.

Text file LifeTable.txt: In the file LifeTable.txt, it is possible to find the life table of the population as in table 2.2.1 page 27.

Then all the immediate information of Section 2.2.2 page 26 are given: the net reproductive rate per generation, the mean generation time, theoretical rate of natural increase, and the theoretical annual growth rate.

I also give better estimates for the rate of natural increase and for the annual growth rate. One estimation is given using the total number of elephants obtained in the last year of simulation ($N(iter)$) and the formula:

$$r = \frac{ln(N(iter)) - ln(N(1))}{iter - 1} \tag{3.5}$$

The best estimation[5] is based on equations (2.29) and (2.30) page 31. To find the eigenvalue λ_1, we use the function “eig” from Matlab (which uses LAPACK routines):

```
lambda_1=max(abs(eig(L)));
```

Since the Leslie matrix is a square sparse large matrix, it would have been better to use the function *eigs* from Matlab, but this function is not supported by the compiler to build a stand alone application.

```
fid=fopen('LifeTable.txt','w');

fprintf(fid,'%s\n','Life Table');
fprintf(fid,'%s\n','----------');
fprintf(fid,'%s\n','   x     px      lx      mx');
y=[[1:mxage];prsur;lx;fec];
fprintf(fid,'%4d %7.3f %7.3f %7.3f\n',y);

% IMMEDIATE INFORMATION
fprintf(fid,'%s\n',' ');
fprintf(fid,'%s\n','Immediate information');
fprintf(fid,'%s\n','---------------------');
R0=lx*fec_fem';
fprintf(fid,'%s %7.3f\n','Net reproductive rate per generation:',R0);
T=([1:mxage].*lx*fec_fem')/R0;
```

[5] This is this estimation of r that we use in the exponential model and in the s-shape model.

```
fprintf(fid,'%s %6.3f\n','Mean generation time :',T);
r=log(R0)/T;
fprintf(fid,'%s %6.3f\n','Theoretical rate of natural increase:',r);
g=(exp(r)-1)*100;
fprintf(fid,'%s %6.3f\n','Theoretical annual growth rate (%):',g);
fprintf(fid,'%s\n',' ');

fprintf(fid,'%s\n','Better estimates of r and g');
fprintf(fid,'%s\n','---------------------------');
r2=(log(N_tot(iter))-log(N_tot(1)))/(iter-1);
fprintf(fid,'%s %6.3f\n','Leslie Model rate of natural increase...
                  (mean): ',r2);
g2=(exp(r2)-1)*100;
fprintf(fid,'%s %6.3f\n','Leslie Model annual growth rate (%) (mean):'
                           ,g2);
fprintf(fid,'%s\n',' ');

fprintf(fid,'%s\n','Best estimates of r and g');
fprintf(fid,'%s\n','---------------------------');
fprintf(fid,'%s %6.3f\n','Largest magnitude eigenvalue of Leslie
                   Matrix',lambda_1);
r3=log(lambda_1);
fprintf(fid,'%s %6.3f\n',...
           'rate of natural increase based on the perron eigenvalue:'
   ,r3);
g3=100*(lambda_1-1);
fprintf(fid,'%s %6.3f\n','annual growth rate based on the perron
                  eigenvalue: ',g3);

fclose(fid);
```

DOS window: In the DOS window, I give two suggestions in order to obtain a stationary population. The first possibility is changing the survival rate for young calves (we can do this by removal or culling). In this case , I give the probability of surviving for a calf which is less than 1 year old so that the population is stationary. This value is easy to calculate if we use results from the Section 2.2.4 on the properties of the Leslie Matrix, page 30. We know that the population is stationary if and only if $\lambda_1 = 1$. But λ_1 satisfies the equation (2.23) and so we have:

$$1 = \sum_{x=1}^{k} l_x b_{fx} = \sum_{x=1}^{k} (b_{fx} \prod_{i=1}^{x} p_i) \tag{3.6}$$

What we want to display[6] is p_1, and it is equal to:

$$p_1 = \frac{1}{b_{f1} + \sum_{x=2}^{k}(b_{fx}\prod_{i=2}^{x} p_i)} \tag{3.7}$$

We can also display the expected number of cows in the stationary population using the fundamental theorem of demography (equation (2.32) page 32 and equations (2.25) and (2.27)) to calculate the right and left eigenvectors.

The second possibility is scaling the calving interval (by utilizing contraception). In fact, we can see with equation (2.23) that if we scale each b_{fx} by $R_0 = \sum_{x=1}^{k} l_x b_{fx}$, $\lambda_1 = 1$ is a root and so the population is stationary. Looking equation (3.4), we can see that dividing b_{fx} by R_0 is equivalent to multiplying the calving interval CI by R_0.

```
%survival rate for young elephants (in the first age category) for a
%stationary population
a=fec_fem(1)+fec_fem(2:mxage)*cumprod(prsur(2:mxage))';
p1=1/a;
disp('How To obtain a stationary Population in the Leslie Matrix model?')
disp('*****************************************************************')
disp('')
disp('first possibility : changing the survival rate for a young calf:')
disp('*******************')
disp('Probability of surviving for a calf which is less than 1 year old')
disp('so that the population is stationary :')
if p1<1
    p1
    disp('Number of cows in the stationary population:')
    %Right eigenvector associated to lambda_1 for the new Leslie Matrix
    power=[1:mxage-1];
    u_1=[1,cumprod([p1,prsur(2:mxage-1)])];
    %Left eigenvector associated to lambda_1 for the new Leslie Matrix
    v_1=zeros(1,mxage);
    v_1(1)=1;
    v_1(2)=(1-p1*fec_fem(1))/p1;
    for i=3:mxage
        v_1(i)=(v_1(i-1)-L(1,i-1))/prsur(i-1);
    end
    v_1=v_1/(v_1*u_1');

    disp(sum(v_1*x0_fem*u_1'));
```

[6] If $p_1 > 1$, we do not display p_1 and say that it is impossible to obtain a stationary population.

```
else
    disp('IMPOSSIBLE (p1>1)');
end

%scaling the calving interval
disp('');
disp('Second possibility: scaling the calving interval (contraception)');
disp('*******************');
disp('Calving interval so that the population is stationary:');
disp(mean_ci*R0);
```

3.3.3 Compilation: simulele.exe

It is not very usual to compile a Matlab code which is an interpreted language. So I am going to describe in a few words how to do it.

Why compiling the m-file into a stand-alone application is better

Because it is possible to create Matlab applications that take advantages of the mathematical functions of Matlab, yet do not require that the user owns Matlab. Moreover, compiled code typically runs faster than its M-file equivalents because compiled code usually runs faster than interpreted code. Especially if the code contains loops.

How to compile the m-file

In order to compile the m-file “simulele.m”, I used the Matlab compiler 3.0. The Matlab Compiler (mcc) automatically invokes mbuild to perform compilation and linking (see Figure 3.5 on next page).

To translate my M-file simulele.m that contains Handle Graphics functions into C and to create a stand-alone executable that can be run without Matlab, I used:

```
mcc -B sgl simulele.m
```

Packaging the MATLAB Run-Time libraries

All the Matlab run-time libraries required by stand-alone applications are prepackaged into a single, self-extracting archive file, called the Matlab Compiler Run-Time Library Installer. Instead of including all the run-time libraries individually in my stand-alone application distribution package, I simply included this archive file (mglInstaller.exe).

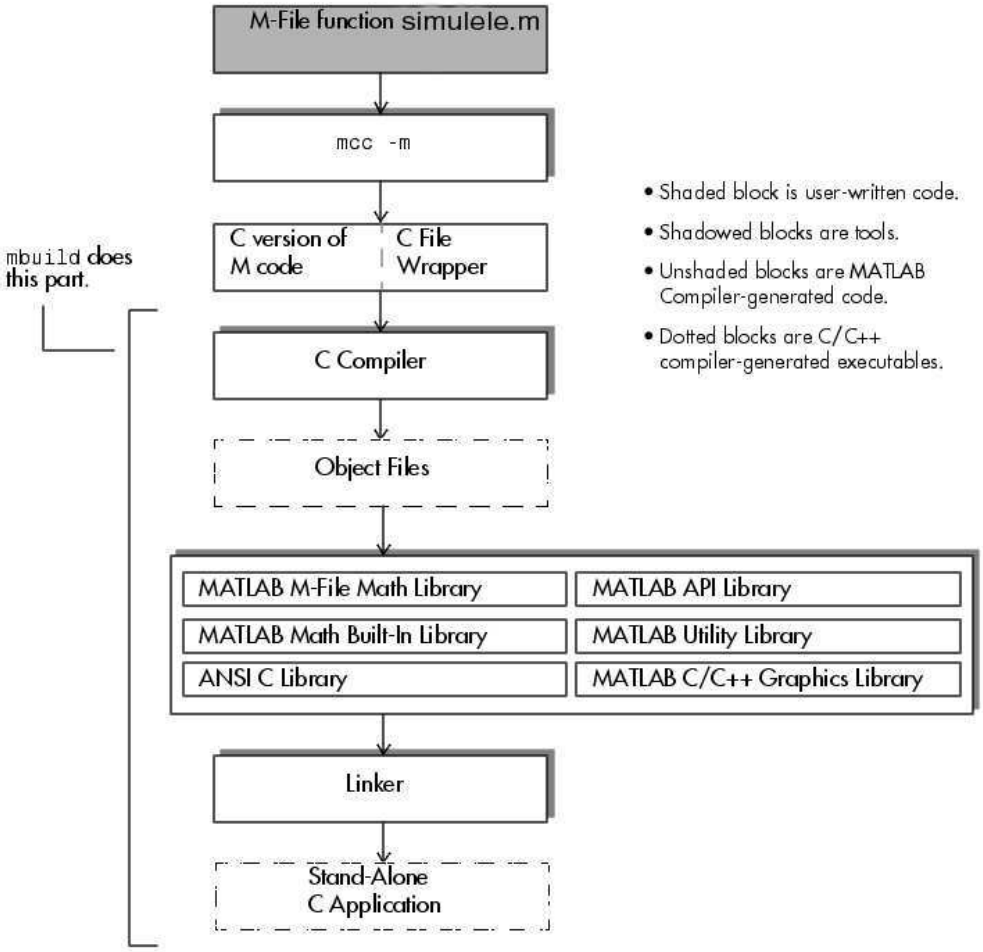

M-File function simulele.m
mcc -m
C version of M code
C File Wrapper
C Compiler
Object Files
MATLAB M-File Math Library
MATLAB API Library
MATLAB Math Built-In Library
MATLAB Utility Library
ANSI C Library
MATLAB C/C++ Graphics Library
Linker
Stand-Alone C Application
mbuild does this part.
• Shaded block is user-written code.
• Shadowed blocks are tools.
• Unshaded blocks are MATLAB Compiler-generated code.
• Dotted blocks are C/C++ compiler-generated executables.

Figure 3.5: Developing a Stand-Alone C Application

Installing the application

I created a file “readme.txt” where all the instructions to install the program were explained. Next frame displays the content of this file:

```
In order to use the Program simulele.exe, you have to:

*Run the MATLAB Compiler Run-Time Library Installer:
mglinstaller.exe.
This program extracts the libraries from the archive
and installs them in subdirectories of a directory
specified by you (By default, the installer puts the
files in the current directory).

*After the installer unpacks and uncompresses the
libraries, you must add the bin\win32 subdirectory to
the system path variable (PATH) (for this, you have to
edit the file C:\autoexec.bat).
For example, if you specify the installation directory
C:\mgl_runtime_dir, then you must add
"C:\mgl_runtime_dir\bin\win32" to PATH as follows:
SET PATH="%PATH%;C:\mgl_runtime_dir\bin\win32"

*Restart your computer.

*Then you can run the program simulele.exe by double
clicking on it.

Note: There is two files containing the initial
population of St Lucia (StLucia.txt) and Pilanesberg
(Pilanesberg.txt). To create another initial population,
you can use the file NewInitialPopulation.xls (you need
to have installed excel and to enable macros). Then you
can save the table in the appropriate text format
selecting in "saveas" the type "Text(Os/2 or MS-DOS)".
```

Optimization

The Matlab Compiler can perform various optimizations on your M-file source code that can make the performance of the generated C/C++ code much faster than the performance of the M-code in the Matlab interpreter. Matlab Compiler 3.0 provides a series of optimizations that can help speed the compiled code.

There are several possible optimizations:

- Optimizing Arrays: Improving the performance of code that manipulates scalar arrays
- Optimizing Loops: Improving the performance of simple one- and two-dimensional array index expressions
- Optimizing Conditionals: Reducing the Matlab conditional operators to scalar C conditional operators

To turn on all optimizations, I used the option -O all. Finally, the compiling command I typed is:

```
mcc -B sgl -O all simulele.m
```

Limitations and restrictions

Matlab Compiler 3.0 supports almost all of the functionality of Matlab. However, there are some limitations and restrictions. This version of the Compiler cannot compile:

- Script M-files
- M-files that use some functions that are not supported in stand-alone mode (the detailed list is in the Matlab help)
- M-files that use “input” or “eval” to manipulate workspace variables
- M-files that dynamically name variables to be loaded or saved
- M-files that load text files (for example: load -ascii popinit.txt).

The only limitations which are annoying for us are the second and the last point.

A workaround for the last point is to use the fscanf function of Matlab as explained in the last section.

For the second point, we have to be careful. A function such as “eigs” which calculates a few eigenvalues and eigenvectors of a square large sparse matrix is not supported and we have to find other solutions (see page 55).

3.4 Programming with Fortran (Age and State model)

3.4.1 What has been improved

As I have already mentioned, the code for the age and state model was originally created by Professor Botkin some years ago. I updated it so that the program ModelingEle7.exe is called by the main program simulele.exe. All the data are exchanged by text files as explained in the next section (Section 3.5).

I am not going to display the code here but the interested reader can find it in Appendix B.

I updated the program from Fortran77 to Fortran90. The Arrays are now allocatable, so that the size is not fixed before the execution of the program. It permits the user to change the maximum age and to fix the pregnancy length and the anoestrous lactation length.

I changed the input in order to adapt it to the program simulele. Now the fecundity is calculated according to the calving interval, the pregnancy length and the anoestrous lactation length. The output was also changed to add more details.

The subroutine which I modified the most is **ENVIRN**. It now includes environmental condition and density dependence.

I also added one function **INTERPOL** to interpolate linearly between best parameters and user parameters or between user parameters and worst parameters, depending on the environmental conditions.

Finally, I added some comments to make the code more readable.

3.4.2 Input

The parameters and the initial population are read with the subroutines **GETPAR** and **GETPOP** in the file “data.txt”. The parameters in this file should be entered in this order:

1. *PAR(i)* is the one dimensional array containing the demographic parameters :

   PAR(1) = number of repetitions of the iteration period (denoted *REP*)
   PAR(2) = number of iterations in each repetition (denoted *ITER*)
   PAR(3) = maximum age (in periods of iteration) of the species (denoted *MXAGE*)
   PAR(4) = length of the gestation period in periods of iteration (denoted *MXPREG*)
   PAR(5) = length of the lactation period in periods of iteration (denoted *MXLACT*)
   PAR(6) = minimum age of sexual maturity (denoted *MINMAT*)
   PAR(7) = mean age of sexual maturity (denoted *MEANMAT*)
   PAR(8) = age of menopause (denoted *MENOAGE*)
   PAR(9) = carrying capacity of the park (denoted *CARRYING*)
   PAR(10) = option selected for the model (denoted *OPTION*):

   0 No options

   1 We kill orphan elephants whose mothers were lactating

   2 Parameters vary linearly depending on values of the environment

   3 Parameters vary linearly depending on values of the environment and the carrying capacity

4 We kill orphan calves and parameters vary linearly depending on values of the environment

5 We kill orphan calves and parameters vary linearly depending on values of the environment and the carrying capacity

PAR(11) = output flag *F1*

2. *PROBS(i)* are the probabilities of surviving in 7 different age categories: between 0 and 1, 2 and 5, 6 and 10, 11 and 15, 16 and 45, 46 and 55, 56 and *MXAGE*

3. *CALVIN* : calving interval
   SEXRATIO : the sex ratio at birth (percentage of females)

4. *POP(i,j)* is the two dimensional array (called population table) which tracks the numbers of individuals in each age (rows), each functional group (columns).

   POP(i,1) = males

   POP(i,2) = immature females

   POP(i,3) = non-pregnant, mature females

   POP(i,4...3+MXPREG) = pregnant females in the j^{th} "iteration interval" of pregnancy

   POP(i,4+MXPREG...3+MXPREG+MXLACT) = lactating females in the j^{th} "iteration interval" of lactation

   POP(i,4+MXPREG+MXLACT) = total population at age i (all functional groups)

   POP(MXAGE+1,j) = total population in class j for all ages

   POP(MXAGE+1,6+MXPREG+MXLACT) = total size of population (all ages all functional groups)

The environmental conditions are read in file “conditions.txt”.

3.4.3 Loops

Then we have two imbricated loops, the main one on the repetitions in order to do statistics, the other one on years to update the array *POP* every year of the simulations.

In the loop on years, we call each of these subroutines:

ENVIRN This subroutine sets various probability values for survival, maturity and conception based on the environment and the current population. This subroutine calls the function **INTERPOL** to interpolate linearly between best parameters and user parameters or between user parameters and worst parameters, depending on the environmental conditions. Best and worst parameters are set as constants in the program (Table 3.1) whereas the user parameters are read in the input file “data.txt”.

	Worst Parameter	Best Parameter
Probability of surviving		
between 0 and 1	0.5	1.0
between 2 and 5	0.7	1.0
between 6 and 10	0.9	1.0
between 11 and 15	0.95	1.0
between 16 and 45	0.95	1.0
between 46 and 55	0.6	1.0
between 56 and *MXAGE*	0.0	1.0
Minimum age of sexual maturity	14	7
Mean age of sexual maturity	20	8
Age of menopause	45	55
Calving interval	20	*MXPREG+MXLACT*

Table 3.1: Best and worst parameters used in the Age and State model

SURVIV This subroutine selects the elephants which will survive to the next year. This subroutine calls (if the option is chosen) the subroutine **SURSUB** to kill the orphan calves whose mothers were lactating.

AGE This subroutine increases the age of each elephant which survived.

MATURE This subroutine causes the immature members to become mature, based on the probability contained in the array *PRMAT* (provided by the subroutine ENVIRN).

BIRTH This subroutine causes several population changes to occur: new members are born; once pregnant, females become lactating females; once lactating, females become receptive members.

CONCEP This subroutine causes conception of non pregnant females according to the probabilities contained in the array *PRCON* (provided by the subroutine EVIRN).

STATS This subroutine only calculates totals and stores the results that will be needed for statistics in the array *TOTALS*.

OUTPT This subroutine writes program results in results.txt.

3.4.4 Output

In the file results.txt, the user can find the number of elephants in each functional groups for each iteration and repetition. Here is an example of the results of a simulation (we can read the first four years of the first repetition) :

```
(IREP,ICNT) MAL IMM NPG  PG     LAC     TOT
 -------------------------------------------
(   1,   1)  72  40  22  21   4   3   1 163
(   1,   2)  75  34  15  15  21   4   3 167
(   1,   3)  82  42  14  10  15  21   4 188
(   1,   4)  88  44  12   9  10  15  21 199
```

During the third year of the simulation, there were 82 males, 42 immature females, 14 non-pregnant females, 10 females in the first year of pregnancy, 15 females in the second year of pregnancy, 21 females in the first year of anoestrous lactation, 4 females in the second year of anoestrous lactation (so the total number of elephants is 188).

At the end of the main loop on the repetitions, we call the subroutine **REPORT** which calculates means and standard deviations (and create files agetot.txt, meanfem.txt, meantot.txt, stdfem.txt and stdtot.txt for simulele.exe). It also produces a final report in result.txt with the means, the standard deviations and the final population table.

Figure 3.6 on next page summarizes the content of this section.

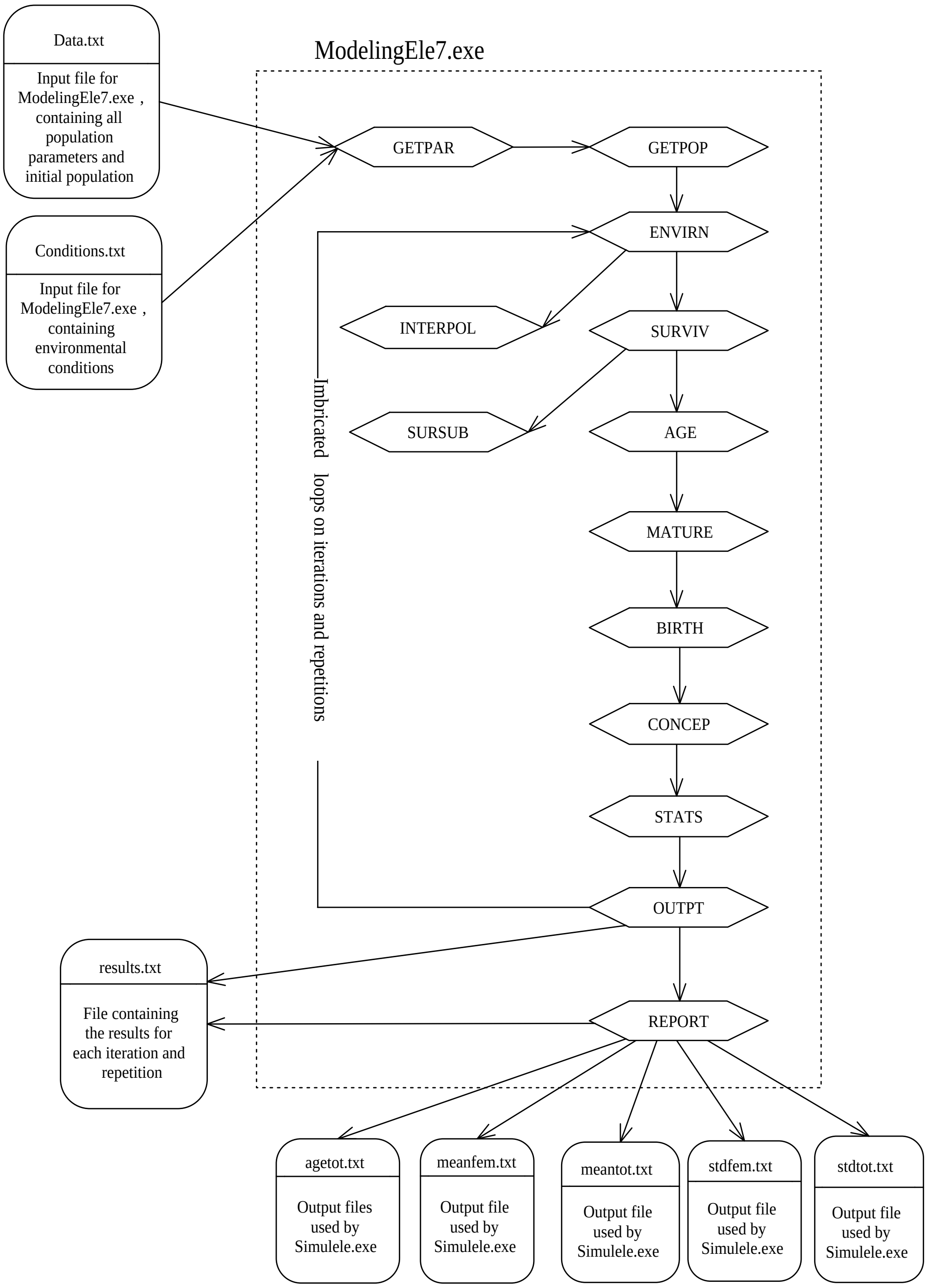

Data.txt
Input file for ModelingEle7.exe , containing all population parameters and initial population
Conditions.txt
Input file for ModelingEle7.exe , containing environmental conditions
ModelingEle7.exe
GETPAR
GETPOP
ENVIRN
INTERPOL
SURVIV
SURSUB
AGE
MATURE
BIRTH
CONCEP
STATS
OUTPT
REPORT
Imbricated loops on iterations and repetitions
results.txt
File containing the results for each iteration and repetition
agetot.txt
Output files used by Simulele.exe
meanfem.txt
Output file used by Simulele.exe
meantot.txt
Output file used by Simulele.exe
stdfem.txt
Output file used by Simulele.exe
stdtot.txt
Output file used by Simulele.exe

Figure 3.6: Fortran Program (Input, Output and Subroutines)

3.5 Files organization

All the files which are necessary to run the program simulele.exe are zipped into a single file named simulele.zip.

The Figure 3.7 on the next page shows how all the files (figures, text files, libraries and executable files) are organized into the main folder (the main folder is the folder where the zip file simulele.zip is unzipped).

3.6 Some tips for the users

3.6.1 Building a new initial population

If you installed Excel on your computer, it is possible to use the excel file NewInitialPopulation.xls in order to build a new initial population[7]. Instead of typing all the spaces and zeros, you just have to enter the maximum age, the number of years of pregnancy and the number of years of anoestrous lactation and the population table will be filled automatically with zeros. Then you can type the number of elephants in each age and state. To save the file, click on “Save As...” in the menu, indicate the main folder and choose an appropriate file name. The file format should be chosen as “Text(OS/2 or MS-DOS)”. It is also advised to save the file in excel format in order to modify it later if necessary.

Figure 3.8 shows what the users see when they open file NewInitialPopulation.xls.

3.6.2 Building new environmental conditions

It is also possible to build a text file containing the environmental conditions using the excel file “NewEnvironment.xls”. Setting the maximum, minimum, period and delay, these conditions can be based on a cosinus function. In order to save the file to the correct format, the user must accomplish the same procedure as for the new population.

Figure 3.9 shows what the users see when they open file NewEnvironment.xls.

3.6.3 Printing graphs

In order to print a graph generated by simulele.exe, you have to click on the menu figure “File/Save As...” and then choose the name of the figure. You can choose among several extensions which will determine the file format. The most famous formats are “.bmp” and “.jpg”. The second format is compressed so it is lighter, but you loose a little bit in quality.

[7] You must not forget to change the initial population file if you change the anoestrous length.

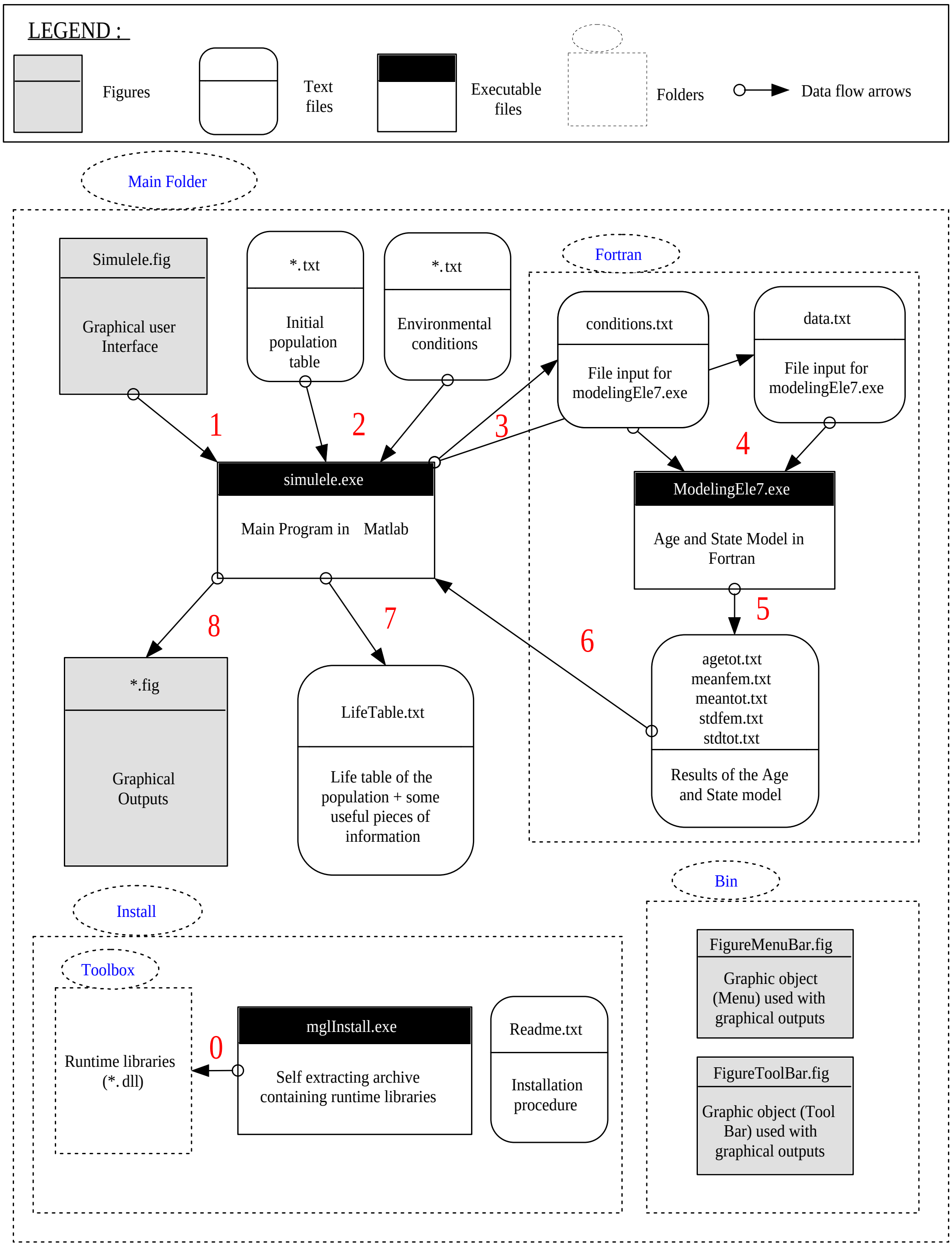
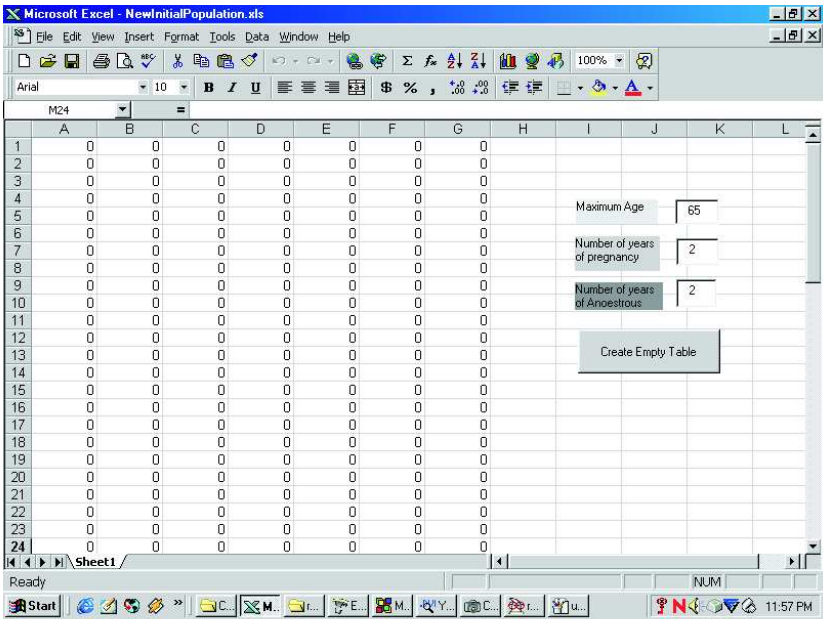


Figure 3.7: Files organization


Microsoft Excel - NewInitialPopulation.xls
File Edit View Insert Format Tools Data Window Help
Arial
10
100%
M24
Sheet1
Ready
NUM
Start
11:57 PM
Create Empty Table

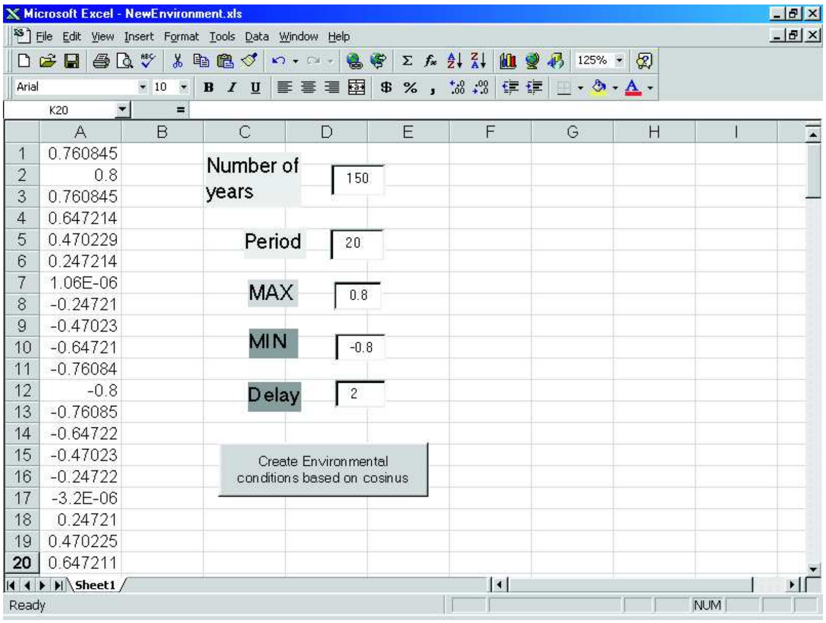
	A	B	C	D	E	F	G
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0

Maximum Age 65

Number of years of pregnancy 2

Number of years of Anoestrous 2

Figure 3.8: File NewInitialPopulation.xls


Microsoft Excel - NewEnvironment.xls
File Edit View Insert Format Tools Data Window Help
Arial
10
125%
K20
Sheet1
Ready
NUM
Create Environmental conditions based on cosinus

	A
1	0.760845
2	0.8
3	0.760845
4	0.647214
5	0.470229
6	0.247214
7	1.06E-06
8	-0.24721
9	-0.47023
10	-0.64721
11	-0.76084
12	-0.8
13	-0.76085
14	-0.64722
15	-0.47023
16	-0.24722
17	-3.2E-06
18	0.24721
19	0.470225
20	0.647211

Number of years 150

Period 20

MAX 0.8

MIN -0.8

Delay 2

Figure 3.9: File NewEnvironment.xls

3.6.4 Supplementation or removal

If the user wants to add or remove some elephants after x years of simulation, the easiest way is to start a simulation for x years, then copy the final population from the file results.txt (in the subfolder fortran) into another text file of the main folder, modify it to add or remove elephants and start running another simulation with this population table as the initial population table[8].

3.6.5 Reminder

Do not forget to save the interface before closing it so that you can recover the same parameters the next time you launch simulele.exe. It can also be interesting to use the tool bar in the figure to zoom in and out some parts of the graphs.

[8] It can be useful to do the same process in order to change the parameters (survival rate, age of sexual maturity,...) for example.

Chapter 4

Example of application: The St Lucia Population

This chapter displays an example of application of the program simulele.exe in order to show how it works. Because this small population has been introduced by managers, we know exactly the number of elephants (thirty one), their sexes and ages. It is therefore very easy to build the initial population file which is very precise.

4.1 Introduction

4.1.1 Presentation of St Lucia

St. Lucia is one of the oldest Game Reserves in Africa and is situated on the East Coast of South Africa. The Greater St. Lucia Wetland Park is a saline system consisting of five individual ecosystems. These ecosystems function totally independent yet fully integrated with each other. This system was proclaimed a world heritage site in December 1999.

The Greater St. Lucia Wetland Park is situated about 275 km North of Durban, on the main N2 tourist road to Kruger park via Swaziland (see map page 13 key 625). Figure 4.1 shows the different parks of St Lucia and their land use.

4.1.2 The elephants in St Lucia

Elephants are a key component of a reserve system wishing to attract tourists, and as such, elephants were introduced to Eastern Shores as part of a programme of restocking animals that previously occurred there. The first introduction of elephants in this park dates from the 7 August 2001. Four elephants from Hluhluwe Umfolozi Game Reserve were brought onto the Eastern Shores. This family group consisted of an adult female, her calf and two sub adult females.

During the 11 September 2001, five elephants were introduced from Hluhluwe: two bulls, an adult female with her young calf as well as a sub adult female. Because the two bulls broke the fences and escaped, both bulls were recaptured and brought back to Hluhluwe.

Between the 22 and 24 September, a further fourteen elephants were brought into the Eastern shores. During October, all elephants combined forming one large herd of twenty one elephants. One calf died the 29 October, probably because of the stress as a result of relocation. A new calf was born on Western shores during April 2002, bringing the total number of elephants to twenty one.

Finally, a new family group was relocated from the Kruger National Park on the 13 June 2002. To that date, the total number of elephants was therefore thirty one. Since June 2002 to december 2002 this number did not change. Table 4.1 references all the elephants at the beginning of 2003. Some ages are known, some have been derived from the shoulder-height. All data come from the Chantal Dickson's report.

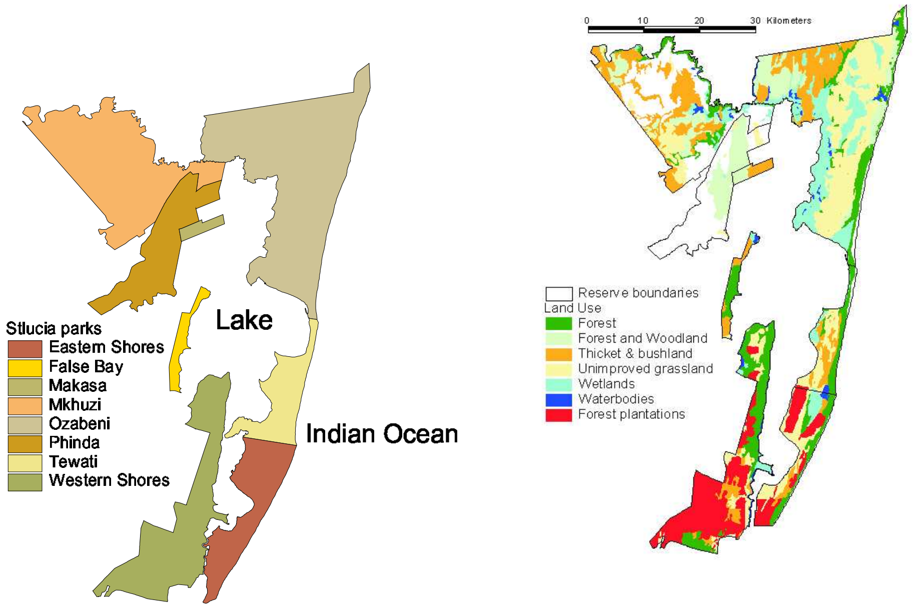


Figure 4.1: St Lucia Parks

Ref. No	Date of introduction	Sex	Age
From Hluhluwe			
ESF1	7/8/01	F	16
ESF2	7/8/01	F	10
ESF4	7/8/01	F	10
ESM5	11/9/01	M	2
ESF6	11/9/01	F	19
ESF7	11/9/01	F	7
ESM8	20/9/01	M	17
ESF9	20/9/01	F	15
ESM10	20/9/01	M	2
ESF11	20/9/01	F	19
ESF12	21/9/01	F	17
ESF13	21/9/01	F	19
ESF14	21/9/01	F	2
ESM15	21/9/01	M	17
ESF16	21/9/01	F	21
ESF17	22/9/01	F	9
ESM18	22/9/01	M	9
ESF19	22/9/01	F	18
ESF20	22/9/01	F	12
ESF21	22/9/01	F	21
?	(born in April 2002)	M	1
From Kruger National Park			
1	13/6/02	F	23
2	13/6/02	M	1
3	13/6/02	F	43
4	13/6/02	M	9
5	13/6/02	F	13
6	13/6/02	M	7
7	13/6/02	M	5
8	13/6/02	F	35
9	13/6/02	M	3
10	13/6/02	M	13

Table 4.1: St Lucia Initial population (2003)

4.2 Simulation

4.2.1 Park parameters

The fenced area in Saint Lucia is approximately 1075 km^2. The approximate rainfalls range between 820 mm and 1270 mm per year, say 1000 mm in average. An estimate for the ratio “high nutrient soils:low nutrient soils” is 1:5. We can imagine that the carrying capacity of such a park is huge and it is not going to be reached before a long time. Indeed, the calculated carrying capacity with these parameters is around 1070 elephants.

4.2.2 Population parameters

Here are the probabilities of surviving used for the population:

Age	0-1	2-5	6-10	11-15	16-45	46-55	56-65
Probability	0.90	0.95	0.99	1	1	0.95	0.90

Table 4.2: Probabilities of surviving used in the simulation

The other parameters are:

- Minimum age of sexual maturity: 10
- Mean age of sexual maturity: 13
- Age of menopause: 50
- Mean calving interval: 4
- Pregnancy length: 2
- Anoestrous lactation length: 2
- Sex ratio at birth: 1 male for 1 female

With these parameters, we obtain an annual growth rate based on the Leslie Matrix’ largest magnitude eigenvalue of 4.2%.

4.2.3 Graphs

Figure 4.3 shows the exponential model in continuous time. We can see that if all the parameters stay constant, the model predicts that the carrying capacity will be reached in approximately 86 years. Figure 4.4 shows the Leslie model and we can see that the prediction gives now 77 years. This comes from the fact that the initial population is very young and indeed the age distribution is not stable. So during the first decades, the population grows faster than expected with the exponential model. The growth rate with this model is 4.6%. Figure 4.5 shows the age and state model with no option chosen. The curve is very similar to the Leslie model one. If we zoom (Figure 4.6) on the first twenty years, we can see a phenomenon appearing which we cannot see with the Leslie model: every four years, there is a “step” in the population growth. This phenomenon has been observed in other game reserves (Pilanesberg) where a lot of young cows get mature and then pregnant at the same time. This phenomenon tends to disappear after thirty years or so.

If we want to obtain a stationary population, simulele advises us to use a calving interval of 13 years. The stable population is then around 60 elephants. Another possibility is changing the probability of surviving for a calf aged less than 1 year to 0.2769 (Figure 4.7).

If we choose to change the parameters according to environmental conditions based on a cosinus of period 20 years, we obtain Figure 4.8. But since the population is far from the carrying capacity, we can imagine that bad conditions are not going to affect the population very much. Figure 4.9 shows what happens if we take into account the carrying capacity. The population grows much faster than the one modeled with the Leslie model at the beginning (carrying capacity is reached after 63 years), but then we can see a decrease and the beginning of oscillations. However this example shows us that the model is not very realistic because the oscillations should occur around the carrying capacity. This model needs future adjustments but with some given parameters, the general behaviour seems to be promising.

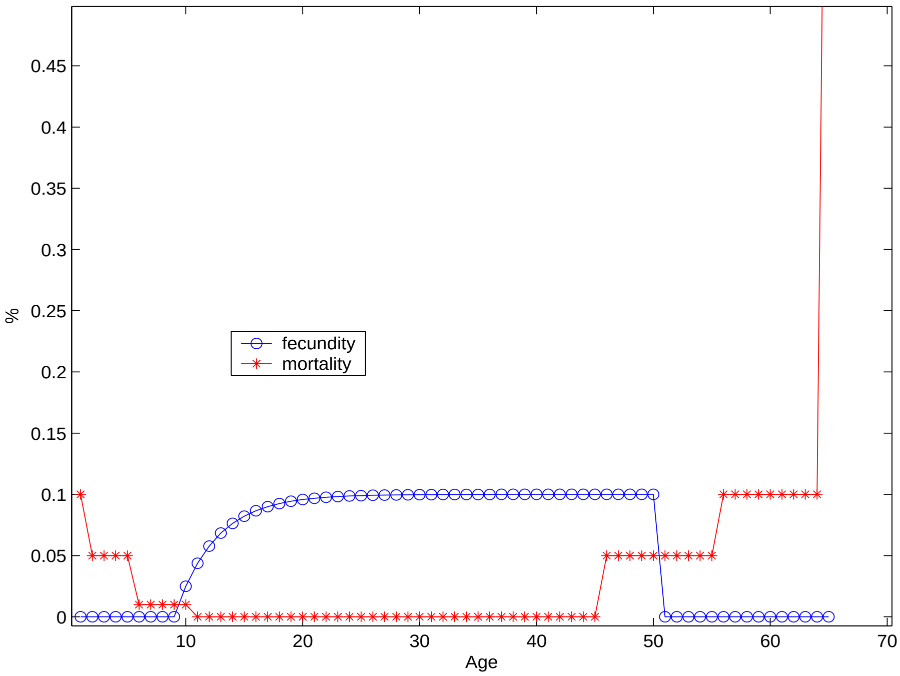


Figure 4.2: Fecundity and Mortality used for St Lucia population

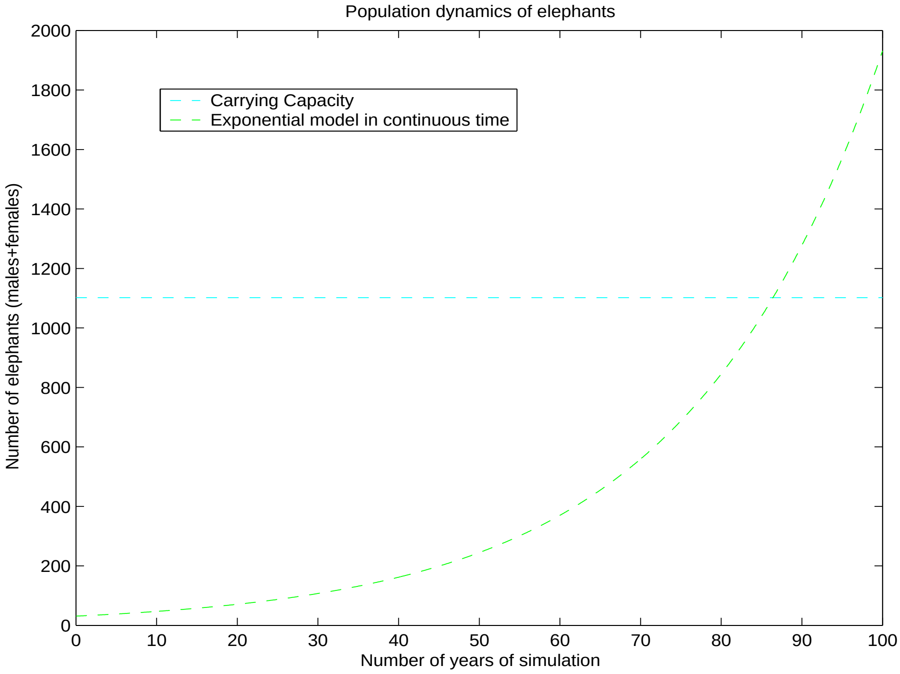


Figure 4.3: Exponential model applied to St Lucia population

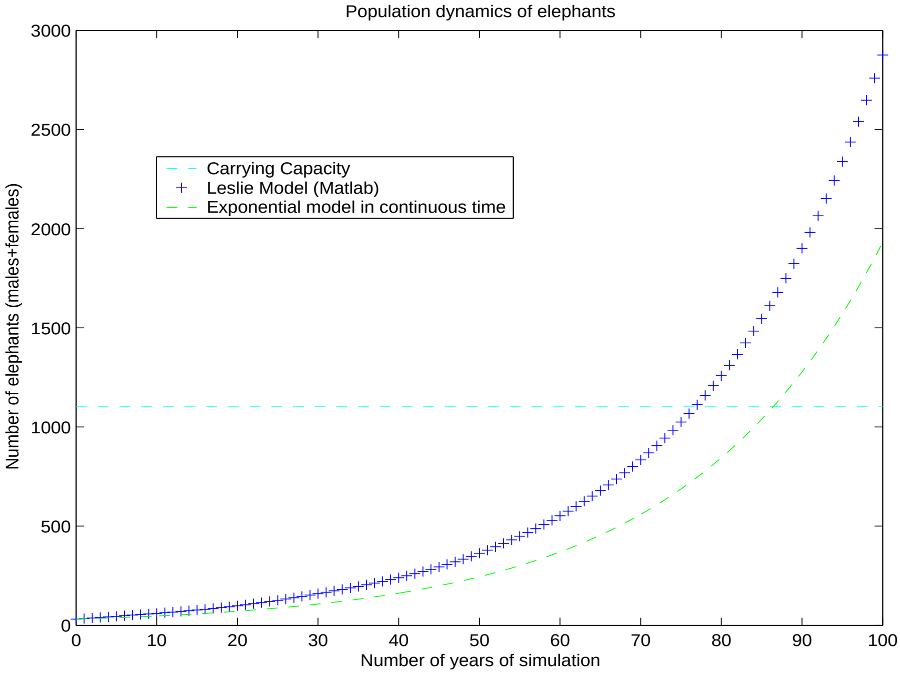


Figure 4.4: Leslie model applied to St Lucia population

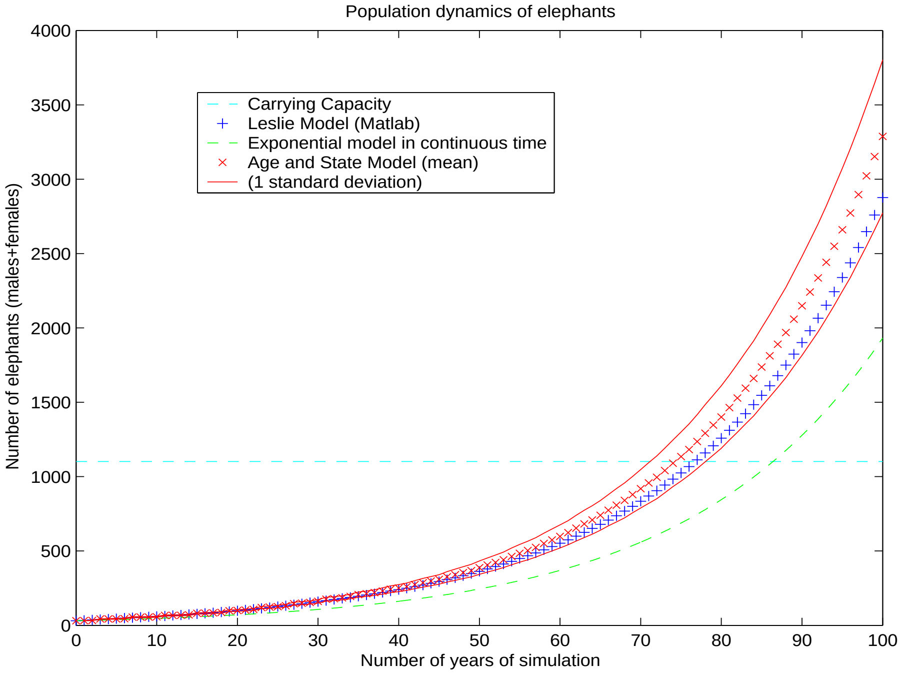


Figure 4.5: Age and State model applied to St Lucia population

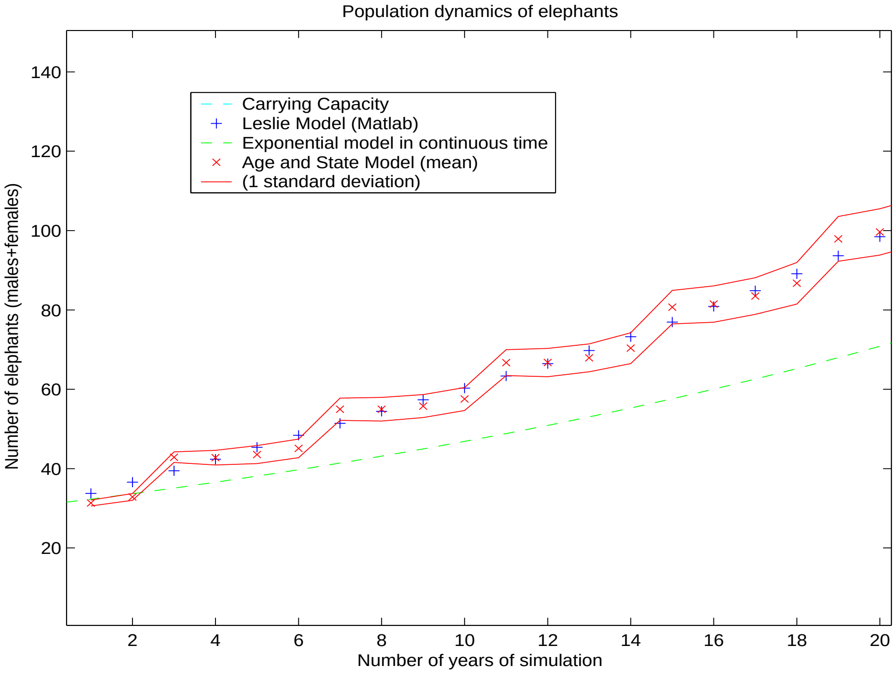


Figure 4.6: Zoom on the Age and State model

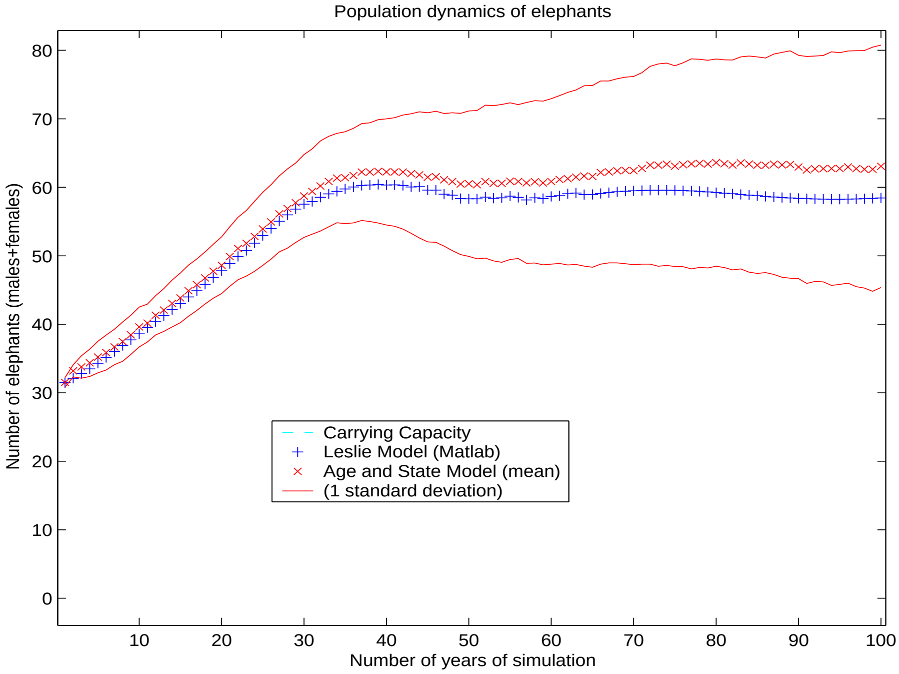


Figure 4.7: Stationary population

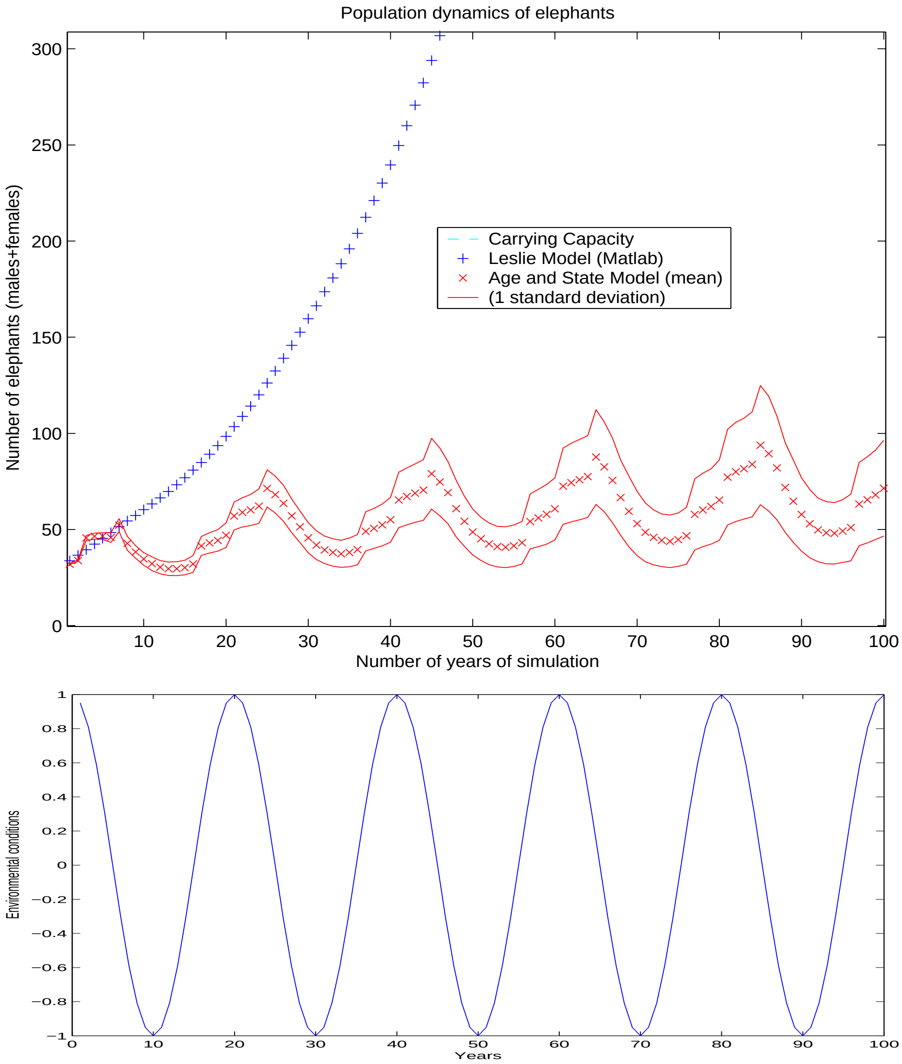


Figure 4.8: Age and State model used with environmental conditions

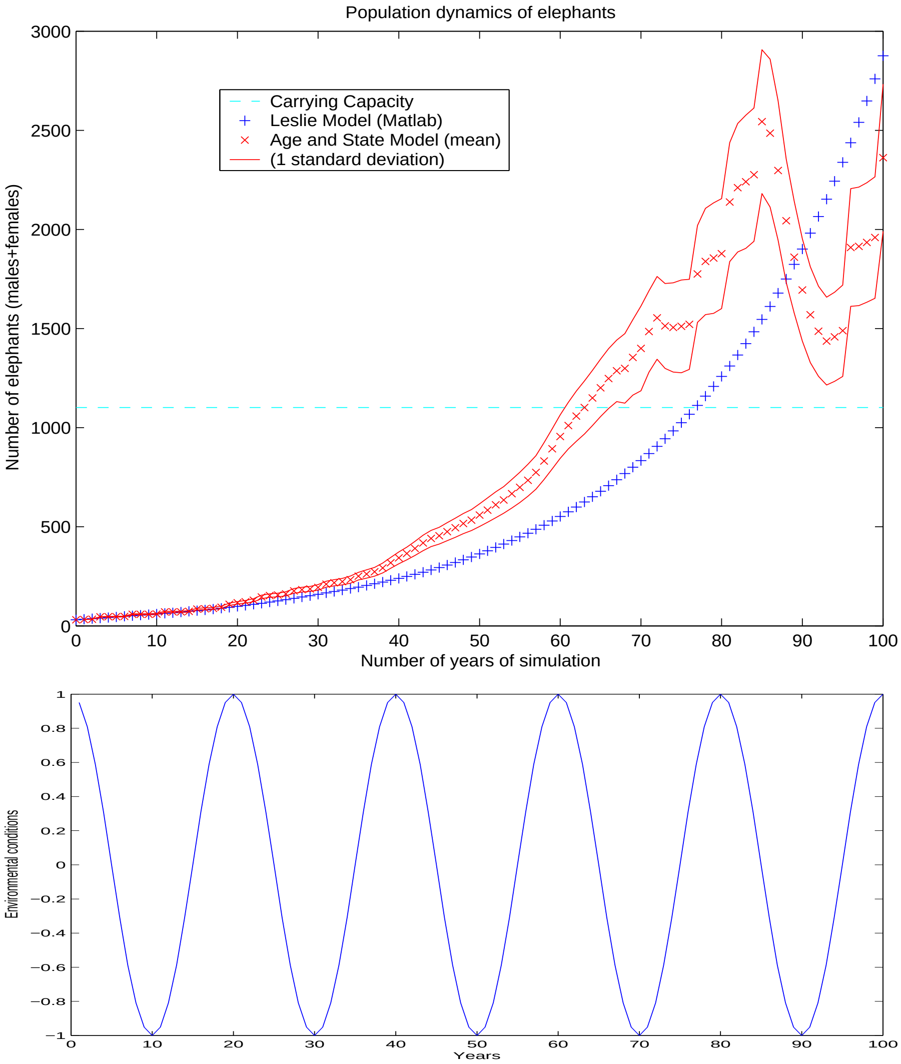


Figure 4.9: Age and State model used with environmental conditions and density dependence

Part II

Elephant-Tree ecosystems

Chapter 5

Caughley's model

In the previous part, we built several single species models because elephants have no serious predators. But elephants depend strongly on the amount of food available and especially trees. This is why it can be interesting to build some two species models.

In this chapter, we expose the Caughley's model for elephant-tree dynamics and the work that was done by [4] K.J. Duffy and [22] Hugh Murrell.

5.1 Model

The hypothesis that elephants and trees coexist in a stable limit cycle was first proposed by Graeme Caughley in 1976. The model is based on the well known predator-prey models developed by Lotka in 1925 and independently by Volterra in 1927. Caughley's version is an example of a more realistic predator-prey model, but its exact form is unique in the literature.

The model is given by the system of differential equations[1]:

$$\frac{dx}{dt} = x(a - \frac{a}{K}x - \frac{cy}{x+g}) \tag{5.1}$$

$$\frac{dy}{dt} = y(-A + \frac{kx}{y+B}) \tag{5.2}$$

where t is time (in year), x is the density of the trees (in trees/km^2) and y is the density of elephants(in elephants/km^2).

For the trees, a is the natural rate of increase (in year^{-1}), K is the tree carrying capacity of the environment (in trees/km^2), c is the instantaneous rate of elimination of trees by elephant (in trees/(elephant year)) and g determines the threshold above which tree destruction depends on elephant density alone (in trees/km^2).

[1] We can compare these equations with the logistic equation 2.6 page 25 describing a single species model.

For the elephants, A is their rate of decrease (in year^{-1}) in the absence of trees, k is the rate at which this decrease is ameliorated at a given ratio of trees to elephants (in elephant/(tree year)), and B determines the threshold above which this amelioration depends only on the density of trees (in elephant/km^2).

5.2 Parameters estimation

In [4] K.J Duffy et al. (1999), realistic parameters were estimated and Table 5.1 gives these parameter ranges:

Parameters	'Lower limit'	'Upper limit'	'Average conditions'
a	0.01	0.04	0.025
K	60000	500000	250000
c	40	1500	1200
g	4	100	15
k	5.9×10^{-6}	9.5×10^{-6}	7.5×10^{-6}
B	0.1	0.1	.01

Table 5.1: Parameter ranges for the Caughley's model

The only parameter which could not be estimated precisely is A, the rate of elephant decrease in the absence of trees. One expects A to be large, otherwise equations (6.1) and (6.2) would not apply to the system because other terms would be needed to describe other significant sources of food. For this reason, A was set equal to unity.

It was shown in [4] that limit cycles are theoretically possible for certain parameters conditions (see Appendix C page 135). With parameters of Table 5.2, we obtain the graph displayed in Figure 5.2. However these parameters are very unrealistic and elephant population is running to close to 0 to make this system biologically possible. In fact, only equilibrium solutions result for the ranges of parameters given in table 5.1 (Figure 5.1).

Parameters	a	K	c	g	k	A	B
Values	0.04	500000	15000	100	8×10^{-5}	1	0.1

Table 5.2: Parameters used to obtain limit cycles (unrealistic)

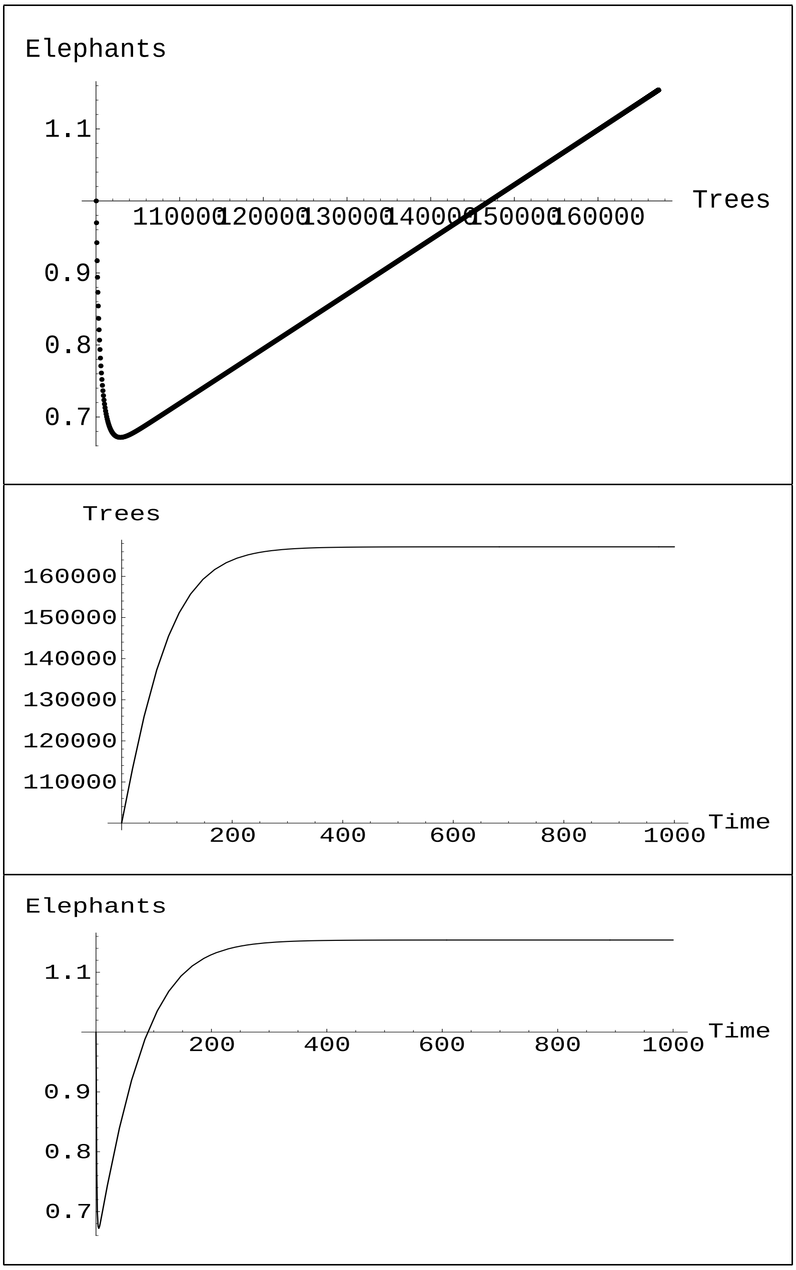


Figure 5.1: Model trajectory against time started at $x = 100000$ $trees/km^2$ and $y = 1$ $elephants/km^2$ with average conditions of Table 5.2

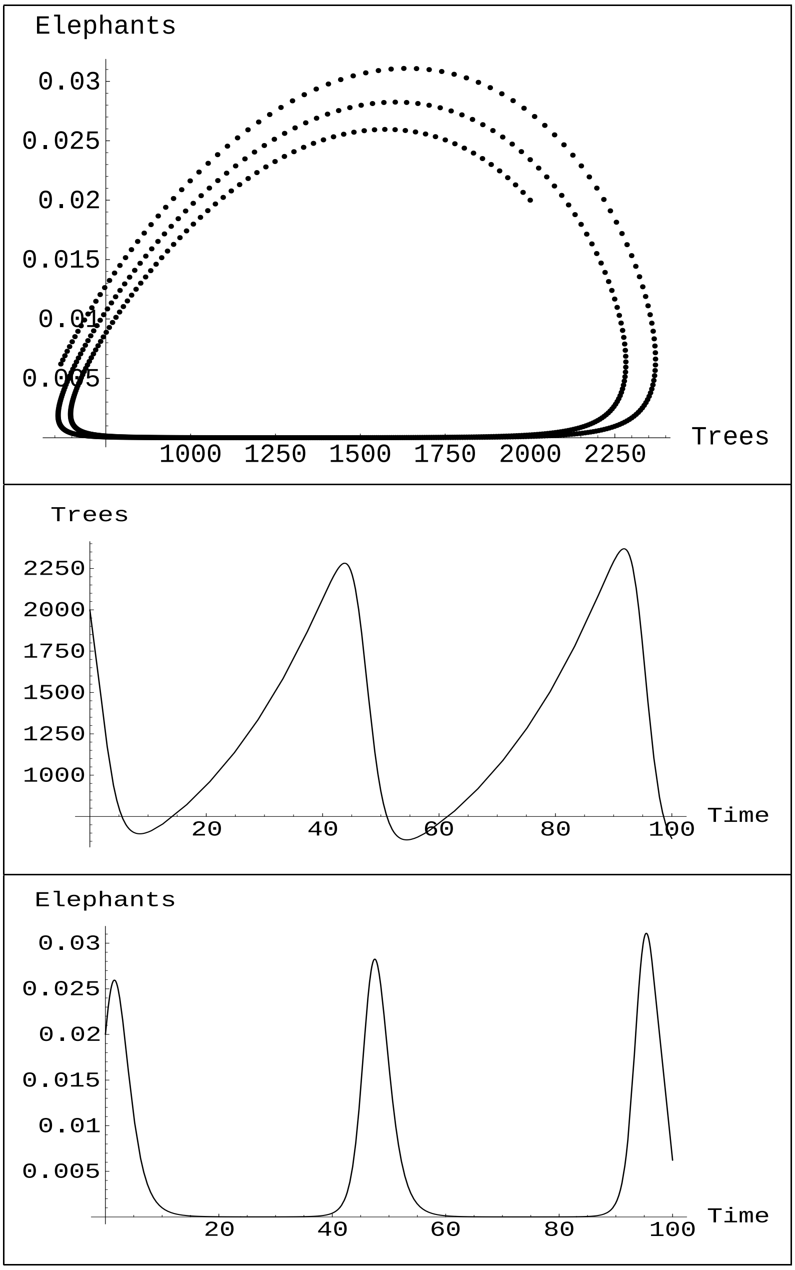


Figure 5.2: Model trajectory against time started at $x = 2000$ $trees/km^2$ and $y = 0.02$ $elephants/km^2$ with parameters of Table 5.2

5.3 Programming with Mathematica

Mathematica was used to plot the different graphs displayed in the last section. To find the solution of the differential equation, we can use either the Euler's method or the Mathematica function *NDSolve* (much faster).

Here is the function to plot the Model trajectory of the Elephant-Tree ecosystem:

```
predPreyTraj[a_,c_,g_,K_,A_,B_,k_,{T0_,E0_},{dt_,n_},opt_]:=Module[{},
    If[opt==0, (*Euler's Method*)
      Trees={T0};
      Elephants={E0};
      Ti=T0;
      Ei=E0;

      For[i=1,i<n,i++,
        Ti=Ti+dt (a Ti-a/K (Ti)^2-c Ti Ei/(Ti+g));
        Ei=Ei+dt (Ei (-A+k Ti/(Ei+B)));

        Trees=Append[Trees,Ti];
        Elephants=Append[Elephants,Ei];
        ];

      ListPlot[Transpose[{Table[j,{j,dt,n*dt,dt}],Trees}],
        AxesLabel->{"Time","Trees"}];
      ListPlot[Transpose[{Table[jj,{jj,dt,n*dt,dt}],Elephants}],
        AxesLabel->{"Time","Elephants"}];
      ListPlot[Transpose[{Trees,Elephants}],
        AxesLabel->{"Trees","Elephants"}]
      ];

    If[opt==1, (*Use function NDSolve *)
      solution=
        NDSolve[{x'[t]==a x[t]-a/K x[t]^2-c x[t] y[t]/(x[t]+g),
            y'[t]==y[t] (-A+k x[t]/(y[t]+B)),x[0]==T0
            y[0]==E0},{x,y},{t,0,n dt}];

      Plot[Evaluate[x[t] /. solution],{t,0,n dt},
        AxesLabel->{"Time","Trees"},PlotRange->All];
      Plot[Evaluate[y[t] /. solution],{t,0,n dt},
        AxesLabel->{"Time","Elephants"},PlotRange->All];
      ListPlot[
        Transpose[
          {Table[x[j] /. solution[[1]],{j,0,n dt,dt}],
            Table[y[j] /. solution[[1]],{j,0,n dt,dt}]}
          ],
        AxesLabel->{"Trees","Elephants"},PlotRange->All];
      ]
    ]
```

In order to obtain the equilibrium solution calculated with the function *NDSolve*, we use:

```
predPreyTraj[0.025, 1200, 15, 250000, 1, 0.1,
  7.5*10^-6, {100000, 1}, {0.1, 10000}, 1];
```

In order to obtain the limit cycles calculated with the function *NDSolve*, we use:

```
predPreyTraj[0.04, 15000, 100, 500000, 1, 0.1,
 8*10^-5, {2000, 0.02}, {0.1, 1000}, 1];
```

It could be interesting now to add space variables in this model.

Chapter 6

Spatio-temporal generalization

In this chapter, I try two different approaches to adapt the Caughley's model to a two dimensional lattice.

6.1 Reaction Diffusion

6.1.1 Model

Now assume that elephants and trees are spread out over a two-dimensional landscape representing a game reserve (a square of size L) so that now the tree density and the elephant density are function of space and time: x(u,v,t) and y(u,v,t). A first idea could be to assume that each species obeys a reaction-diffusion equation of the form:

$$\begin{aligned} \frac{dx}{dt} &= x(a - \frac{a}{K}x - \frac{cy}{x+g}) + D_1\nabla^2 x && (6.1) \\ \frac{dy}{dt} &= y(-A + \frac{kx}{y+B}) + D_2\nabla^2 y && (6.2) \end{aligned}$$

where D_1 and D_2 are coefficients of diffusion.

Because the game reserve is fenced, we must use the zero-flux boundary conditions (Neumann):

$$\frac{\partial x}{\partial n}(u,0) = \frac{\partial x}{\partial n}(u,L) = \frac{\partial x}{\partial n}(0,v) = \frac{\partial x}{\partial n}(L,v) = 0 \qquad (6.3)$$

$$\frac{\partial y}{\partial n}(u,0) = \frac{\partial y}{\partial n}(u,L) = \frac{\partial y}{\partial n}(0,v) = \frac{\partial y}{\partial n}(L,v) = 0 \qquad (6.4)$$

where $\vec{n}$ is the normal vector to the boundary and $\frac{\partial x}{\partial n}$ is equal to $\vec{n} \cdot \nabla x$.

The difficulty here is to determine the coefficients D_1 and D_2. But in fact, when we try the model with different initial conditions and different values for D_1 and D_2, we can see that this model always tends to homogeneity.

6.1.2 Programming with Mathematica

I used the package “ReactionDiffusionLab” (built by Selwyn Hollis) which provides two functions, RDDensityPlots and RDSurfacePlots, for animating solutions of autonomous reaction-diffusion systems with two or three components. The domain is the unit square, and boundary conditions are any combination of homogeneous Neumann type and (possibly nonhomogeneous) Dirichlet type. Time stepping is done via a combination of Crank-Nicolson and improved Euler’s methods. At each step, solution values are stored in a $n \times n$ matrix corresponding to a simple square grid with $\Delta u = \Delta v = \frac{1}{n-1}$. Regardless of the boundary conditions, the first/last rows correspond to the top/bottom edges of the region, and the first/last entries in all rows correspond to the left/right edges of the region. The two functions are as follows in the two-component case:

```
RDDensityPlots[{f[x, y], g[x, y]}, {x, y}, {x0, y0}, {d1, d2}, dt, {A1,
    A2}, nsteps, options]
RDSurfacePlots[{f[x, y], g[x, y]}, {x, y}, {x0, y0}, {d1, d2}, dt, {A1,
    A2}, nsteps, options]
```

In each of these, x0 and y0 are $n \times n$ matrices containing initial values, and $\{A1, A2\}$ is a pair of matrices returned by the function CNInverses as follows:

```
{A1,A2} = CNInverses[n,{d1,d2},dt];
```

The Package was slightly modified so that it is possible to plot the average density (sum of all local densities divided by n^2) for elephants and trees. We also added green colour for trees and red colour for elephants.

6.1.3 Results

Limit cycles

In order to obtain limit cycles, we use the same parameters as Table 5.2 page 84 on a 20 × 20 lattice. We take $dt = 0.01$ and we choose arbitrary small D_1 and D_2. For different D_1 and D_2, the system tends to homogeneity and oscillates exactly like the Caughley’s model predicts it.

```
a = 0.04;
K = 500000;
c = 15000;
g = 100;
k = 8*10^-5;
B = 0.1;
A = 1;
```

```
n = 20;
du=1/(n-1);
d1 = .01; d2 = .001; dt = .01;
{A1, A2} = CNInverses[n, {d1, d2}, dt];
```

The initial states are built randomly (for trees, density between 0 and 3000 with a high density in the center[1] and for elephants, density between 0.01 and 0.03)

```
x0=Table[3000*(1 - Sqrt[(u - 1/2)^2 + (v - 1/2)^2]/Sqrt[2]), {u, 0,1, du},
     {v,0, 1, du}];
y0 = Table[0.02 + 0.01 Random[Real, {-1, 1}], {u, 0, 1, du}, {v, 0, 1, du}];
```

We plot the results every 100 steps (every year) for 80 years

```
RDDensityPlots[{a x - a/K x^2 - c x y/(x + g), y (-A + k x/(y + B))},
    {x,y}, {x0, y0}, {d1, d2}, dt, {A1, A2}, 80, PlotEvery -> 100,
    PlotRange -> {{0, 4000}, {0, 0.04}}, ReturnLast -> True];
```

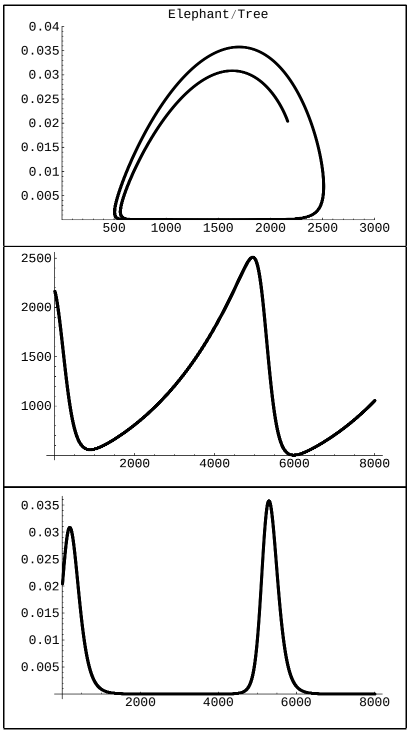


Figure 6.1: Model trajectories of average densities for elephants and trees ($D_1 = 0.01$ and $D_2 = 0.001$)(same axes labels as Figure 5.2 page 86)

[1] We can imagine there is a dam in the center.

For very small D_1 and D_2 (0.0001 and 0.00001), we obtain interesting behaviours, even if it is not realistic. Homogeneity disappears and some “waves” appear from the exterior and spread to the interior of the lattice.

Figure 6.2 and 6.3 show frames 1, 2, 3, 5, 9, 44, 59 and 67.

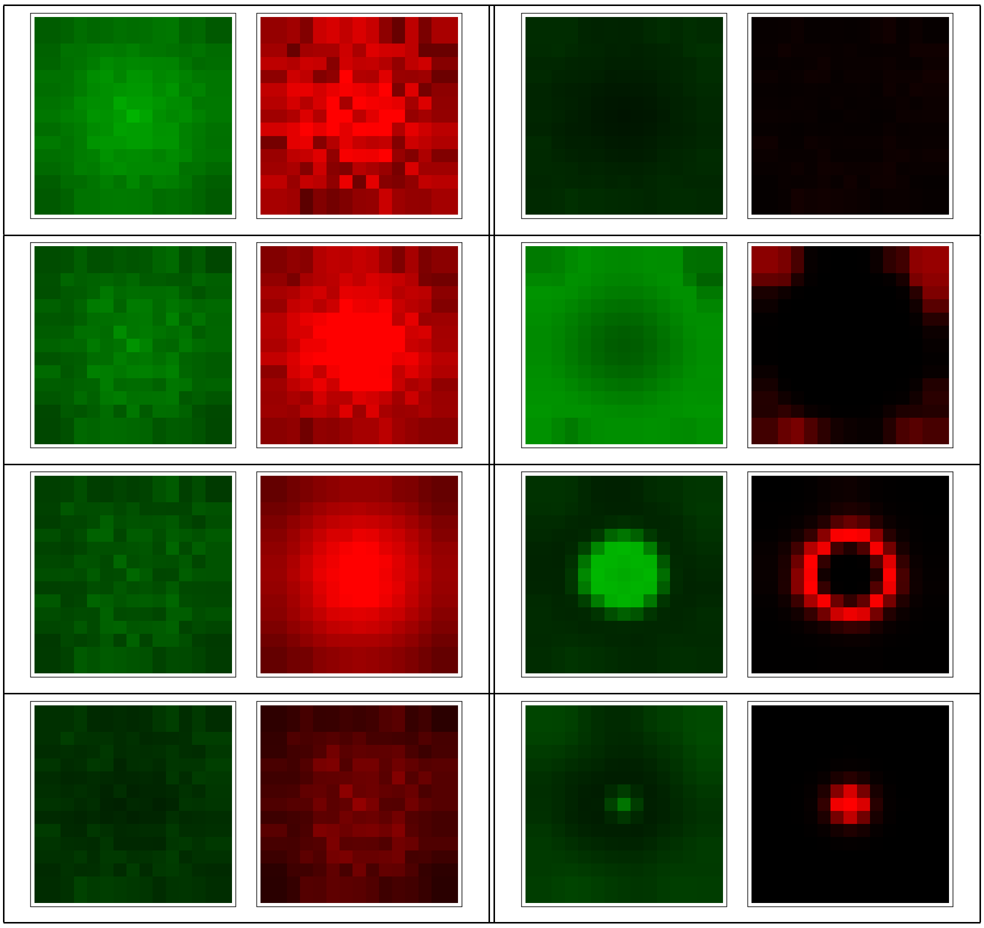


Figure 6.2: Interesting limit cycle obtained with very low D_1 and D_2, frames 1, 2, 3, 5, 9, 44, 59 and 67 (tree densities in green and elephant densities in red)

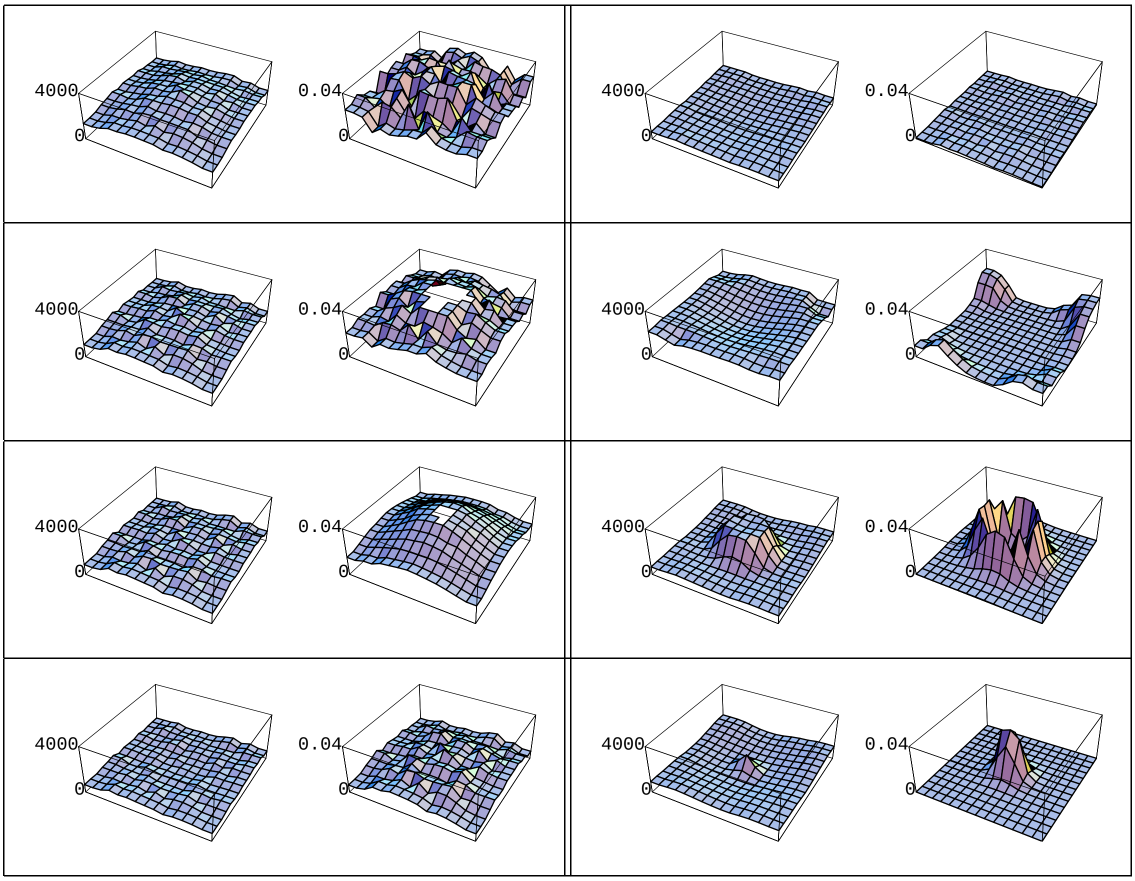


Figure 6.3: Interesting limit cycle obtained with very low D_1 and D_2, frames 1, 2, 3, 5, 9, 44, 59 and 67 (left surface: trees, right surface: elephants)

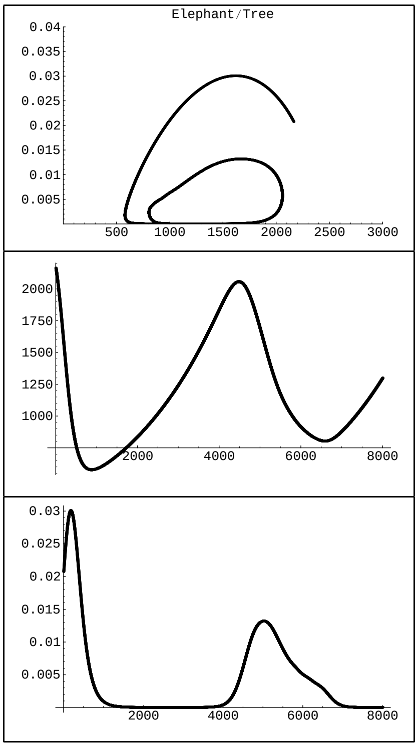


Figure 6.4: Interesting model trajectories of average densities for elephants and trees ($D_1 = 0.0001$ and $D_2 = 0.00001$) (same axes labels as Figure 5.2 page 86)

Equilibrium

If we use average parameters, whatever D_1 and D_2 we use, the system tends to homogeneity and then to equilibrium predicted by Caughley's model. Diffusion is perhaps not the best model for modelling elephant-tree dynamics, and we can try another one.

6.2 Other model

Assume, as [25] Julien Clinton Sprott proposes for foxes and rabbits, that elephants and trees interact not just with the other species in its own cell but also in the four nearest-neighbour cells (a von Neumann neighbourhood), giving:

$$\frac{dx}{dt} = x(a - \frac{a}{K}x - \frac{c\overline{y}}{x+g}) \tag{6.5}$$

$$\frac{dy}{dt} = y(-A + \frac{k\overline{x}}{y+B}) \tag{6.6}$$

where

$$\overline{x}(u,v) = \frac{x(u+d,v) + x(u,v+d) + x(u-d,v) + x(u,v-d) + \alpha_1 x(u,v)}{4+\alpha_1}$$

$$\overline{y}(u,v) = \frac{y(u+d,v) + y(u,v+d) + y(u-d,v) + y(u,v-d) + \alpha_2 y(u,v)}{4+\alpha_2}$$

is a weighted average of the neighbourhood. We can think of α_i as the tendency for the elephants to eat out of their place. In fact, in this model, we use the Caughley's model for each cell in a local neighbourhood.

6.2.1 Programming with Mathematica

First we define different functions which will allow us to display the results:

```
myColorFunctionTree[x_, max_] := RGBColor[0, x/max, 0];
myColorFunctionEle[x_, max_] := RGBColor[x/max, 0, 0];
myColorFunctionTreeEle[x_] :=
    If[x > 1, RGBColor[x - 1, 0, 0], RGBColor[0, x, 0]];

showTree[Tree_, opt___] :=
  ListDensityPlot[Tree, ColorFunctionScaling -> False,
    ColorFunction -> (myColorFunctionTree[#, maxTree] &), Mesh->
  False, opt]
showTree3D[Tree_, opt___] :=
  ListPlot3D[Tree, ColorFunctionScaling -> False, Mesh->False,
    PlotRange -> {0, maxTree}, opt]

showEle[Ele_, opt___] :=
  ListDensityPlot[Ele, ColorFunctionScaling -> False,
    ColorFunction->(myColorFunctionEle[#,maxEle]&),Mesh->False,opt]
showEle3D[Ele_, opt___] :=
  ListPlot3D[Ele, ColorFunctionScaling -> False, Mesh -> False,
    PlotRange -> {0, maxEle}, opt]

myMax[x_, y_] := If[Max[x, y] == x, x, y + 1];
showLandscape[Tree_, Ele_, {maxTree_, maxEle_}, opt___] :=
    ListDensityPlot[MapThread[myMax[#1, #2] &,
    {Tree/maxTree, Ele/maxEle}, 2],ColorFunctionScaling -> False,
     ColorFunction -> myColorFunctionTreeEle, Mesh -> False, opt]
```

Then we define functions that we will use in the main function:

```
Ker[alpha_] := { {0, 1, 0}, {1, alpha, 1}, {0, 1, 0}}/(4 + alpha)
myChop[x_, {min_, max_}] := If[x < min, min, If[x > max, max, x]]
```

Finally, we define the main function which calculates the next step[2] using Euler's method:

```
nextIteration[{Tree_,Ele_},{alpha1_,alpha2_},a_,c_,g_,K_,A_,B_,k_,dt_]:=
  Module[{nextTree, nextEle},
     nextTree =
      Map[myChop[#, {0, maxTree}] &,
        Tree + dt (a Tree - a/K Tree^2 -
           c Tree ListConvolve[Ker[alpha1], Ele, 2]/(Tree + g)), {2}];
    nextEle = Map[myChop[#, {0, maxEle}] &,
        Ele + dt(Ele
          (-A + k ListConvolve[Ker[alpha2],Tree,2]/(Ele + B))),{2}];
    {nextTree, nextEle}
    ]
```

6.2.2 Results

Equilibrium

Using average parameters (of Table 5.2) and a time step of 0.1 year, nothing surprising happens. The mean density for each species tends to its equilibrium value. One more time, the behaviour obtained using unrealistic parameters gives interesting results.

[2] We use cyclic convolution with a kernel (3 × 3 matrix) to calculate the weighted averages with Neumann neighbourhood.

Cycles

Figure 6.5 shows the results obtained using the same initial state as for diffusion (for trees, density between 0 and 3000 with a high density in the center[3] and for elephants, density between 0.01 and 0.03), parameters of Table 5.2, a time step of 0.1 and $\alpha_1 = \alpha_2 = 0.1$. We plot every 10 time steps (every year) for 100 years. We can see frames 1, 2, 3, 5, 9, 30, 40 and 75. This time the "wave" of vegetation and elephant appears in the center and spreads to the exterior.

```
maxTree = 2500;
maxEle = 0.03;

nbIter = 1000;
step = 0.1;
every = 10;
nbGrid = 50;

initTree[r_] :=
    Table[2000*(1 - Sqrt[(x - r/2)^2 + (y - r/2)^2]/r Sqrt[2]),
     {y, 1, r}, {x,1, r}];
initEle[r_] :=
    Table[0.015 + 0.01 Random[Real, {-1, 1}], {y, 1, r}, {x, 1, r}];

Tree = initTree[nbGrid];
Ele = initEle[nbGrid];

a = 0.04;
K = 500000;
c = 15000;
g = 100;
k = 8*10^-5;
B = 0.1;
A = 1;
alpha1=0.1;
alpha2=0.1;

animation =
  NestList[nextIteration[#,{alpha1,alpha2},a,c,g,K,A,B,k,step]&, {Tree, Ele},
   nbIter];

Table[Show[
    GraphicsArray[{showTree3D[animation[[every*i + 1, 1]],
```

[3] We can imagine there is a dam in the center.

```
     DisplayFunction -> Identity],
   showEle[animation[[every*i + 1, 2]],
     DisplayFunction -> Identity]}]], {i, 0, nbIter/every}]
```

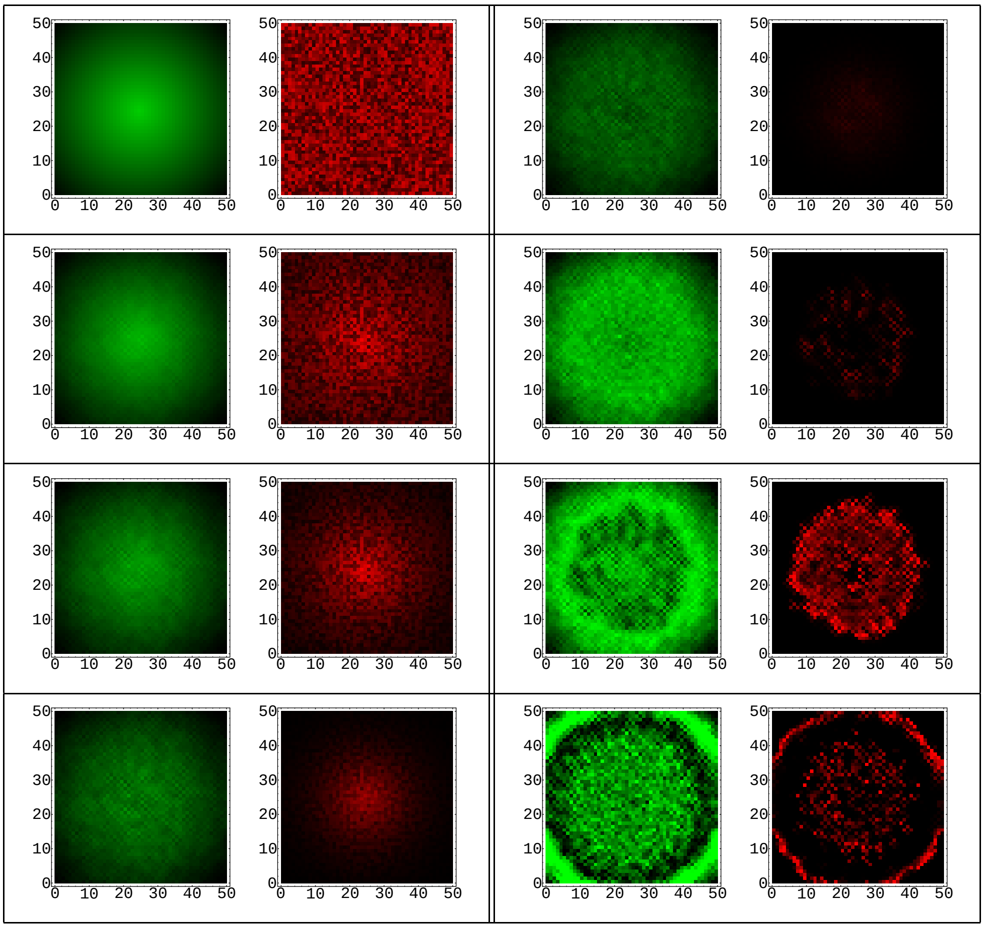


Figure 6.5: Interesting behaviour, frames 1, 2, 3, 5, 9, 30, 40, 75 (tree densities in green and elephant densities in red)

Figure 6.6 shows the total average density. The result is very different from the reaction-diffusion model and we do not see any limit cycle. The computer power on which I am working does not allow further investigations but it would be interesting to run the simulation on a longer time.

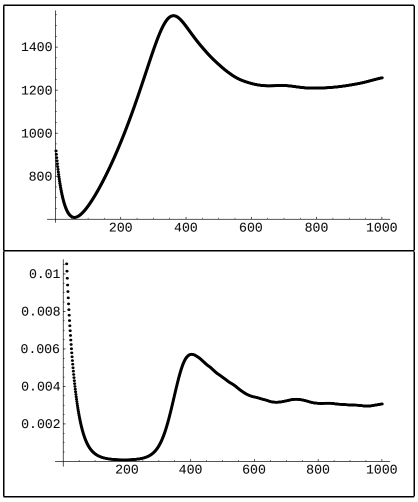


Figure 6.6: Model trajectory against time for the mean densities

Another way to display the result is to plot on the same lattice the elephants and trees. We plot in each cell the dominant species with two different colours : green for trees and red for elephants (Figure 6.7).

In this chapter, models are based on coupled differential equations and assume a continuous predator continuously in search of a continuous prey. It could be worth trying a discrete approach to modelling so called predator-prey ecosystems.

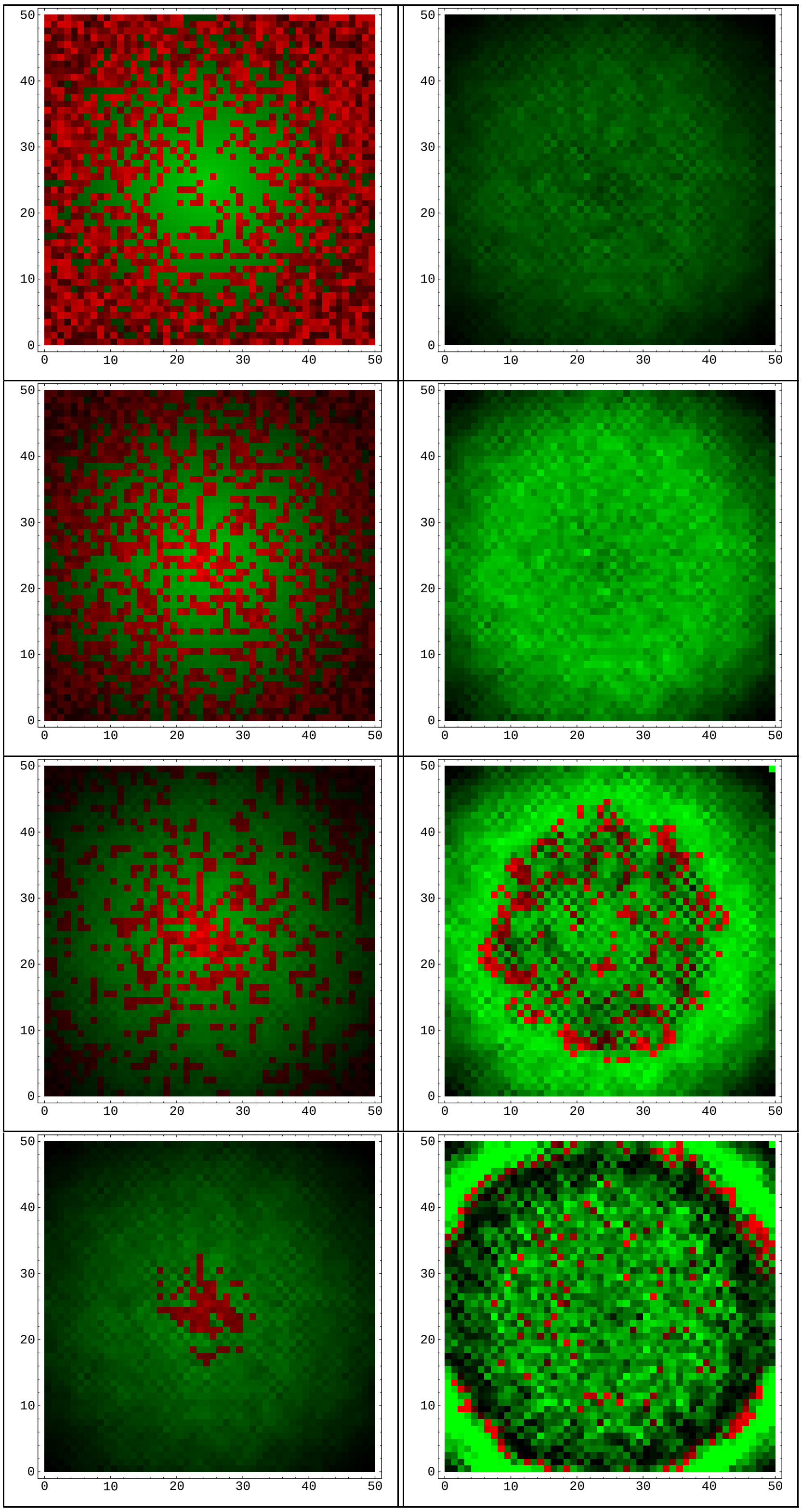


Figure 6.7: Landscape pattern showing the dominant species in each cell for frames 1, 2, 3, 5, 9, 30, 40, 75 (tree densities in green and elephant densities in red)

Chapter 7

Grazing herd cellular automaton model

In this last chapter, I try to obtain cycles as observed by Caughley using a completely different model based on a probabilistic cellular automaton. I use essentially the book [7] “Modeling Nature” by R.J. Gaylor and K. Nishidate (1996).

7.1 Cellular automata

A cellular automaton (CA) model is essentially a discrete space-time-state representation of a system of many objects that simultaneously interact with nearby objects. The sine qua non of a CA model is that the behaviour of the model is determined not by some centralized authority but by local interactions among decentralized components.

7.2 Model

The model employs a lattice system that represents territory that predators and prey co-habitate. Some of the lattice site are empty and other sites are occupied by a predator or a prey. In the grazing herd CA, the prey (trees) remain stationary while the predators (elephants) move randomly in vacant places in each time step. We will use periodic boundaries for simplicity.

The lattice sites that are empty or that have been destroyed by an elephant have value 0 and the lattice sites that contain trees still intact have value 1. The probability for a site that has been destroyed to regenerate is p (This gives a mean time of regeneration of $\frac{1}{p}$ units of time[1]).

The value of a lattice site occupied by an elephant is a tuplet $\{x, y, z\}$. The first component of the tuplet is an integer value 1, 2, 3 or 4 indicating the

[1] See the geometric distribution page 48.

direction faced by the elephant. The value 1 indicates a site occupied by a north-facing elephant, the value 2 indicates a site occupied by an east-facing elephant, the value 3 indicates a site occupied by a south-facing elephant, the value 4 indicates a site occupied by a west-facing elephant. The value of the first component is randomly chosen in each time step.

The second component of the tuplet is a nonnegative integer value indicating the amount of time that remains until a calf is due to born[2]. At the moment that the elephant gives birth, the value of the birth time clock is set to "preg" (this is the normal gestation period for elephants). At each ensuing time step, the value of the birth clock decreases by 1 unless its value is 0, in which case it remains 0. When the birth clock value is 0, the calf is born when the elephant moves but not while it remains in place.

The third component of the tuplet is a nonnegative integer value indicating the amount of time that remains before an elephant starves to death. At the moment that an elephant eats trees, its diet time clock is set equal to "starve". At each ensuing time step, the value of the diet clock decreases by 1. When the diet clock value is 0, the elephant dies and vanishes from the lattice.

Initially, elephants and trees are randomly placed on the lattice with probability eleDensity and treeDensity respectively [3].

7.3 Programming with Mathematica

There are a number of features of the Mathematica programming language that make it especially appropriate for the task of writing CA programs:

- Mathematica is fundamentally a term rewriting system (TRS) based on pattern matching.
- Mathematica can manipulate data structures in their entirety, rather than in a piecemeal fashion.
- Mathematica has an extensive, easy-to-use graphics capabilities.

The rules governing the behaviours of the elephants and trees are implemented as rewrite rules for updating the values of the sites of the CA lattice. These rules take 13 arguments in the following order:

```
eco[site,N,E,S,W,NE,SE,SW,NW,Nn,Ee,Ss,Ww]
```

where the 13 arguments represent the value of the site, the values of the four nearest neighbours in the N, E, S, W directions, the values of the four

[2] This implies that we are modelling female population only.

[3] The density for trees here is different from the previous chapter. The density for trees represents the ratio of cell number containing trees still intact over the total number of cells.

nearest neighbours in the NE, SE, SW, NW directions and the values of the four next nearest neighbours in the N, E, S, W directions.

I am not going to display the main program which can be found and described in the book [7] “Modeling Nature” by R.J. Gaylor and K. Nishidate (1996), or in Appendix D.

There are two ways to look at the results. One (used in [7]) is calculating and plotting the number of elephants and trees as a function of time.

I created functions that allow to display the lattice as a landscape. Cells containing trees are green and cells containing elephants are red. I used a gradation of red to show the diet time clock. Light red represents healthy elephants whereas dark red represents elephants which are close to starve to death. I also used the colour blue to show female elephants which are calving (birth clock is equal to 0).

Here is an example (not very realistic) of the result running the program with n=40 (the lattice can represent a park of 40 × 40 kms), treeDensity=0.4, eleDensity=0.4, preg=36 (if we take months for time unit, it gives a calving interval of 3 years), starve=10 (if an elephant does not meet a cell full of trees in 10 months, the elephant will die), p=$\frac{1}{12}$ (a cell destroyed by an elephant can be regenerated in 1 year), and t=240 months (we run the simulation for 240 months which is 20 years).

```
n = 40;
treeDensity = 0.4;
eleDensity = 0.4;
preg = 36;
starve = 10;
p = 1/12;
t = 240;

SeedRandom[2];
graze = PredatorPrey[n,treeDensity,eleDensity,preg,starve,p,t];
```

We plot the number of elephants and trees as a function of time using Mathematica function “MultipleListPlot” (Figure 7.1):

```
predatorPop = Map[Count[Flatten[#, 1], {__}]&, graze];
preyPop = Map[Count[Flatten[#, 1], 1]&, graze];
PredPreyPop = {predatorPop, preyPop};
PredPreyPopTimePairs = Map[Transpose[{Range[0, t], #}]&, PredPreyPop];

MultipleListPlot[PredPreyPopTimePairs[[2]], PredPreyPopTimePairs[[1]],
  PlotJoined -> True, AxesLabel -> {"Time(months)", "Population"},
  PlotLegend -> {"Trees", "Elephants"}, LegendPosition -> {0, -0.2},
  LegendSize -> {0.7, 0.3}]
```

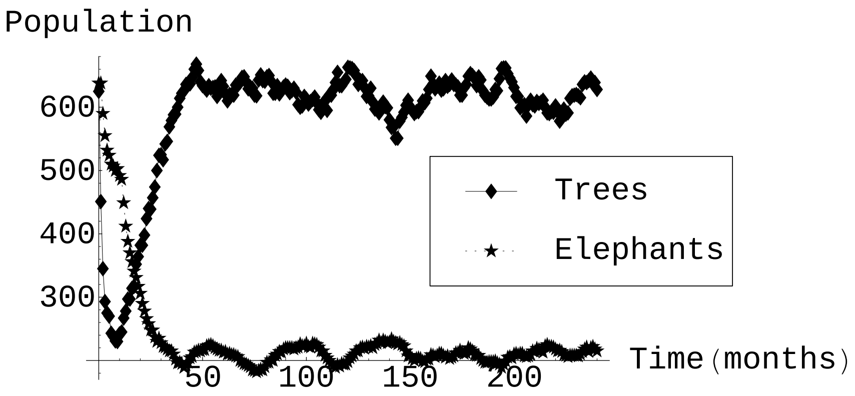


Figure 7.1: Number of cells containing elephants and trees as a function of time

As Figure 7.1 shows, elephants and trees display oscillatory population density fluctuations which are-out-of-phase with one another. This cycle behaviour is not surprising. At the beginning, food source is not sufficient for the whole population so elephants start dying of starvation and therefore breed less. When the size of the herd is sufficiently diminished (after 10 months), the growth of new food source begins increasing. After 50 months, the two species have reached a sort of equilibrium and we can see small cycles appearing (the elephants eat (and destroy) trees and reproduce, increasing the size of the herd while depleting the food source supply and so on).

Figure 7.2 shows landscapes for months 1, 2, 3, 4, 201, 202, 203 and 204 (Using Mathematica function “ListDensityPlot”).

```
myColorFunction[x_] :=
 If[x == -1, RGBColor[0, 0, 1],
   If[x == 1, RGBColor[0, 1, 0],
     If[x > 1, RGBColor[(x - 1)/(starve + 1), 0, 0], RGBColor[0, 0, 0]]]];
graze1 = Map[Replace[#, {_, y_, x_} -> If[y == 0, -1, 2 + x], 2] &, graze];
Table[ListDensityPlot[graze1[[j]], ColorFunctionScaling -> False,
  ColorFunction -> myColorFunction, Frame -> True], {j, 1,Length[graze1]}]
```

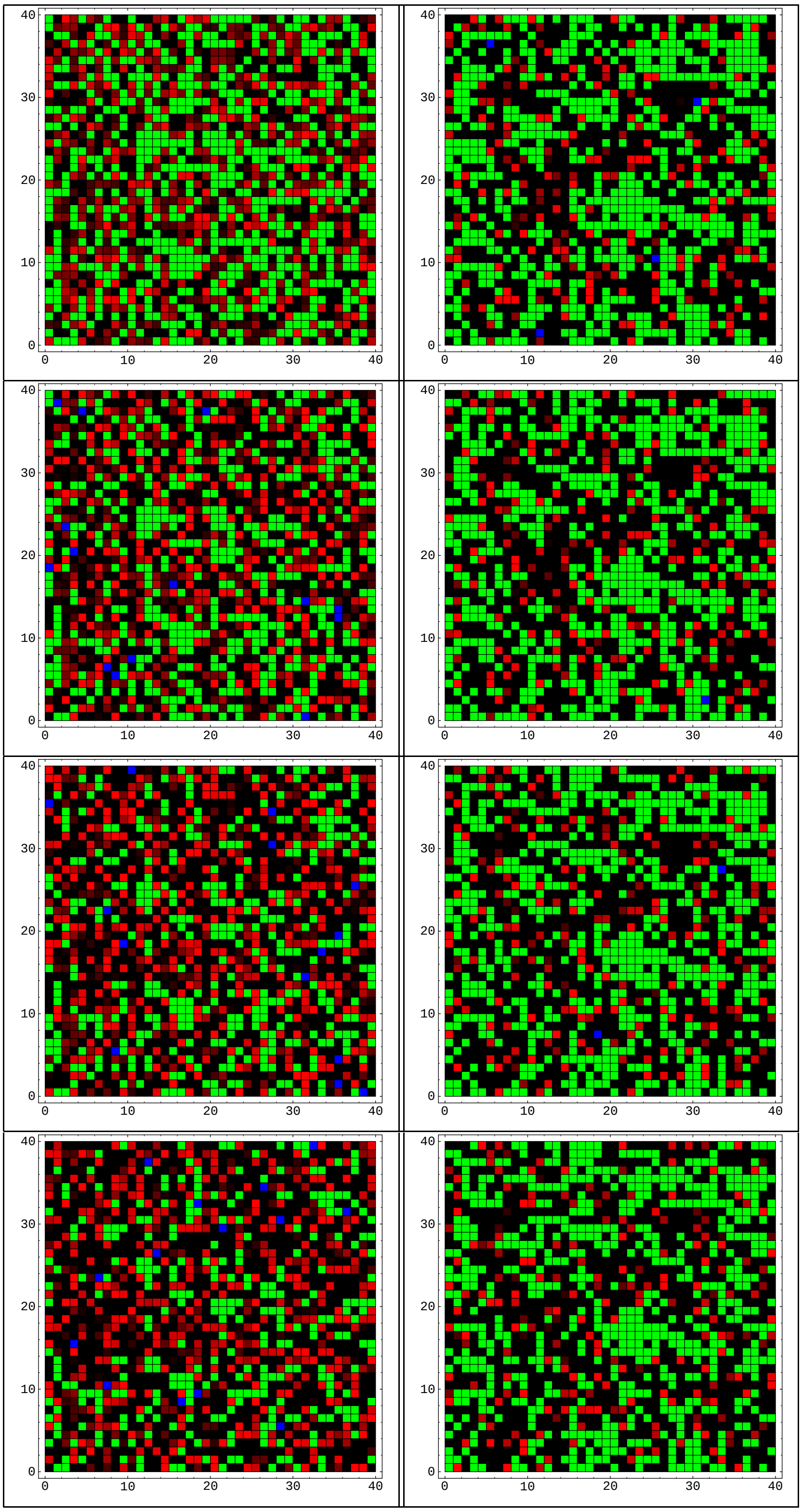


Figure 7.2: Landscapes for the cellular automaton model, frames 1, 2, 3, 4, 201, 202, 203 and 204 (Trees in green, elephants in red graduations and elephants with birth clock equal to 0 in blue)

This model is promising and using system based on cellular automata instead of differential equations could perhaps explain in a better way the cycles observed by Caughley. Unfortunately, I had no sufficient time to attempt making this model more realistic but we can imagine it could be a very interesting project for the future.

Conclusion

The creation of a stand alone application with an easy-to-use graphical interface was a success and no bugs have been yet reported. Users (probably reserve managers and biologists) can test four different models: two in continuous time (Exponential model and s-shape model) and two in discrete time (Leslie model and Age and State model).

Clearly, the most realistic one is the Age and State model but it can still be improved. Data concerning elephant mortality and fecundity need to be collected in different game reserves and a statistical work has to be done in order to find relationships between rainfall, carrying capacity and parameters describing elephant mortality and fecundity. Using this report, it is then easy to understand the program which is behind the graphical interface and to modify it. However, the current program simulele can be used with accuracy to predict population growth in game reserves which population is far from the carrying capacity such as St Lucia.

It can also be very interesting for people working on contraception programmes to see the theoretical calving interval given by simulele so that the elephant population is stationary.

The work on elephant-tree ecosystems sums up what has been done in the past with the Caughley's model and tries to generalize adding space variables to it. Two different models are tried in continuous time but the most promising one is based on a discrete model involving cellular automata.

It seems in both parts of my report that discrete models based on simple programs are much more adapted than continuous models based on equations to describe the complex behaviour of population dynamics. I will end with a quotation from [28] A New Kind of Science by Stephen Wolfram (page 368):
" It does not help that models based on equations are often stated in a purely implicit form, so that rather than giving an actual procedure for determining how a system will behave—as a program does—they just give constraints on what the behavior must be, and provide no particular guidance about finding out what, if any, behavior will in fact satisfy these constraints. [...] Models based on simple programs essentially involve only discrete elements which can be handled quite directly on a practical computer. And this means that it becomes straightforward in principle—and often highly efficient in

practice—to work out at least the basic consequences of such models."

Appendices

Appendix A

Some Theorems and proofs for the Leslie Matrix model

In all these theorems and proofs, we use the same notation as in Section 2.2 page 26.

Theorem 1 *The eigenvalues of L are given by the solutions to an equation of the form*

$$1 = \sum_x \lambda^{-x} l_x b_{fx} \tag{A.1}$$

Proof: Write

$$\begin{aligned}
0 &= \begin{vmatrix} p_1 b_{f1} - \lambda & p_2 b_{f2} & p_3 b_{f3} & \dots & p_k b_{fk} \\ p_1 & -\lambda & 0 & \dots & 0 \\ 0 & p_2 & -\lambda & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & p_{k-1} & -\lambda \end{vmatrix} \\
&= (-1)^{1+1}(p_1 b_{f1} - \lambda) \begin{vmatrix} -\lambda & 0 & \dots & 0 \\ p_2 & -\lambda & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \dots & p_{k-1} & -\lambda \end{vmatrix} + \\
&\qquad (-1)^{1+2}(p_2 b_{f2}) \begin{vmatrix} p_1 & 0 & \dots & 0 \\ 0 & -\lambda & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \dots & p_{k-1} & -\lambda \end{vmatrix} + \ etc \dots \\
&= (+1)(p_1 b_{f1} - \lambda)(-\lambda)^{k-1} + (-1)(p_2 b_{f2})[p_1(-\lambda)^{k-2}] + \dots \\
&= (p_1 b_{f1} - \lambda)(-1)^{k-1}\lambda^{k-1} - (p_2 b_{f2})p_1(-1)^{k-2}\lambda^{k-2} + (p_3 b_{f3})p_1 p_2(-1)^{k-3}\lambda^{k-3} - \dots \\
&= (-1)^{k-1}[\lambda^k - p_1 b_{f1}\lambda^{k-1} - p_1 p_2 b_{f2}\lambda^{k-2} - p_1 p_2 p_3 b_{f3}\lambda^{k-3} + \dots
\end{aligned}$$

Thus

$$\begin{aligned} 0 &= \lambda^k - p_1 b_{f1}\lambda^{k-1} - p_1 p_2 b_{f2}\lambda^{k-2} - p_1 p_2 p_3 b_{f3}\lambda^{k-3} - \dots \\ &= \lambda^k - l_1 b_{f1}\lambda^{k-1} - l_2 b_{f2}\lambda^{k-2} - l_3 b_{f3}\lambda^{k-3} - \dots \end{aligned}$$

Finally,

$$\lambda^k = l_1 b_{f1}\lambda^{k-1} - l_2 b_{f2}\lambda^{k-2} - l_3 b_{f3}\lambda^{k-3} - \dots$$

Dividing through by λ^k gives us

$$1 = l_1 b_{f1}\lambda^{-1} - l_2 b_{f2}\lambda^{-2} - l_3 b_{f3}\lambda^{-3} - \dots = \sum_{x=1}^{k} \lambda^{-x} l_x b_{fx}$$

Theorem 2 *Equations of the form*

$$1 = \sum_{x=1}^{k} \lambda^{-x} l_x b_{fx} \tag{A.2}$$

have one and only one real positive root λ_1 of algebraic multiplicity 1. The other roots λ_i are either negative or complex, and $\lambda_0 > |\lambda_i|$

Proof: The function of λ:

$$f(\lambda) = \sum_x \lambda^{-x} l_x b_{fx}$$

is positive on $]0; +\infty[$ and strictly decreasing from ∞ to 0 (since all l_x and b_{fx} are positive). Hence there is only one real positive root of multiplicity one.

Let λ_i be any other latent root, and write $\lambda_i^{-1} = e^{\alpha + i\beta}$, where α and β are real and positive, and $\beta \neq 2r\pi$. That is, λ_i must be negative or complex.. We then have

$$\lambda_i^{-n} = (e^{\alpha}(cos\beta + isin\beta))^n = e^{n\alpha}(cos(n\beta) + isin(n\beta))$$

Substituting for λ_i^{-n} in equation (A.2) and equating the real parts of the left-hand side and right-hand side, we obtain

$$\sum_x l_x b_{fx} e^{x\alpha} cos(x\beta) = 1 \tag{A.3}$$

Since $\beta \neq 2r\pi$, $cos(x\beta)$ and $cos((x+1)\beta)$ cannot both be unity and so either or both must be less than unity. A comparison of equation (A.2) and (A.3) shows that e^{α} must be greater than λ_1^{-1}, and we can conclude that $|\lambda_i| < \lambda_1$ for all $i \neq 1$.

Theorem 3 (Theorem of Perron-Frobenius) *Let T be a square matrix with non-negative elements only, and such that all elements of T^n are positive for some positive integer n (we say that T is primitive). Then T has a positive eigenvalue of algebraic multiplicity one, which correspond to a right eigenvector U_1 and to a left eigenvector V_1', both of which have only positive elements. This eigenvalue is greater in absolute size than any other eigenvalues of T.*

Theorem 4 *The general solution to an equation such as*

$$N_f(t) = L^t \cdot N_f(0) \tag{A.4}$$

with L diagonalizable, is

$$n_{fi}(t) = \sum_j C_{ij} \lambda_j^t \tag{A.5}$$

where λ_j are the eigenvalues of L and C_{ij} are constants whose values depend on the initial population vector.

Proof:

$$\begin{aligned}
N_f(t) &= L^t \cdot N_f(0) \\
&= (P\Delta P^{-1})^t \cdot N_f(0) \\
&\quad \text{(where } P \text{ is the eigenvector matrix and} \\
&\quad \Delta \text{ is the diagonal matrix containing the eigenvalues of } L \text{ denoted} \lambda_i) \\
&= (P\Delta P^{-1})(P\Delta P^{-1})(P\Delta P^{-1})\ldots(P\Delta P^{-1}) \cdot N_f(0) \\
&= P(\Delta P^{-1}P)(\Delta P^{-1}P)(\Delta P^{-1}P)\ldots(\Delta P^{-1}P)\Delta P^{-1} \cdot N_f(0) \\
&= (P\Delta^t P^{-1}) \cdot N_f(0)
\end{aligned}$$

where

$$\Delta^t = \begin{pmatrix} \lambda_1^t & 0 & \ldots & 0 \\ 0 & \lambda_2^t & \ldots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \ldots & \lambda_k^t \end{pmatrix}$$

Now, consider any element of $N_f(t)$, noticing that it is a sum of the various λ_i^t, each multiplied by a value reflecting a combination of the elements of P, P^{-1} and $N_f(0)$. That is

$$n_{fi}(t) = \sum_j C_{ij} \lambda_j^t$$

where $\{\lambda_j\}$ are the eigenvalues of L and the C_{ij} are scalar constants whose values depend on the initial population vector.

Theorem 5 *Let T be a square matrix of order $k+1$ with an algebraically simple strictly dominant eigenvalue λ_1. Let u_1 and v_1' be the corresponding right and left eigenvectors. It is possible to choose u_1 and v_1' such that $v_1' \cdot u_1 = 1$, and then for large values of n,*

$$\frac{T^n}{\lambda_1^n} = u_1 v_1' + E(n) \tag{A.6}$$

where the moduli of the elements of $E(n)$ are not greater than a number of order $n^{k-1}|\frac{\lambda_i}{\lambda_0}|$ and $|\lambda_i| < \lambda_0$.

Proof: Let λ_2 be the eigenvalue with the second largest modulus. There exists a non-singular matrix H such that

$$T = H \begin{pmatrix} \lambda_1 & 0' \\ 0 & Z \end{pmatrix} H^{-1}$$

where Z consists of diagonnaly arranged blocks like

$$Z(r) = \begin{pmatrix} \lambda_r & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_r \end{pmatrix}$$

and each λ_r is an eigenvalue of T (Turnbull and Aitken, 1951).
Let us consider a 3×3 block as an example. If

$$Z(r) = \begin{pmatrix} \lambda_r & 1 & 0 \\ 0 & \lambda_r & 1 \\ 0 & 0 & \lambda_r \end{pmatrix}$$

then

$$(Z(r))^5 = \begin{pmatrix} \lambda_r^5 & 5\lambda_r^4 & 10\lambda_r^3 \\ 0 & \lambda_r^5 & 5\lambda_r^4 \\ 0 & 0 & \lambda_r^5 \end{pmatrix}$$

The upper triangular elements of $(Z(r))^5$ belong to the binomial expansion of $(\lambda_r + 1)^5$. In the more general situation, $Z(r)$ is a $p \times p$ matrix, and we see that for large n, the modulus of the largest element of $(Z(r))^n$ will be of order $n^{p-1}|\lambda_2|^n$.

The matrix T is of dimension $(k+1) \times (k+1)$, and the worst possible case occurs when $Z(i)$ is of dimension $k \times k$. It follows that

$$\frac{T^n}{\lambda_1^n} = H \begin{pmatrix} \lambda_1 & 0' \\ 0 & 0 \end{pmatrix} H^{-1} + E(n) \tag{A.7}$$

where the moduli of the elements of $E(n)$ are all less than or equal to a number of order $n^{k-1}|\frac{\lambda_2}{\lambda_1}|^n$.
The next step is to note that

$$\lim_{n\to\infty}\frac{T^n}{\lambda_1^n} = H\begin{pmatrix}\lambda_1 & 0' \\ 0 & 0\end{pmatrix}H^{-1},$$

and because u_1 is the right eigenvector of T corresponding to λ_1, we must have

$$H^{-1}u_1 = \begin{pmatrix}\lambda_1 & 0' \\ 0 & 0\end{pmatrix}H^{-1}u_1 = \alpha\begin{pmatrix}1 \\ 0\end{pmatrix}.$$

In a similar manner, $v_1'H = \beta(1\ \ 0')$.
The constants α and β are both non-zero, and it is therefore possible to normalize u_1 and v_1' so that

$$v_1'u_1 = v_1'HH^{-1}u_1 = \alpha\beta = 1.$$

We now note that

$$u_1v_1' = H\begin{pmatrix}1 & 0' \\ 0 & 0\end{pmatrix}H^{-1}u_1 = \alpha\begin{pmatrix}1 \\ 0\end{pmatrix}H^{-1},$$

and it follows from equation (A.7) that

$$\frac{T^n}{\lambda_1^n} = u_1v_1' + E(n)$$

Theorem 6 (Fundamental theorem of demography) *Let T be the projection matrix of a standard population model $N_f(t)$. Suppose that T is primitive (such that all elements of T^n are positive for some positive integer n) with spectral radius $\rho(T) = \lambda_1$ and has left and right Perron vectors V_1' and U_1 respectively normalized so that $V_1'U_1 = 1$. Then*

$$\lim_{t\to\infty}\frac{N_f(t)}{\lambda_1^t} = V_1'\cdot N_f(0)\cdot U_1. \tag{A.8}$$

consequently, if we denote $|N_f(t)|$ the sum of entries of the vector $N_f(t)$, so that $|N_f(t)|$ will denote the total population at time t in the population model, then

$$\lim_{t\to\infty}|N_f(t)| = \begin{cases} 0 & \textit{if } \lambda_1 < 1, \\ |V_1'\cdot N_f(0)\cdot U_1| & \textit{if } \lambda_1 = 1, \\ \infty & \textit{if } \lambda_1 > 1. \end{cases} \tag{A.9}$$

Appendix B

Fortran code

Here is the code contained in file ModelingEle7.for.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!       ELEPHANT POPULATION DYNAMICS MODEL, VERSION 1.7 JUNE 2003
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!        THE DATA SHOULD BE ENTERED IN THE FILE DATA.TXT IN THIS ORDER :
!
!       1/ PAR(I) IS THE ONE DIMENSIONAL ARRAY CONTAINING THE DEMOGRAPHIC PARAMETERS
!
!       PAR(1) = NUMBER OF REPETITIONS OF THE ITERATION PERIOD
!       PAR(2) = NUMBER OF ITERATIONS IN EACH REPETITION
!       PAR(3) = MAXIMUM AGE (IN PERIODS OF ITERATION) OF THE SPECIES
!       PAR(4) = LENGTH OF THE GESTATION PERIOD IN PERIODS OF ITERATION
!       PAR(5) = LENGTH OF THE LACTATION PERIOD IN PERIODS OF ITERATION
!       PAR(6) = MINIMUM AGE OF SEXUAL MATURITY
!       PAR(7) = MEAN AGE OF SEXUAL MATURITY
!       PAR(8) = AGE OF MENOPAUSE
!       PAR(9) = CARRYING CAPACITY OF THE PARK
!       PAR(10) = OPTION (0 : NO OPTION
!                         1 : WE KILL ORPHAN ELEPHANTS (WHICH MOTHERS WERE LACTATING)
!                         2 : PARAMETERS VARY LINEARLY DEPENDING ON VALUES OF
!                             THE ENVIRONMENT
!                         3 : PARAMETERS VARY LINEARLY DEPENDING ON VALUES OF
!                             THE ENVIRONMENT AND THE CARRYING CAPACITY)
!                         4 : WE KILL ORPHAN CALVES AND PARAMETERS VARY LINEARLY
!                             DEPENDING ON VALUES OF THE ENVIRONMENT
!                         5 : WE KILL ORPHAN CALVES AND PARAMETERS VARY LINEARLY
!                             DEPENDING ON VALUES OF THE ENVIRONMENT AND THE CARRYING CAPACITY)
!       PAR(11) = OUTPUT FLAG F1
!       PAR(12) = ITERATION COUNTER
!       PAR(13) = REPETITION COUNTER
!
!       [RK : THE VALUES ASSIGNED TO F1 CONTROL THE TYPE OF OUTPUT
!       =0 NO OUTPUT
!       =1 OUTPUT PAR AND GROUP TOTALS AT END OF EACH REPETITION
!       =2 OUTPUT PAR AND AGE TOTALS AT THE END OF EACH REPETITION
!       =3 OUTPUT PAR AND ALL OF POP AT THE END OF EACH REPETITION
!       =4 OUTPUT PAR AND GROUP TOTALS AT THE END OF EACH ITERATION AND REPETITION
!       =5 OUTPUT PAR AND AGE TOTALS AT THE END OF EACH ITERATION AND REPETITION
!       =6 OUTPUT PAR AND ALL OF POP AT END OF EACH ITERATION AND REPETITION]
!
```

```
!       2/ PROBS (I) ARE THE PROBABILITIES TO SURVIVE IN 7 DIFFERENT AGE CATEGORIES :
!       BETWEEN 0 AND 1, 2 AND 5, 6 AND 10, 11 AND 15, 16 AND 45, 46 AND 55, 56 AND MXAGE
!       THE FOLLOWING THREE ARRAYS ARE PROBABILITIES DEPENDENT ON ENVIRONMENTAL CONDITIONS
!       PRSUR(I,J) = PROBABILITY OF SURVIVAL OF INDIVIDUALS OF AGE I AND CLASS J
!       PRMAT(I,1) = PROBABILITY OF IMATURE FEMALES OF AGE I BEING MATURE AT AGE I+1
!       PRCON(I) = PROBABILITY OF CONCEPTION FOR NONPREGNANT FEMALES OF EACH AGE I
!
!       3/ CALVIN : CALVING INTERVAL
!       SEXRATIO : THE SEX RATIO AT BIRTH (PERCENTAGE OF FEMALES)
!
!       4/ POP(I,J) IS THE TWO DIMENSIONAL ARRAY WHICH TRACKS THE NUMBERS OF INDIVIDUALS
!       IN EACH AGE/ITERATION PERIOD (ROWS),EACH FUNCTIONAL GROUP (COLUMNS) AND TOTALS
!
!       POP(I,1) = MALES
!       POP(I,2) = IMATURE FEMALES
!       POP(I,3) = NON-PREGNANT, MATURE FEMALES
!       POP(I,4...3+MXPREG) = PREGNENT FEMALES IN THE JTH "ITERATION
!       INTERVAL" OF PREGNANCY
!       POP(I,4+MXPREG...3+MXPREG+MXLACT) = LACTATING FEMALES IN THE
!       JTH "ITERATION INTERVAL" OF LACTATION
!       POP(I,4+MXPREG+MXLACT) = TOTAL POPULATION AT AGE I (ALL FUNCTIONAL GROUPS)
!       POP(MXAGE+1,J) = TOTAL POPULATION IN CLASS J FOR ALL AGES
!       POP(MXAGE+1,4+MXPREG+MXLACT) = TOTAL SIZE OF POPULATION
!       (ALL AGES ALL FUNCTIONAL GROUPS)
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!       MAIN SETUP, DATA READING AND CALLING SECTION
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        IMPLICIT NONE
        INTEGER PAR(13),NREP,ITER,MXAGE,MXPREG,MXLACT,OPTION,F1
        INTEGER NBCOL,NBLIN,MINMAT,MEANMAT,MENOAGE,CARRYING
        INTEGER :: IREP,ICNT,I
        INTEGER,ALLOCATABLE :: POPINIT(:,:),POP(:,:),TOTALS(:,:)
        REAL,ALLOCATABLE :: PRSUR(:,:),PRMAT(:),PRCON(:),COND(:)
        REAL PROBS(7)
        REAL SEXRATIO,CALVIN

                                !Definition of the interface for the subroutines
                                !using allocatable arrays
        INTERFACE
        SUBROUTINE OUTPT(PAR,POP)
        INTEGER PAR(13)
        INTEGER POP(:,:)
        END SUBROUTINE

        SUBROUTINE GETPOP(POPINIT)
        INTEGER POPINIT(:,:)
        END SUBROUTINE

        SUBROUTINE SURVIV(PAR,POP,PRSUR,SEXRATIO)
        INTEGER PAR(13)
        INTEGER POP(:,:)
        REAL PRSUR(:,:),SEXRATIO
        END SUBROUTINE

        SUBROUTINE AGE(POP)
        INTEGER POP(:,:)
        END SUBROUTINE
```

```
SUBROUTINE MATURE(POP,PRMAT)
INTEGER POP(:,:)
REAL PRMAT(:)
END SUBROUTINE

SUBROUTINE STATS(POP,PAR,TOTALS)
INTEGER POP(:,:),TOTALS(:,:),PAR(13)
END SUBROUTINE

SUBROUTINE ENVIRN(PAR,PRSUR,PROBS,PRMAT,PRCON,CALVIN,SIZEPOP,
1    CONDITION)
INTEGER PAR(13),SIZEPOP
REAL PRSUR(:,:),PROBS(7),PRMAT(:),PRCON(:),CONDITION,CALVIN
END SUBROUTINE

SUBROUTINE REPORT(PAR,POP,TOTALS)
INTEGER POP(:,:),PAR(13),TOTALS(:,:)
END SUBROUTINE

SUBROUTINE BIRTH(PAR,POP,SEXRATIO)
INTEGER POP(:,:),PAR(13)
REAL SEXRATIO
END SUBROUTINE

SUBROUTINE CONCEP(POP,PRCON)
INTEGER POP(:,:)
REAL PRCON(:)
END SUBROUTINE

END INTERFACE

                        !
                        !Processor reinitializes seed value
CALL RANDOM_SEED()

OPEN(2,FILE='Fortran\conditions.txt')
OPEN(3,FILE='Fortran\data.txt')

                        !Open Output file
OPEN(4,FILE='Fortran\results.txt')
OPEN(5,FILE='Fortran\meanfem.txt')
OPEN(6,FILE='Fortran\stdfem.txt')
OPEN(7,FILE='Fortran\meantot.txt')
OPEN(8,FILE='Fortran\stdtot.txt')
OPEN(9,FILE='Fortran\agetot.txt')

                        !1/ Read Parameters
CALL GETPAR(PAR)
NREP = PAR(1)
ITER = PAR(2)
MXAGE = PAR(3)
MXPREG = PAR(4)
MXLACT = PAR(5)
MINMAT = PAR(6)
MEANMAT = PAR(7)
MENOAGE = PAR(8)
CARRYING = PAR(9)
OPTION = PAR(10)
F1 = PAR(11)
IREP = 0
ICNT = 0
PAR(13) = IREP
```

```
      PAR(12) = ICNT

                              ! NBCOL and NBLIN are the number of lines and columns
                              ! in POPINIT (initial population matrix)
      NBCOL=4+MXPREG+MXLACT
      NBLIN=MXAGE+1

                              !Allocate arrays
      ALLOCATE(POPINIT(NBLIN,NBCOL))
      ALLOCATE(POP(NBLIN,NBCOL))
      ALLOCATE(PRSUR(NBLIN-2,NBCOL-1))
      ALLOCATE(PRMAT(NBLIN-1))
      ALLOCATE(PRCON(NBLIN-1))
      ALLOCATE(COND(ITER))

                              !Totals is the array containing the data with which we
                              !will do statistics in REPORT
      IF(F1.EQ.1 .OR. F1.EQ.3) ALLOCATE(TOTALS(NREP,NBCOL))
      IF(F1.EQ.2) ALLOCATE(TOTALS(NBLIN,NREP))
      IF(F1.EQ.4 .OR. F1.EQ.6) ALLOCATE(TOTALS(NREP*ITER,NBCOL))
      IF(F1.EQ.5) ALLOCATE(TOTALS(NBLIN,NREP*ITER))

                              !2/ PROBS(I) are the probabilities of survival in 7 different age
                              !categories. They are used in SUBROUTINE ENVIRN.
      READ (3,801) (PROBS(I),I = 1,7)
801       FORMAT(7F7.4)

                              !3/ Read CALVIN and SEXRATIO
      READ (3,802) CALVIN,SEXRATIO
802       FORMAT(2F8.5)

                              !4/ Read POP
      CALL GETPOP(POPINIT)

                              !5/ Read Environmental conditions
      IF(OPTION.EQ.0.OR.OPTION.EQ.1) THEN
         COND=0.0
      ELSE
         READ (2,803) (COND(I),I = 1,ITER)
      END IF
803       FORMAT(<ITER>F7.4)

                              !First calculation and output before entering the loop
      CALL STATS(POPINIT,PAR,TOTALS)
      CALL OUTPT(PAR,POPINIT)

                              !!!!!!!!!!!!!!!!
                              !Principal loop!
                              !!!!!!!!!!!!!!!!

                              !Loop on repetitions
      DO IREP = 1,NREP
         POP=POPINIT
         PAR(13) = IREP
                              !Loop on iterations
         DO ICNT = 1,ITER

            PAR(12) = ICNT

            CALL ENVIRN(PAR,PRSUR,PROBS,PRMAT,PRCON,CALVIN,
     1           POP(MXAGE+1,4+MXPREG+MXLACT),COND(ICNT))
            CALL SURVIV(PAR,POP,PRSUR,SEXRATIO)
```

```
              CALL AGE(POP)

              CALL MATURE(POP,PRMAT)
              CALL BIRTH(PAR,POP,SEXRATIO)
              CALL CONCEP(POP,PRCON)

              CALL STATS(POP,PAR,TOTALS)
              CALL OUTPT(PAR,POP)
           END DO

        END DO

                                !Build the last report
        CALL REPORT(PAR,POP,TOTALS)


                                !Deallocate arrays to free memory
        DEALLOCATE(POP,PRMAT,PRCON,TOTALS,COND)


                                !Stop the program
        STOP

        END



!
!
!       SUBROUTINE OUTPT
!       This subroutine writes program results in results.txt
!
        SUBROUTINE OUTPT(PAR,POP)
        IMPLICIT NONE
        INTEGER PAR(13),POP(:,:),NBCOL,NBLIN
        INTEGER ITER,IREP,ICNT,I,J,F1,MXPREG,MXLACT
!
        ITER = PAR(2)
        MXPREG=PAR(4)
        MXLACT=PAR(5)
        F1 = PAR(11)
        ICNT = PAR(12)
        IREP = PAR(13)

        NBCOL=SIZE(POP,DIM=2)
        NBLIN=SIZE(POP,DIM=1)

                                !
                                ! Report done before entering the loop

        IF (IREP.EQ.0.AND.ICNT.EQ.0) THEN

           WRITE(4,*) ’***************************************’
           WRITE(4,*) ’*ELEPHANT POPULATION MODEL VERSION 1.0*’
           WRITE(4,*) ’***************************************’
           WRITE(4,*) ’’

                                !Parameters
           WRITE(4,*) ’ ************’
           WRITE(4,*) ’ PARAMETERS :’
           WRITE(4,*) ’ ************’
           WRITE(4,*) ’’
```

```
          WRITE(4,*) '  NREP ITER MXA  MXP  MXL  MAT  F1 '
          WRITE(4,*) '  ---------------------------------'
          WRITE(4,8000) (PAR(J),J=1,5),PAR(7),PAR(11)
          WRITE(4,*) ''
          WRITE(4,*) '**************************'
          WRITE(4,*) 'RESULTS OF THE SIMULATION :'
          WRITE(4,*) '**************************'
          WRITE(4,*) ''
                               !Population group totals
          IF (F1.EQ.1.OR.F1.EQ.4) THEN
             WRITE(4,*) 'Group Totals :'
             WRITE(4,*) '--------------'
             WRITE(4,8001) '(IREP,ICNT)','MAL','IMM','NPG','PG',
       1           'LAC','TOT'
             WRITE(4,*) ('-',I=1,11+4*(4+MXPREG+MXLACT))
             WRITE (4,8004) '(',IREP,',',ICNT,')',
       1           (POP(NBLIN,J),J=1,NBCOL)
                               !Population age totals
          ELSE IF (F1.EQ.2.OR.F1.EQ.5) THEN
             WRITE(4,*) 'Age Totals :'
             WRITE(4,*) '------------'
             WRITE (4,8003) '(IREP,ICNT)',(I,I=1,NBLIN)
             WRITE(4,8010) ('-', I=1,11+4*NBLIN)
             WRITE (4,8005) '(',IREP,',',ICNT,')',
       1           (POP(I,NBCOL),I=1,NBLIN)
                               !All the population
          ELSE IF (F1.EQ.3.OR.F1.EQ.6) THEN
             WRITE(4,*) 'Population matrix'
             WRITE(4,*) '-----------------'
             WRITE (4,8002) '(',IREP,',',ICNT,')'
             WRITE (4,8006)(I,(POP(I,J),J=1,NBCOL),I=1,NBLIN)
          END IF
                               ! Other reports
       ELSE
                               !If the user wants a report at the end of each iteration
          IF (F1.EQ.4) WRITE (4,8004) '(',IREP,',',ICNT,')',
       1        (POP(NBLIN,J),J=1,NBCOL)
          IF (F1.EQ.5) WRITE (4,8005) '(',IREP,',',ICNT,')',
       1        (POP(I,NBCOL),I=1,NBLIN)
          IF (F1.EQ.6) WRITE (4,8002) '(',IREP,',',ICNT,')'
          IF (F1.EQ.6) WRITE (4,8006) (I,(POP(I,J),J=1,NBCOL),I=1,NBLIN)
          IF (F1.EQ.6) WRITE (4,*) ''

                               ! If the user wants a report at the
                               ! end of each repetition (ie F1=1,2 or 3)
          IF (ICNT.EQ.ITER) THEN
             WRITE(4,*) ''
             IF (F1.EQ.1) THEN
                WRITE (4,8004) '(',IREP,',',ICNT,')',
       1              (POP(NBLIN,J),J=1,NBCOL)
             ELSE IF (F1.EQ.2) THEN
                WRITE (4,8005) '(',IREP,',',ICNT,')',
       1              (POP(I,NBCOL),I=1,NBLIN)
             ELSE IF (F1.EQ.3) THEN
                WRITE (4,8002) '(',IREP,',',ICNT,')'
                WRITE (4,8006)(I,(POP(I,J),J=1,NBCOL),I=1,NBLIN)
                WRITE (4,*) ''
             END IF
          END IF
       END IF

8000        FORMAT(7I5)
```

```
 8001       FORMAT(A11,A4,A4,A4,A4,A<MXPREG*4>,A<MXLACT*4>)
 8002       FORMAT(A,I4,A,I4,A)
 8003       FORMAT(A,<NBLIN>I4)
 8004       FORMAT(A,I4,A,I4,A,<NBCOL>I4)
 8005       FORMAT(A,I4,A,I4,A,<NBLIN>I4)
 8006       FORMAT(I4,<NBCOL>I4)
 8010       FORMAT(<NBLIN*4+11>A)

        END SUBROUTINE OUTPT
!
!
!       SUBROUTINE GETPAR
!       This subroutine reads the main program parameter file
!
        SUBROUTINE GETPAR(PAR)
        IMPLICIT NONE
        INTEGER PAR(13),JJ
!
        READ (3,1000)(PAR(JJ),JJ=1,11)
 1000       FORMAT(11I4)


        END SUBROUTINE GETPAR
!
!
!       SUBROUTINE GETPOP
!       This subroutine reads the initial population file
!
        SUBROUTINE GETPOP(POPINIT)
        IMPLICIT NONE
        INTEGER POPINIT(:,:),I,J,NBLIN,NBCOL

        NBCOL=SIZE(POPINIT,DIM=2)
        NBLIN=SIZE(POPINIT,DIM=1)

        DO I=1,NBLIN
           DO J=1,NBCOL
              POPINIT(I,J)=0
           END DO
        END DO

        DO I = 1,NBLIN-1
           READ (3,1000)(POPINIT(I,J),J=1,NBCOL-1)
        END DO

 1000       FORMAT(<NBCOL-1>I2)

!
        END SUBROUTINE GETPOP
!
!
!       SUBROUTINE ENVIRN
!       This subroutine sets various probability values for survival
!       maturity and conception based on the environment and the current
!       population
!
        SUBROUTINE ENVIRN(PAR,PRSUR,PROBS,PRMAT,PRCON,CALVIN,SIZEPOP,
        1    CONDITION)
        IMPLICIT NONE
        INTEGER PAR(13),I,J,MINMAT,MEANMAT,MXLACT,MXPREG,MXAGE,MENOAGE
        INTEGER NBLIN,NBCOL,CARRYING,SIZEPOP,OPTION
        REAL PRMAT(:),PRCON(:),PRSUR(:,:),PROBS(7),CALVIN,CONDITION
        REAL BESTPROBS(7),WORSTPROBS(7),BESTCALVIN,WORSTCALVIN,CALVINUSED
        INTEGER BESTMENOAGE,WORSTMENOAGE
        INTEGER BESTMINMAT,WORSTMINMAT,BESTMEANMAT,WORSTMEANMAT
        INTEGER MENOAGEUSED,MINMATUSED,MEANMATUSED

                                !Definition of the interface for the function INTERPOL
        INTERFACE
           REAL FUNCTION INTERPOL(BEST,USER,WORST,CONDITION)
           REAL BEST,USER,WORST,CONDITION
           END FUNCTION
        END INTERFACE
        MXAGE = PAR(3)
        MXPREG = PAR(4)
        MXLACT = PAR(5)
        MINMAT = PAR(6)
        MEANMAT = PAR(7)
        MENOAGE = PAR(8)
        CARRYING = PAR(9)
        OPTION = PAR(10)

        NBCOL=4+MXPREG+MXLACT
        NBLIN=MXAGE+1

                                !
                                !BEST AND WORST PARAMETERS
                                !
                                ! Probabilities of surviving
        BESTPROBS=(/ 1.0,1.0,1.0,1.0,1.0,1.0,1.0 /)
        WORSTPROBS=(/ 0.7,0.8,0.9,0.96,0.96,0.6,0.0 /)
                                ! Ages
        BESTMINMAT=7
        BESTMEANMAT=7
        WORSTMINMAT=12
        WORSTMEANMAT=18
        BESTMENOAGE=55
        WORSTMENOAGE=45

                                ! Calving interval
        BESTCALVIN=REAL(MXPREG+MXLACT)
        WORSTCALVIN=15.0
                                !
                                !If we have to take into account carrying capacity
        IF(OPTION.EQ.3.OR.OPTION.EQ.5) THEN
                                !If the population size is greater than carrying capacity
           IF(SIZEPOP.GE.CARRYING) THEN
                                !Good conditions do not affect population
              IF(CONDITION>0) CONDITION=0
                                !If the population size is lower than the carrying capacity
           ELSE
                                !Bad conditions do not affect population
              IF(CONDITION<0) CONDITION=0
           END IF
        END IF
                                !
                                !Fill PRSUR
                                !
        DO J = 1,NBCOL-1

           PRSUR(1,J) = INTERPOL(BESTPROBS(1),PROBS(1),
        1        WORSTPROBS(1),CONDITION)

           DO I = 2,5
```

```
              PRSUR(I,J) = INTERPOL(BESTPROBS(2),PROBS(2),
        1           WORSTPROBS(2),CONDITION)
           END DO

           DO I = 6,10
              PRSUR(I,J) = INTERPOL(BESTPROBS(3),PROBS(3),
        1           WORSTPROBS(3),CONDITION)
           END DO

           DO I = 11,15
              PRSUR(I,J) = INTERPOL(BESTPROBS(4),PROBS(4),
        1           WORSTPROBS(4),CONDITION)
           END DO

           DO I = 16,45
              PRSUR(I,J) = INTERPOL(BESTPROBS(5),PROBS(5),
        1           WORSTPROBS(5),CONDITION)
           END DO

           DO I = 46,55
              PRSUR(I,J) = INTERPOL(BESTPROBS(6),PROBS(6),
        1           WORSTPROBS(6),CONDITION)
           END DO

           DO I = 56,NBLIN-1
              PRSUR(I,J) = INTERPOL(BESTPROBS(7),PROBS(7),
        1           WORSTPROBS(7),CONDITION)
           END DO
        END DO

                                !Fill PRMAT
                                !
        MINMATUSED=NINT(INTERPOL(REAL(BESTMINMAT),REAL(MINMAT),
        1    REAL(WORSTMINMAT),CONDITION))
        MEANMATUSED=NINT(INTERPOL(REAL(BESTMEANMAT),REAL(MEANMAT),
        1    REAL(WORSTMEANMAT),CONDITION))
           DO I=1,NBLIN-1
           PRMAT(I) = 1.0/(MEANMATUSED-MINMATUSED+1)
        END DO

        DO I = 1,MINMATUSED-1
           PRMAT(I) = 0
        END DO

                                !Fill PRCON
                                !
        MENOAGEUSED=NINT(INTERPOL(REAL(BESTMENOAGE),REAL(MENOAGE),
        1          REAL(WORSTMENOAGE),CONDITION))

        CALVINUSED=INTERPOL(BESTCALVIN,CALVIN,WORSTCALVIN,CONDITION)

        DO I = 1,MENOAGEUSED
           PRCON(I) = 1.0/(CALVINUSED-MXPREG-MXLACT+1)
        END DO
!
        END SUBROUTINE ENVIRN
!
!
!       SUBROUTINE SURVIV
!       This subroutine selects the elephants who will survive
!
        SUBROUTINE SURVIV(PAR,POP,PRSUR,SEXRATIO)
        IMPLICIT NONE
        INTEGER POP(:,:),PAR(13),ICTOT,IC ,I,J,K,OPTION
        INTEGER JMINL,JMAXL,NBLIN,NBCOL,MXLACT,MXPREG
        REAL PRSUR(:,:),X,SEXRATIO

                                !Definition of the interface for the subroutine SURSUB
        INTERFACE
           SUBROUTINE SURSUB(POP,ICTOT,J,MXPREG,SEXRATIO)
           INTEGER POP(:,:),ICTOT,J,MXPREG
           REAL SEXRATIO
           END SUBROUTINE
        END INTERFACE
!
        MXPREG = PAR(4)
        MXLACT = PAR(5)
        OPTION = PAR(10)
        NBCOL = SIZE(POP,DIM=2)
        NBLIN = SIZE(POP,DIM=1)
                                !Column boundaries in which females are lactating
        JMINL = 4+MXPREG
        JMAXL = 3+MXPREG+MXLACT

                                !We look at each column (ie. category)
        DO J = 1,NBCOL-1
           ICTOT = 0
                                !We look at each line (ie. age)
           DO I = 1,NBLIN-1
              IC=0
              IF (POP(I,J).NE.0) THEN
                                !We look at each elephant to see if he dies
                 DO K = 1,POP(I,J)
                    CALL RANDOM_NUMBER(X)
                    IF (X.GT.PRSUR(I,J)) IC = IC + 1
                 END DO
                                !We reduce the population of age I and category J
                 POP(I,J) = POP(I,J) - IC
              END IF
                                !We count the number of dead elephants in the category J
              ICTOT = ICTOT + IC
           END DO

           IF (OPTION.EQ.1.OR.OPTION.EQ.4.OR.OPTION.EQ.5) THEN
                                !  If females are lactating, we must kill calves too
              IF (ICTOT.GT.0.AND.(J.GE.JMINL.AND.J.LE.JMAXL)) THEN
                 CALL SURSUB(POP,ICTOT,J,MXPREG,SEXRATIO)
              END IF
           END IF

        END DO

        END SUBROUTINE SURVIV
!
!
!       SUBROUTINE AGE
!       This subroutine increases the age of each elephant
!
        SUBROUTINE AGE(POP)
        IMPLICIT NONE
        INTEGER POP(:,:),I,J,NBLIN,NBCOL

        NBCOL = SIZE(POP,DIM=2)
        NBLIN = SIZE(POP,DIM=1)
        DO J = 1,NBCOL-1
           DO I = 2,NBLIN-1
              POP(NBLIN-I+1,J) = POP(NBLIN-I,J)
           END DO
           POP(1,J) = 0
        END DO

        END SUBROUTINE AGE
!
!
!       SUBROUTINE MATURE
!       This subroutine causes the imature members to become mature,
!       based on the probability PRMAT (provided by ENVIRN)
!
        SUBROUTINE MATURE(POP,PRMAT)
        IMPLICIT NONE
        INTEGER POP(:,:),IMM,NPG,I,IC,K,NBLIN,NBCOL
        REAL PRMAT(:),X
!
        NBCOL = SIZE(POP,DIM=2)
        NBLIN = SIZE(POP,DIM=1)

        DO I = 1,NBLIN-1
           IC = 0
           IMM = POP(I,2)
           NPG = POP(I,3)
           IF (IMM.NE.0) THEN
              DO K = 1,IMM
                 CALL RANDOM_NUMBER(X)
                 IF (X.LE.PRMAT(I)) IC = IC + 1
              END DO
           END IF
           NPG = NPG + IC
           IMM = IMM - IC
           POP(I,2) = IMM
           POP(I,3) = NPG
        END DO

        END SUBROUTINE MATURE
!
!
!       SUBROUTINE BIRTH
!       This subroutine causes several population changes to occur:
!       New members are born
!       Once pregnant, females become lactating
!       Once lactating, females become receptive members
!
        SUBROUTINE BIRTH(PAR,POP,SEXRATIO)
        INTEGER POP(:,:),PAR(13),MXAGE,MXPREG,MXLACT
        INTEGER JMINL,JMAXL,JMINP,JMAXP,NBMAL,NBFEM,NBORN,NPG,I,J,K
        REAL SEXRATIO
!
        MXAGE = PAR(3)
        MXPREG = PAR(4)
        MXLACT = PAR(5)
                                !Column boundaries in which females are pregnant
        JMINP = 4
        JMAXP = MXPREG + 3
                                !Column boundaries in which females are lactating
        JMINL = 4+MXPREG
        JMAXL = 3+MXPREG+MXLACT
```

```
                                !Number of new born (males and females)
        NBMAL = 0
        NBFEM = 0

        DO I = 1,MXAGE
           NPG = POP(I,3)
           NBORN = POP(I,JMAXP)
           IF (NBORN.NE.0) THEN

              DO K = 1,NBORN
                 CALL RANDOM_NUMBER(X)
                 IF (X.LT.SEXRATIO) THEN
                    NBFEM = NBFEM + 1
                 ELSE
                    NBMAL = NBMAL + 1
                 END IF
              END DO

           END IF
                                !
                                !Pregnant females evolve in the column J+1
        IF (JMAXP.NE.JMINP) THEN
           DO J = JMAXP,JMINP + 1,-1
              POP(I,J) = POP(I,J-1)
           END DO
        END IF
        POP(I,JMINP) = 0

                                !Once lactating, females become receptive members
        NPG = NPG + POP(I,JMAXL)

                                !lactating females evolve in the column J+1
        IF (JMAXL.NE.JMINL) THEN
           DO J = JMAXL,JMINL+1,-1
              POP(I,J) = POP(I,J-1)
           END DO
        END IF
                                !
                                !Once pregnant, females become lactating
        POP(I,JMINL) = NBORN
        POP(I,3) = NPG

        END DO

        POP(1,1) = NBMAL
        POP(1,2) = NBFEM

        END SUBROUTINE BIRTH
!
!
!       SUBROUTINE CONCEP
!       This subroutine causes conception of non pregnant females according
!       to the probabilities contained in PRCON (provided by EVIRN)
!
        SUBROUTINE CONCEP(POP,PRCON)
        IMPLICIT NONE
        INTEGER POP(:,:),MXAGE,NPG,NCON,I,K
        REAL PRCON(:),X

        MXAGE = SIZE(POP,DIM=1)-1
        DO I = 1,MXAGE
                                !Number of non pregnant females
           NPG = POP(I,3)
                                !Number of conceptions
           NCON = 0
           IF (NPG.NE.0) THEN
              DO K = 1,NPG
                 CALL RANDOM_NUMBER(X)
                 IF (X.LE.PRCON(I)) NCON = NCON + 1
              END DO
           END IF
           NPG = NPG - NCON

           POP(I,3) = NPG
           POP(I,4) = NCON
        END DO

        END SUBROUTINE CONCEP
!
!
!       SUBROUTINE STATS
!       This subroutine only calculates totals.
!
        SUBROUTINE STATS(POP,PAR,TOTALS)
        IMPLICIT NONE
        INTEGER POP(:,:),PAR(13),NBLIN,NBCOL,TOTALS(:,:),IREP,ICNT,F1,ITER

        NBLIN=SIZE(POP,DIM=1)
        NBCOL=SIZE(POP,DIM=2)
        F1=PAR(11)
        ITER=PAR(2)
        IREP=PAR(13)
        ICNT=PAR(12)

                                ! We calculate age totals
        POP(:,NBCOL) = SUM(POP(:,1:NBCOL-1),DIM=2)
                                ! We calculate group totals
        POP(NBLIN,:) = SUM(POP(1:NBLIN-1,:),DIM=1)
                                ! We calculate the population total
        POP(NBLIN,NBCOL) = SUM(POP(NBLIN,1:NBCOL-1))

                                ! We store the results that we we will need for statistics
                                ! in TOTALS
        IF(ICNT.EQ.ITER) THEN
           IF(F1.EQ.1 .OR. F1.EQ.3) TOTALS(IREP,:)=POP(NBLIN,:)
           IF(F1.EQ.2)        TOTALS(:,IREP)=POP(:,NBCOL)
        END IF
        IF(ICNT.NE.0 .OR. IREP.NE.0) THEN
           IF(F1.EQ.4 .OR. F1.EQ.6) TOTALS(ITER*(IREP-1)+ICNT,:)
        1         =POP(NBLIN,:)
           IF(F1.EQ.5)        TOTALS(:,ITER*(IREP-1)+ICNT)=POP(:,NBCOL)
        END IF

        END SUBROUTINE STATS
!
!
!       SUBROUTINE REPORT
!       This subroutine calculates means and standard deviations and
!       produces a final report
!       Rk : If F1=4, calculates mean and standard deviation of the female population
!       to use in Matlab
!
        SUBROUTINE REPORT(PAR,POP,TOTALS)
        IMPLICIT NONE
```

```
INTEGER PAR(13),POP(:,:),TOTALS(:,:)
INTEGER MXPREG,MXLACT,NBCOL,NBLIN,F1,I,II,J,ITER,NREP
REAL,ALLOCATABLE :: MEAN(:,:),STD(:,:)
REAL stdd
REAL,ALLOCATABLE :: MEANFEM(:),STDFEM(:)

NBCOL=SIZE(POP,DIM=2)
NBLIN=SIZE(POP,DIM=1)
ITER=PAR(2)
NREP=PAR(1)
MXPREG=PAR(4)
MXLACT=PAR(5)
F1=PAR(11)
                        !MEANS of totals
WRITE(4,*) ’’
WRITE(4,*) ’ MEANS :’
WRITE(4,*) ’ -------’
                        !Mean of group totals
IF(F1.EQ.1 .OR. F1.EQ.3) THEN
   ALLOCATE(MEAN(1,NBCOL))
   ALLOCATE(STD(1,NBCOL))
   MEAN(1,:)=REAL(SUM(TOTALS,DIM=1))/NREP
   WRITE(4,650) ’MAL’,’IMM’,’NPG’,’PG’,’LAC’,’TOT’
   WRITE(4,600) (MEAN(1,I),I=1,NBCOL)

                        !Mean of age totals
ELSE IF (F1.EQ.2) THEN
   ALLOCATE(MEAN(NBLIN,1))
   ALLOCATE(STD(NBLIN,1))
   WRITE (4,651) (I,I=1,NBLIN)
   MEAN(:,1)=REAL(SUM(TOTALS,DIM=2))/NREP
   WRITE(4,601) (MEAN(I,1),I=1,NBLIN)

ELSE IF(F1.EQ.4 .OR. F1.EQ.6) THEN
   ALLOCATE(MEAN(ITER,NBCOL))
   ALLOCATE(STD(ITER,NBCOL))
   ALLOCATE(MEANFEM(ITER))
   ALLOCATE(STDFEM(ITER))

   WRITE(4,652) ’ITER’,’MAL’,’IMM’,’NPG’,’PG’,’LAC’,’TOT’
   DO I=1,ITER
      IF (NREP.EQ.1) THEN
         MEAN(I,:)=
1              REAL(TOTALS(I,:))
         WRITE(4,602) I,(MEAN(I,J),J=1,NBCOL)
                        !!To compare with Leslie Method in Matlab
                        !! Mean of the female population
         MEANFEM(I)=SUM(MEAN(I,2:NBCOL-1))
         WRITE(5,24) (MEANFEM(I))
         WRITE(7,24) (MEAN(I,NBCOL))
      ELSE
         MEAN(I,:)=
1              REAL(SUM(TOTALS(I:(NREP-1)*ITER+I:ITER,:),DIM=1))/NREP
         WRITE(4,602) I,(MEAN(I,J),J=1,NBCOL)
                        !!To compare with Leslie Method in Matlab
                        !! Mean of the female population
         MEANFEM(I)=SUM(MEAN(I,2:NBCOL-1))
         WRITE(5,24) (MEANFEM(I))
         WRITE(7,24) (MEAN(I,NBCOL))
                        !!
      END IF
   END DO
```

```
       ELSE IF(F1.EQ.5) THEN
          ALLOCATE(MEAN(NBLIN,ITER))
          ALLOCATE(STD(NBLIN,ITER))
          IF (NREP.EQ.1) THEN
             DO I=1,ITER
                MEAN(:,I)=
       1              REAL(TOTALS(:,1))
             END DO
             WRITE(4,606) ('-',I=1,6*ITER+9)
             WRITE(4,603) 'AGE\ITER',(J,J=1,ITER)
             WRITE(4,606) ('-',I=1,6*ITER+9)
             WRITE(4,605) (I,(MEAN(I,J),J=1,ITER),I=1,NBLIN)
                               !!To use in Matlab
             WRITE(9,607) ((MEAN(I,J),J=1,ITER),I=1,NBLIN-1)
          ELSE
             DO I=1,ITER
                MEAN(:,I)=
       1              REAL(SUM(TOTALS(:,I:(NREP-1)*ITER+I:ITER),DIM=2))/NREP
             END DO
             WRITE(4,606) ('-',I=1,6*ITER+9)
             WRITE(4,603) 'AGE\ITER',(J,J=1,ITER)
             WRITE(4,606) ('-',I=1,6*ITER+9)
             WRITE(4,605) (I,(MEAN(I,J),J=1,ITER),I=1,NBLIN)
                               !!To use in Matlab
             WRITE(9,607) ((MEAN(I,J),J=1,ITER),I=1,NBLIN-1)
          END IF
       END IF

600        FORMAT(<NBCOL>F6.2)
601        FORMAT(<NBLIN>F6.2)
602        FORMAT(I4,<NBCOL>F6.2)
24        FORMAT(F8.2)
603        FORMAT(A,<ITER>I6)
605        FORMAT(I9,<ITER>F6.2)
606        FORMAT(<6*ITER+9>A)
607        FORMAT(<ITER>F10.2)

652        FORMAT(A4,A6,A6,A6,A6,A<MXPREG*6>,A<MXLACT*6>)
651        FORMAT(<NBLIN>I6)
650        FORMAT(A6,A6,A6,A6,A<MXPREG*6>,A<MXLACT*6>)

                               !
                               !STANDARD DEVIATION of totals
       WRITE(4,*) ''
       WRITE(4,*) ' STANDARD DEVIATION :'
       WRITE(4,*) ' --------------------'

                               !Standard deviation of group totals
       IF(F1.EQ.1 .OR. F1.EQ.3) THEN
          IF (NREP.EQ.1) THEN
             WRITE(4,750) 'MAL','IMM','NPG','PG','LAC','TOT'
             WRITE(4,700) (0,I=1,NBCOL)
          ELSE
             DO J=1,NBCOL
                stdd=0.
                DO I=1,NREP
                   stdd=stdd+(MEAN(1,J)-TOTALS(I,J))**2
                END DO
                STD(1,J)=SQRT(stdd/(NREP-1))
             END DO
             WRITE(4,750) 'MAL','IMM','NPG','PG','LAC','TOT'
```

```
      WRITE(4,700) (STD(1,I),I=1,NBCOL)
   END IF

                        !Standard deviation of age totals
ELSE IF (F1.EQ.2) THEN
   WRITE (4,751) (I,I=1,NBLIN)
   IF (NREP.EQ.1) THEN
      WRITE(4,701) (0.0,I=1,NBLIN)
   ELSE
      DO I=1,NBLIN
         stdd=0.
         DO J=1,NREP
            stdd=stdd+(MEAN(I,1)-TOTALS(I,J))**2
         END DO
         STD(I,1)=SQRT(stdd/(NREP-1))
      END DO
      WRITE(4,701) (STD(I,1),I=1,NBLIN)
   END IF

ELSE IF(F1.EQ.4 .OR. F1.EQ.6) THEN
   WRITE(4,752) 'ITER','MAL','IMM','NPG','PG','LAC','TOT'
   IF (NREP.EQ.1) THEN
      DO I=1,ITER
         WRITE(4,702) I,(0.,J=1,NBCOL)
                        !! To compare with Leslie Method in Matlab
         WRITE(6,24) 0.
         WRITE(8,24) 0.
                        !!
      END DO
   ELSE

      DO J=1,NBCOL
         DO I=1,ITER
            stdd=0.
            DO II=1,NREP
               stdd=stdd+(MEAN(I,J)-TOTALS(I+ITER*(II-1),J))**2
            END DO
            STD(I,J)=SQRT(stdd/(NREP-1))
         END DO
      END DO
                        !! Standard deviation of the female population (for Matlab)
      DO I=1,ITER
         STDFEM(I)=0.
         DO II=1,NREP
            STDFEM(I)=STDFEM(I)+(MEANFEM(I)-
1                 SUM(TOTALS(I+ITER*(II-1),2:NBCOL-1)))**2
         END DO
         STDFEM(I)=SQRT(STDFEM(I)/(NREP-1))
      END DO
                        !!

      DO I=1,ITER
         WRITE(4,702) I,(STD(I,J),J=1,NBCOL)
                        !! To compare with Leslie Method in Matlab
         WRITE(6,24) (STDFEM(I))
         WRITE(8,24) (STD(I,NBCOL))
                        !!
      END DO
   END IF

ELSE IF(F1.EQ.5) THEN
   DO I=1,NBLIN
```

```
              DO J=1,ITER
                 stdd=0.
                 DO II=1,NREP
                    stdd=stdd+(MEAN(I,J)-TOTALS(I,J+ITER*(II-1)))**2
                 END DO
                 IF (NREP.EQ.1) THEN
                    STD(I,J)=0.0
                 ELSE
                    STD(I,J)=SQRT(stdd/(NREP-1))
                 END IF
              END DO
           END DO
           WRITE(4,706) ('-',I=1,6*ITER+9)
           WRITE(4,703) 'AGE\ITER',(J,J=1,ITER)
           WRITE(4,706) ('-',I=1,6*ITER+9)
           WRITE(4,705) (I,(STD(I,J),J=1,ITER),I=1,NBLIN)
        END IF

 700        FORMAT(<NBCOL>F6.2)
 701        FORMAT(<NBLIN>F6.2)
 702        FORMAT(I4,<NBCOL>F6.2)
 703        FORMAT(A,<ITER>I6)
 705        FORMAT(I9,<ITER>F6.2)
 706        FORMAT(<6*ITER+9>A)
 752        FORMAT(A4,A6,A6,A6,A6,A<MXPREG*6>,A<MXLACT*6>)
 751        FORMAT(<NBLIN>I6)
 750        FORMAT(A6,A6,A6,A6,A<MXPREG*6>,A<MXLACT*6>)


                                ! Final Population Matrix
        WRITE (4,*) ''
        WRITE (4,*) '************************'
        WRITE (4,*) 'FINAL POPULATION MATRIX :'
        WRITE (4,*) '************************'
        WRITE (4,*) ''
        WRITE (4,903) 'AGE','MAL','IMM','NPG','PG','LAC','TOT'
        WRITE (4,902) (I,(POP(I,J),J=1,NBCOL),I=1,NBLIN)

 902        FORMAT(<NBCOL+1>I4)
 903        FORMAT(A4,A4,A4,A4,A4,A<MXPREG*4>,A<MXLACT*4>)

        DEALLOCATE(MEAN,STD,MEANFEM,STDFEM)

        END SUBROUTINE REPORT
!
!
!       SUBROUTINE SURSUB.
!       This subroutine kills immature members which mother died
!       ICTOT IS THE NUMBER OF LACTATING FEMALES KILLED IN THE
!       (J-3-MXPREG)TH YEAR OF LACTATION
!
        SUBROUTINE SURSUB(POP,ICTOT,J,MXPREG,SEXRATIO)
        IMPLICIT NONE
        INTEGER POP(:,:),KM,KF,ICTOT,I,J,MAL,IMM,MXPREG
        REAL X,SEXRATIO
                                !KM is the number of male calves that we must kill
                                !KF is the number of female calves that we must kill
        KM = 0
        KF = 0
                                !We kill one calf for each lactatig female killed
        DO I = 1,ICTOT
           CALL RANDOM_NUMBER(X)
```

```
            IF (X.LT.SEXRATIO) THEN
               KF = KF + 1
            ELSE
               KM = KM + 1
            END IF
         END DO
!
         I = J - 3 - MXPREG
         MAL = POP(I,1)
         IMM = POP(I,2)
         MAL = MAL - KM
         IMM = IMM - KF
         IF (MAL.LT.0) MAL = 0
         IF (IMM.LT.0) IMM = 0
         POP(I,1) = MAL
         POP(I,2) = IMM
!
         END SUBROUTINE SURSUB
!

!        FUNCTION INTERPOL
!        This function interpolate linearly between BEST and USER or USER
!        and WORST, depending on the condition

         REAL FUNCTION INTERPOL(BEST,USER,WORST,CONDITION)
         IMPLICIT NONE
         REAL BEST,USER,WORST,CONDITION

                                 !If the value entered by the user is not in the boundaries,
                                 !we change the boundaries.
         IF(USER<MIN(BEST,WORST)) THEN
            IF(MIN(BEST,WORST).EQ.BEST) THEN
               BEST=USER
            ELSE
               WORST=USER
            END IF
         ELSE IF (USER>MAX(BEST,WORST)) THEN
            IF(MAX(BEST,WORST).EQ.BEST) THEN
               BEST=USER
            ELSE
               WORST=USER
            END IF
         END IF

                                 !If the environmental condition is good
         IF (CONDITION>0.0) THEN
            INTERPOL=USER+CONDITION*(BEST-USER)

                                 !If the environmental condition is bad or normal
         ELSE
            INTERPOL=USER-CONDITION*(WORST-USER)
         END IF

         END FUNCTION INTERPOL
```

Appendix C

Hopf bifurcation (in the Caughley's model)

This appendix was built using essentially papers [22] by Hugh Murrell (1994) and [4] by K.J. Duffy et al. (1999).

Let us analyse a general system of differential equations:

$$\frac{dx}{dt} = f(x, y) \tag{C.1}$$

$$\frac{dy}{dt} = g(x, y) \tag{C.2}$$

Here the dependent variables $x = x(t)$ and $y = y(t)$ are functions of the independent variable t.

The stationary (or critical) points (x_s, y_s) can be obtained solving the equations:

$$f(x_s, y_s) = 0 \tag{C.3}$$

$$g(x_s, y_s) = 0 \tag{C.4}$$

Stationary points are fixed points of the system in that any solution starts at one of these points stays at that point for all time. The behaviours of solutions that start near a stationary point depends on the classification of the stationary point. The classification is determined by linearizing the system in the neighbourhood of the stationary point. Expanding the right-hand side of equations (C.1) and (C.2) in a Taylor series about the stationary point (x_s, y_s) results in:

$$\frac{dx}{dt} = f(x_s, y_s) + (x - x_s)f_x(x_s, y_s) + (y - y_s)f_y(x_s, y_s) + \dots \tag{C.5}$$

$$\frac{dy}{dt} = g(x_s, y_s) + (x - x_s)g_x(x_s, y_s) + (y - y_s)g_y(x_s, y_s) + \dots \tag{C.6}$$

(The x and y subscripts denote partial derivatives.) Droping the higher order terms, making use of equations (C.3) and (C.4), and translating the coordinate system so that the stationary point is at the origin, the linearized system can be written in vector notation as:

$$\begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{C.7}$$

where the matrix is the Jacobian of the system evaluated at the stationary point (x_s, y_s). The behavior of the nonlinear system near the stationary point is approximated by this linear system. It is easy to check that a particular solution is given by:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = e^{\lambda t} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{C.8}$$

where (α, β) is an eigenvector of the Jacobian with eigenvalue λ. There are, in general, two eigenvalue and two different eigenvectors. If the eigenvalues are real, the general solution to the system (C.7) is a linear combination of the two particular solutions (C.8). If the eigenvalues are complex, they (and their eigenvectors) must be complex conjugates, since the matrix is real. The general solution is then a linear combination of the real and imaginary parts of the complex solution.

The linearized system is classified according to the eigenvalues. If the real parts of the eigenvalues are negative, the solution (C.8) gives trajectories that approach the origin as time increases. For the nonlinear system, this means that the stationary point is an *attractor* of any trajectory that passes close to it. If the real parts of the eigenvalues are positive, the trajectories move away from the origin, so the stationary point is a *repellor*. If the eigenvalues are complex, the complex exponential gives trajectories that spiral toward or away from the origin. If the eigenvalues are real and of opposite sign, the trajectories approach the origin along the direction of one eigenvector and move away along the direction of the other; the stationary point is then a *saddle* point. If the eigenvalues are real and have the same sign, the stationary point is a *node*. Eigenvalues that are purely imaginary give closed trajectories surrounding the origin (a *center*). Centers and attracting nodes and spirals are *stable* in the sense that trajectories that pass close to the stationary point remain close. Saddles and repelling nodes and spirals are unstable.

Table C.1 summarizes the classification of stationary points corresponding to nonzero eigenvalues.

Let us now introduce a parameter k into the system of differential equations:

$$\frac{dx}{dt} = f(x, y, k) \tag{C.9}$$

$$\frac{dy}{dt} = g(x, y, k) \tag{C.10}$$

Type of stationary point	Real parts of eigenvalues	Imaginary parts of eigenvalues
Attracting node	negative	zero
Attracting spiral	negative	nonzero
repelling node	positive	zero
repelling spiral	positive	nonzero
saddle	opposite sign	zero
center	zero	nonzero

Table C.1: Classification of stationary points

As the parameter varies, the stationary points may move in the phase plane and change type. An interesting type of bifurcation occurs when a spiral point changes from attracting to repelling. In such a case, a limit cycle can appear or disappear as the parameter passes through the bifurcation point. This phenomenon is called a *Hopf bifurcation.*

In the case of the Caughley's model the system of differential equations is:

$$\frac{dx}{dt} = x(a - \frac{a}{K}x - \frac{cy}{x+g}) \tag{C.11}$$

$$\frac{dy}{dt} = y(-A + \frac{kx}{y+B}) \tag{C.12}$$

In [4] K.J. Duffy et al. (1999), the purpose of the analysis was to determine whether stable limit cycles, coexistence or extinction are the likely outcomes of the elephant-tree interaction. Equations (C.11) and (C.12) have an unstable equilibrium point $(0, 0)$, a stable equilibrium point $(K, 0)$ if $A > kKB^{-1}$ and the only other interesting point of equilibrium at:

$$\begin{aligned} x_0 &= \frac{1}{2}\Big(K - g - \frac{ckK}{aA} + \frac{K}{a}\sqrt{(a - \frac{ag}{K} - \frac{ck}{D})^2 + \frac{4a}{K}(ag + Bc)}\Big) \\ y_0 &= \frac{kx_0}{A} - B \end{aligned}$$

if $A \leq kKB^{-1}$. This means that if the death rate of the elephants in the absence of trees is very high, then they will become extinct, while if it is small enough, the system will tend to the point (x_0, y_0) or to a stable limit cycle around it, all depending on the specific situation determined by initial conditions. In order to determine which of these possibilities hold, after the transformation $X = x - x_0$, $Y = y - y_0$, equations (C.11) and (C.12) are linearized giving:

$$\begin{aligned}\frac{dX}{dt} &= \Big(a - \frac{2a}{K}x_0 - \frac{cgy_0}{(x_0+g)^2}\Big)X - \frac{cx_0}{x_0+g}Y \\ \frac{dY}{dt} &= \frac{ky_0}{y_0+B}X - \frac{Ay_0}{y_0+B}Y\end{aligned}$$

The Jacobian of this system has eigenvalues given by

$$\lambda = \frac{1}{2}\big((\alpha+\beta) \pm \sqrt{(\alpha+\beta)^2 - 4\psi}\big)$$

where

$$\begin{aligned}\alpha &= a - \frac{2a}{K}x_0 - cg\frac{y_0}{(x_0+g)^2} \\ \beta &= -A\frac{y_0}{y_0+B} \\ \psi &= ck\frac{x_0y_0}{(x_0+g)(y_0+B)}\end{aligned}$$

For a Hopf bifurcation to exist it is necessary that some values of the parameters give eigenvalues of the form $\lambda = \lambda_1 \pm i\lambda_2$ where $\lambda_1 \leq 0$ and $\lambda_2 > 0$. The point of bifurcation is then for that combination of parameters such that $\lambda_1 = 0$ and limit cycles occur for $\lambda_1 > 0$ and $\lambda_2 > 0$. A parameter analysis to probe where limit cycles might be possible was performed by varying each parameter value by at least an order of magnitude. Starting reference values for each parameters were the minimum and maximum values from Table 5.2 page 84. Limit cycles are only possible when the parameters k or c are much higher than the range given in this table. For example, Hopf bifurcations can (where $\lambda_1 = 0$) can be seen in Figure 5.2 page 86 with parameters given in Table 5.2.

Appendix D

Grazing herd cellular automaton code

This is the main function for the grazing herd cellular automaton found in book [7] by Richard J. Gaylord, Kazume Nishidate (1996): First the function is defined as a module. The initial predators and preys are randomly placed on the lattice called “pasture” with probability “predDensity” and “preyDensity”.

```
PredatorPrey[n_, preyDensity_, predDensity_, preg_, starve_, p_, t_]
    /; preyDensity + predDensity < 1 :=Module[{eco, RND, pasture},

RND:=Random[Integer,{1,4}];

pasture =
  Table[Floor[Random[] + (preyDensity + predDensity)] *
     Floor[1 + Random[] + predDensity/(preyDensity + predDensity)], {n}, {n}] /.
     2 :> {Random[Integer, {1, 4}], Random[Integer, {1, preg}],
        Random[Integer, {1, starve}]}];
```

Then the 53 rewrite rules are defined:

- A predator with 0 birth clock and diet clock values starve and vacates the site, leaving behind a newborn predator facing a randomly chosen direction with a birth clock value of preg and a diet clock value of starve.

  ```
  eco[{_, 0, 0}, _, _, _, _, _, _, _, _, _, _, _, _] := {RND, preg,starve};
  ```

- A predator with a 0 diet clock value and a nonzero birth clock value starves to death and vanishes, leaving an empty site.

  ```
  eco[{_, _, 0}, _, _, _, _, _, _, _, _, _, _, _, _] := 0;
  ```

- A predator with a positive diet clock value that faces an adjacent empty or food site that is faced by at least one other predator with a positive diet clock value, remains in place, randomly chooses a direction to face, decrements its diet clock by 1, and decrements its birth clock by one unless its value is 0.

```
eco[{1, a_, b_?Positive}, 0 | 1, _, _, _, {4, _, _?Positive}
   , _, _, _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{1, a_, b_?Positive}, 0 | 1, _, _, _, _, _, _,
   {2, _, _?Positive}, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{1, a_, b_?Positive}, 0 | 1, _, _, _, _, _, _, _, {3, _,
   _?Positive}, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{2, a_, b_?Positive}, _, 0 | 1, _, _, {3, _, _?Positive}
   , _, _, _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{2, a_, b_?Positive}, _, 0 | 1, _, _, _, {1, _, _?Positive}
   , _, _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{2, a_, b_?Positive}, _, 0 | 1, _, _, _, _, _, _, _,
   {4, _, _?Positive}, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{3, a_, b_?Positive}, _, _, 0 | 1, _, _, {4, _, _?Positive}
   , _, _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{3, a_, b_?Positive}, _, _, 0 | 1, _, _, _, {2, _, _?Positive},
   _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{3, a_, b_?Positive}, _, _, 0 | 1, _, _, _, _, _, _, _, {1, _,
    _?Positive}, _] := {RND, Max[0, a - 1], b - 1};
eco[{4, a_, b_?Positive}, _, _, _, 0 | 1, _, _, {1, _, _?Positive},
    _, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{4, a_, b_?Positive}, _, _, _, 0 | 1, _, _, _,
    {3, _, _?Positive}, _, _, _, _] := {RND, Max[0, a - 1], b - 1};
eco[{4, a_, b_?Positive}, _, _, _, 0 | 1, _, _, _, _, _, _, _,
    {2, _, _?Positive}] := {RND, Max[0, a - 1], b - 1};
```

- A prey site that is faced by two or more predators with positive diet clock values on adjacent sites remains unchanged.

```
eco[1, {3, _, _?Positive}, {4, _, _?Positive}, _, _, _, _, _, _,
  _, _, _, _] := 1;
eco[1, {3, _, _?Positive}, _, {1, _, _?Positive}, _, _, _, _, _,
  _, _, _, _] := 1;
eco[1, {3, _, _?Positive}, _, _, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := 1;
eco[1, _, {4, _, _?Positive}, {1, _, _?Positive}, _, _, _, _, _,
  _, _, _, _] := 1;
eco[1, _, {4, _, _?Positive}, _, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := 1;
```

```
eco[1, _, _, {1, _, _?Positive}, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := 1;
```

- An empty site that is faced by two or more predators with positive diet clock values becomes a food source with probability p

```
eco[0, {3, _, _?Positive}, {4, _, _?Positive}, _, _, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]];
eco[0, {3, _, _?Positive}, _, {1, _, _?Positive}, _, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]]
eco[0, {3, _, _?Positive}, _, _, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]];
eco[0, _, {4, _, _?Positive}, {1, _, _?Positive}, _, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]];
eco[0, _, {4, _, _?Positive}, _, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]];
eco[0, _, _, {1, _, _?Positive}, {2, _, _?Positive}, _, _, _, _,
  _, _, _, _] := Floor[p + Random[]];
```

- A predator with a 0 birth clock value and a positive diet clock value that faces an adjacent empty or prey site that is faced by no other predator with a positive diet clock value, vacates the site it is occupying, leaving behind a newborn predator facing a randomly chosen direction with a birth clock value of preg and a diet clock value of starve.

```
eco[{1, 0, _?Positive}, 0 | 1, _, _, _, _, _, _, _, _, _, _, _] :=
{RND, preg, starve};
eco[{2, 0, _?Positive}, _, 0 | 1, _, _, _, _, _, _, _, _, _, _] :=
{RND, preg, starve};
eco[{3, 0, _?Positive}, _, _, 0 | 1, _, _, _, _, _, _, _, _, _] :=
{RND, preg, starve};
eco[{4, 0, _?Positive}, _, _, _, 0 | 1, _, _, _, _, _, _, _, _] :=
{RND, preg, starve};
```

- A predator with positive birth and diet clock values that faces an adjacent empty or prey site that is faced by no other predator with a positive diet clock value, vacates the site it is occupying, leaving it empty.

```
eco[{1, _, _?Positive}, 0 | 1, _, _, _, _, _, _, _, _, _, _, _] := 0;
eco[{2, _, _?Positive}, _, 0 | 1, _, _, _, _, _, _, _, _, _, _] := 0;
eco[{3, _, _?Positive}, _, _, 0 | 1, _, _, _, _, _, _, _, _, _] := 0;
eco[{4, _, _?Positive}, _, _, _, 0 | 1, _, _, _, _, _, _, _, _] := 0;
```

- A prey site that is faced by exactly one predator with a 0 birth clock value and a positive diet clock value on an adjacent site, is occupied by the predator, which randomly chooses a direction to face, resets its birth clock to preg, and resets its diet clock to starve.

```
eco[1, {3, 0, _?Positive}, _, _, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, starve};
eco[1, _, {4, 0, _?Positive}, _, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, starve};
eco[1, _, _, {1, 0, _?Positive}, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, starve};
eco[1, _, _, _, {2, 0, _?Positive}, _, _, _, _, _, _, _, _] :=
  {RND, preg, starve};
```

- An empty site that is faced by exactly one predator with a 0 birth clock value and a positive diet clock value on an adjacent site is occupied by the predator, which randomly chooses a direction to face, decrements its diet clock by 1, and resets its birth clock to preg.

```
eco[0, {3, 0, b_?Positive}, _, _, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, b - 1};
eco[0, _, {4, 0, b_?Positive}, _, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, b - 1};
eco[0, _, _, {1, 0, b_?Positive}, _, _, _, _, _, _, _, _, _] :=
  {RND, preg, b - 1};
eco[0, _, _, _, {2, 0, b_?Positive}, _, _, _, _, _, _, _, _] :=
  {RND, preg, b - 1};
```

- A prey site that is faced by exactly one predator with positive birth and diet clock values on an adjacent site is occupied by the predator, which randomly selects a direction to face, decrements its birth clock by 1, and resets its diet clock to starve.

```
eco[1, {3, a_, _?Positive}, _, _, _, _, _, _, _, _, _, _, _] :=
  {RND, a -1, starve};
eco[1, _, {4, a_, _?Positive}, _, _, _, _, _, _, _, _, _, _] :=
  {RND, a -1, starve};
eco[1, _, _, {1, a_, _?Positive}, _, _, _, _, _, _, _, _, _] :=
  {RND, a -1, starve};
eco[1, _, _, _, {2, a_, _?Positive}, _, _, _, _, _, _, _, _] :=
  {RND, a -1, starve};
```

- An empty site that is faced by exactly one predator with positive diet and birth clock values on an adjacent site is occupied by the predator, which randomly selects a direction to face and decrements its diet and birth clocks by 1.

```
eco[0, {3, a_, b_?Positive}, _, _, _, _, _, _, _, _, _, _, _] :=
  {RND, a - 1, b - 1};
eco[0, _, {4, a_, b_?Positive}, _, _, _, _, _, _, _, _, _, _] :=
  {RND, a - 1, b - 1};
eco[0, _, _, {1, a_, b_?Positive}, _, _, _, _, _, _, _, _, _] :=
  {RND, a - 1, b - 1};
eco[0, _, _, _, {2, a_, b_?Positive}, _, _, _, _, _, _, _, _] :=
  {RND, a - 1, b - 1};
```

- Any other predator remains in place, randomly chooses a direction to face, and decrements its diet and birth time clocks by 1.

```
eco[{_, a_, b_}, _, _, _, _, _, _, _, _, _, _, _, _] := {RND, a - 1, b- 1};
```

- Any other prey site remains unchanged.

```
eco[1, _, _, _, _, _, _, _, _, _, _, _, _] := 1;
```

- Any other empty site becomes a food source with probability p.

```
eco[0, _, _, _, _, _, _, _, _, _, _, _, _] := Floor[p + Random[]];
```

The sites in the lattice are updated at each time step by applying the following anonymous function to the CA lattice:

```
MvonN[eco, #]&
```

where

```
MvonN[func__, lat_] :=
MapThread[func, Map[RotateRight[lat, #]&, {{0, 0}, {1, 0}, {0, -1},
{-1, 0}, {0, 1},{1,  -1}, {-1, -1}, {-1, 1}, {1, 1},{2,0}, {0,
-2}, {-2, 0}, {0, 2}}], 2];
```

The pasture of predators and preys evolves over t time steps using the following Nest operation.

```
evolve = NestList[MvonN[eco, #]&, pasture, t]
```

Appendix E

To cull or not to cull

Article from the Mail & Guardian (Johannesburg)
(May 30, 2003 by Fiona Macleod)

Sol Kerzner's new venture into the Southern African game lodge industry has become embroiled in a bitter battle about whether Botswana needs to cull its elephants.

One&Only, the company set up by the "sun king" and his son, Butch Kerzner, to develop "six-star" ecotourism resorts, is in negotiations involving a new lodge in the Chobe National Park in northern Botswana.

The proposed lodge will be built in an area where huge herds of free-ranging elephants congregate to drink from the Chobe river. The herds are a great tourist drawcard, but scientists argue that the elephants are hammering vegetation and reducing biodiversity in the area.

The debates are expected to heat up in the next two months after the Department of Wildlife and National Parks releases its draft elephant management policy on June 7. The final policy is due to be submitted to the Botswana government at the end of July.

Insiders are worried that One& Only's entry into Botswana's tourism market is being used by pro-culling lobbyists to put pressure on the government to reduce its elephant population. The company is known to be concerned that Chobe has turned into a "single-species" reserve and that the elephants might eventually start dying of starvation.

"The apartheid-era conservation management style with the gun has a strong lobby in Botswana. But culling thousands of elephants in Chobe will be unbelievably devastating - it will turn the place into an elephant desert. We now know that elephants run like hell after there has been a cull," says a member of the anti-culling group. Like most of the people at the coalface of this acrimonious battle, he preferred not to be named.

Estimates of how many elephants there are in northern Botswana vary between 50 000 and 110 000. Scientists argue that, to keep the population stable, between 6 000 and 10 000 need to be removed each year.

"If the animal rightists win this fight, they will be responsible for the loss of incredible biodiversity in Chobe. There will inevitably be a die-off of elephants. Those who have been responsible for preventing or avoiding management will not hold themselves accountable for the consequences," says an ecologist.

One&Only has consulted conservation scientists about Chobe, but refuses to be drawn directly into the debate. "Whether the elephants live or die won't be a call we make," says the company's development director, Graeme Stephens.

The Kerzners have been dragged into the Chobe debacle because they are negotiating with Debswana, the diamond conglomerate jointly owned by De Beers and the Botswana government. Chobe is one of a "bouquet of sites" Debswana is contemplating developing in a quest to break into the ecotourism industry.

Stephens says One&Only is keen to partner Debswana's ecotourism ventures, though Chobe is not the plum the Kerzners were hoping for. "We are aiming at the six-star market, but Chobe does not offer that any more. Its trees have been destroyed and the tourist volumes are high."

Debswana is getting into tourism because it is the "next engine of growth after diamonds", says corporate communications manager Jacob Sesinyi. The company was given permission to develop a site in Chobe in May last year and is proud of the fact that it will only start designing and developing the site after a thorough environmental impact assessment has been completed.

The proposal to set up a development in the national park has been controversial from the start. Critics argue that the site was badly chosen, and that further development along the Chobe river front will exacerbate tourism pressures and disturb game the tourists come to see. The original idea was to set up a five-star 96-room hotel with tennis courts and a jetty.

The controversy over the development fuels the elephant-culling debate. Dereck Joubert, a renowned wildlife filmmaker who has been working in northern Botswana for 25 years, says one of the reasons elephants increasingly congregate along the Chobe river is because growing tourist numbers have disturbed their drinking patterns and they no longer feel secure about moving off inland to feed.

Joubert's latest documentary records how lions in the Chobe area are preying on elephants, including healthy adults. "Heavy-handed" management of wildlife such as culling could interfere with people being able to observe such surprising behaviour among animals, he says.

"Any decisions on whether to cull elephants or not need to be made on a conservation basis and should not be influenced by business decisions," he says.

Alternatives suggested by Joubert and like-minded conservationists include opening up transfrontier parks and encouraging the elephants to use migratory corridors, linking a growing number of game reserves being opened in neighbouring countries.

These options could take a while to realise, so physically translocating some elephants to other reserves might be an interim solution. In the meantime, flights to Maun, capital of northern Botswana, are filled daily with adversaries in the culling debate.

Bibliography

[1] R.F.W Barnes, G.C.Craig, H.T.Dublin, G.Overton, W. Simons and C.R Thouless (1998): *African Elephant Database 1998*: 139-143.
A copy of the total database can be downloaded in pdf format at the web address: www.iucn.org/themes/ssc/sgs/afesg/aed/pdfs/aed1998.pdf

[2] Christian Chad Cheney (1998): *A GIS based approach to determining the ecological carrying capacity and species mix of the extant large herbivores in the North Eastern Tuli Block, Botswana.*

[3] Timothy F. Corfield (1973): *Elephant mortality in Tsavo National Park, Kenya.* E. Afr. Wildl. J. Volume 11: 339-368.

[4] Kevin J. Duffy, Bruce R. Page, John H. Swart, Vladimir B. Bajić (1999): *Realistic parameter assessment for a well known elephant-tree ecosystem model reveals that limit cycles are unlikely.* Ecological modelling 121: 115-125.

[5] R. East (1984): *Rainfall, soil nutrient status and biomass of large african savanna mammals.* African journal of ecology 22: 245-270.

[6] John Merritt Emlen (1984): *POPULATION BIOLOGY: The Coevolution of Population Dynamics and Behavior.*

[7] Richard J. Gaylord, Kazume Nishidate (1996): *Modeling Nature. Cellular Automata Simulations with Mathematica*: 143-154.

[8] J. Hanks (1972): *Reproduction of elephant (Loxondonta africana) in the Luangwa Valley, Zambia.* Fert. 30: 13-26.

[9] J. Hanks and J.E.A McIntoch (1973): *Population dynamics of the African elephant.* Journal of zoology 169: 29-38.

[10] H. Jachmann (1980): *Population dynamics of the elephants in the Kasungu National Park, Malawi* Netherlands Journal of zoology: 622-634.

[11] J. Johnson O.W and I.O. Buss (1965): *Molariform teeth of male African elephant in relation to age, body dimensions and growth.* J. Mammal 46: 373-384.

[12] J.H. Kenneth, G.R. Ritchie (1953): *Gestation periods. A table and bibliography.*

[13] W.A. Krumrey and I.O. Buss (1968): *Age estimation, growth and relationships between body dimensions of the female african elephant.* J. Mammal 49: 22-31.

[14] E.M Lang (1967): *The birth of an african elephant (loxodonta africana) at Basle Zoo.* International Zoo Yearbook 7: 154-157.

[15] R.M. Laws (1966): *Age criteria for the african elephant.* E. Afr. Wildl. J. 5: 49-52.

[16] R.M. Laws (1969): *Aspect of reproduction in the african elephant, Loxodonta Africana.* J. Reprod. Fert, Suppl. 6: 193-217.

[17] R.M. Laws (1969b): *The Tsavo Research Project.* J. Reprod. Fert, Suppl. 6: 495-531.

[18] R.M. Laws, I.S.C. Parker and R.C.B. Johnstone (1970): *Elephants and habitats in north Bunyoro, Uganda.* E. Afr. Wildl. J. 8: 163-180.

[19] R.M. Laws, I.S.C. Parker and R.C.B. Johnstone (1975): *Elephants and their habitats. The ecology of elephants in north Bunyoro, Uganda.*

[20] Chi-Kwong Li, Hans Schneider (2001): *Applications of Perron-Frobenius Theory to population dynamics.*

[21] MATLAB help: (the complete documentation can be found at the web address: http://www.mathworks.com/support)

[22] Hugh Murrell (1994): *Planar Phase Plots and Bifurcation Animations.* The Mathematica journal, Vol4, issue3: 77-81.

[23] R.M. Nisbet, W.S.C. Gurney (1982): *Modelling fluctuating populations.*

[24] J.H. Pollard (1973): *Mathematical models for the growth of human populations.*

[25] Julien Clinton Sprott: *Predator-Prey Dynamics for rabbits, trees and romance.* (http://sprott.physics.wisc.edu/lectures/predprey)

[26] Eric Weisstein: *Eric Weisstein's world of mathematics.* (http://mathworld.wolfram.com/)

[27] Ian John Whyte (2001): *Conservation management of the Kruger National Park Elephant population.*

[28] Stephen Wolfram (2002): *A New Kind of Science.*

Index