

THE EFFECT OF PHENOTYPIC PLASTICITY ON POPULATION DYNAMICS

12th August 2005

Richèl Bilderbeek¹

University of Bristol
School of Biological Sciences
Woodland Road
Bristol, BS8 1UG

This work was submitted as a Master's research project for the Rijksuniversiteit Groningen, Holland

Abstract

We investigated whether phenotypic plasticity should be incorporated in population dynamics models to predict future trends. Using an individual-based model inspired by soil mites, we found that phenotypic plasticity can significantly change population mean and coefficient of variation (CV) in population size. Our results also suggest that under some circumstances there can be a complete dynamic feedback in which plasticity alters the population size and vice versa. The properties of this feedback loop deserve further investigation. We consider two possible routes to extinction: (a) via large oscillations (stochastic chance of extinction because of high CV in population size). (b) via a monotonic approach to extinction (intrinsicly doomed populations). We found that as conditions get harsher (as mimicked by increasing mortality or higher metabolic costs) both populations are less likely to reach extinction via (a). Instead they seem to pass a bifurcation point that led to extinction due to (b). If in a gradually harsh environment, the non-plastic populations are predicted to go extinct earlier than the plastic populations via (b). Before passing the bifurcation point to go extinct via (b), both populations decrease in CV. The probability of extinction via (a) in plastic populations could be both greater and less than that of non-plastic populations, depending on the circumstances. We argue that for species when sufficient data is available concerning plastic responses, predictions from population dynamics models could be substantially improved by incorporating plasticity into the model.

1 Introduction

Phenotypic plasticity is the ability of an organism to react to an environmental input with a change in form, state, movement, or rate of activity[8]. Examples of plastic life-history parameters can be e.g. the partitioning

¹Enquiries to Dr Robert J.H. Payne, School of Biological Sciences, Bristol University, Woodland Road, Bristol.
email:robert.payne@bristol.ac.uk

of resources over growth and build-up of reserves: the individual might invest more resources in reserves when conditions are poor. Another example is the trade-off between number of offspring and quality of offspring.

An important state transition in an organism's life history is when it can start reproducing. This has already been noted in e.g. [2], describing the optimization for an age-structured life-history model. This has the underlying assumption that at the same age, all individuals are in the same physiological state[6]. In nature, this is unlikely. For example, it has been found in soil mites that maternal effects have effects on population dynamics, that can be detected even after three generations [1]. Some modelling on the evolution of phenotypic plasticity has been done [3], but to the best of our knowledge, phenotypic plasticity has not been included in population dynamics models.

The goal of this research is to investigate whether phenotypic plasticity should be incorporated in population dynamics models to predict future trends. It uses a system inspired on soil mites. Soil mites have many traits making them suitable as a study organism: the species has suitable size and generation time for experiments at the population level. Except for reproduction, juveniles are ecologically similar to adults. Some evidence for phenotypic plasticity has been found in the species [7, 1]. The plastic trait examined was the partitioning of resources to either growth or increase in reserves. Plasticity in this trait enables the organism to invest more resources into reserves when conditions are harsh at the cost of growth, thus delaying reproduction.

The main research questions asked are:

1. Can mean population size be changed due to phenotypic plasticity?
2. Can oscillations in populations size be damped, amplified or induced due to phenotypic plasticity?
3. Can the parameter space in which a population survives be increased or decreased due to phenotypic plasticity?
4. Is there a two-way interaction between plasticity and population size in individuals?

2 In silico model system

An individual-based model inspired on the soil mite system has programmed. The structure of the computer simulation is described in appendix B. The full mathematical system is described in detail in appendix A. We set up a mathematical system in such a way it can quantitatively mimick the biology of the organism. Many constants have been chosen so, that a simpler system is obtained. This simplified system can be found in the box 'Simplified mathematical system used'.

In every experiment, simulations were set to a standard set of parameters. Comparisons were done from simulations differing only one parameter. These parameters can be found in the box 'Simplified mathematical system used'. A typical simulation lasted 500 timesteps. To avoid transient effects, only the last 300 were used in measurements. In this research the coefficient of variation (from now on: CV) was used as an approximate estimator of extinction risk. We consider two possible routes to extinction: (a) via large oscillations (stochastic chance of extinction because of high CV in population size). (b) via a monotonic approach to extinction (intrinsicly doomed populations). The higher the CV, the more likely a population gets extinct due to the route of extinction of type (a). To investigate whether there is a two-way interaction between plasticity and population dynamics, a phase plane is made of the plastic trait and the number of individuals in the population.

3 Results

Population size are found to always oscillate in time or exponentially decrease to extinction. Finding oscillations in age-structured species is very common [5]. A typical timeplot is figure 1. It shows that population size, resources, body size, reserves and resource partition are all oscillating. The effect of stochasticity in foraging success, c_f , has been examined, but was found to have no significant qualitative effect.

3.1 Effect of phenotypic plasticity on population size

Always, the plastic population has a higher population size than the non plastic. In figure 2 we investigated the effect of metabolic rate. The higher this value, the higher the metabolic rate. For example, this value can increase when foraging costs are high or at lower temperatures. The non-plastic population went extinct sooner via route (b) when increasing either the death rate, d_i , or the metabolic rate, m_i . There is a region in parameter space in which the non-plastic population inevitably goes extinct (via (b)), where the plastic survives. This has been shown in three timeplots, figures 3, 4 and 5. Note that in 4 intermittency in population size can be seen. Intermittency has been observed frequently in other timeplots. When we analyse the same simulation as figure 2, non-plastic populations have a lower CV (figure 6). This is a counter-intuitive results: although the non-plastic population has a lower CV, it is the first to go extinct via (b). The plastic populations have a higher probability to go extinct due to (a). For the plastic populations, the CV is monotonically decreasing for increasing values of m_i . But, at a certain value of m_i , the population will also go extinct via (b).

Another parameter examined is the death rate, d_i . Higher values mean that an individual will have a higher probability to die, due to lack of reserves. Here the mean population size of the plastic population is above the non-plastic values as well (fig. 7).

When analysing the simulation of figure 7 for CV, non-plastic populations do not have always have a lower CV anymore (see figure 8). For low values of d_i , the plastic population has a lower CV instead. When increasing d_i the non-plastic populations will pass these and have a lower CV. Increasing d_i causes the non-plastic populations first to die due to (b).

Plasticity in partitioning resources has found to always result in higher population size means. Also, a plastic population is found to survive under a broader parameter range. When the experimental setup gets more difficult for the individual to survive, the non plastic populations get extinct earlier.

For the CV's over a range of metabolic rate, m_i , a counter-intuitive result has been found: although the non-plastic simulation has a lower CV (which is assumed to be related to extinction risk) it is the first to go extinct when increasing m_i . In the parameter range of m_i in which only the plastic populations survive, the related counter-intuitive effect is found: when increasing m_i even more, the plastic population decreases in CV. A further increases leads to unviable populations. A similar pattern in CV is observed when varying death rate d_i . Interestingly, for a low death rate, the plastic population has a lower CV compared with the non-plastic population.

3.2 Two-way interaction between phenotypic plasticity and population dynamics

The interaction between phenotypic plasticity and population dynamics was investigated by making a phase plane between population size and the plastic trait. In the non-plastic population, p_i was fixed, resulting in the phase plane of figure 9. When the trait was plastic and metabolic rate, m_i , was varied the following interactions were seen:

- for a low value of m_i , there appears to be a linear interaction, as in figure 10.
- Increasing m_i results in a pie-shaped trajectory, as in figure 11.
- Increasing m_i even more, this trajectory is increasingly losing its shape, as in figure 12.

This sequence is found in a simulation without stochasticity in the resources. Adding this stochasticity makes, as expected, the trajectories less sharp. When increasing the death rate, d_i , a similar sequence was found. The interaction between population dynamics and plasticity is hinted to be a two-way interaction (fig. 11). Under different parameters, this interaction is either linear or a less sharp version of the two-way interaction. Sometimes the trajectory was a figure of eight (not shown), which implies there are at least three dimensions causing the effect, i.e. there are at least three parameters responsible for this trajectory.

4 Discussion

We have identified several ways in which phenotypic plasticity can significantly affect population dynamics, to the extent that predictions of future population dynamics trends would in some cases be seriously misleading if the phenotypic plasticity would not be taken into account.

The pattern found on coefficient of variation (CV) when increasing the death rate or metabolic rate is interesting. For low death or metabolic rates, it can not be predicted whether the plastic or non-plastic population will have a lower CV. A lower CV means a higher likelihood that a population goes extinct due to large oscillations combined with stochastic chance (route (a)). When increasing the death or metabolic rate, both non- and plastic simulations have lowest CV before going extinct via a monotonic approach to extinction (route(b)). If this will be a general pattern in theoretical and experimental work on phenotypic plasticity, this gives conservation biologists an important biological clue: when an oscillating population decreases its relative oscillations in population size, it might be on the verge of extinction via (b). Also, it raises the question of whether extinction risk might be correlated to the level of phenotypic plasticity in a species. Perhaps when investigating conservational biology data this correlation can be found.

The linear relation between developmental plasticity and population dynamics might contain important clues: it occurs under environmentally mild conditions. This might suggest that then there is either no individual variation or the feedback between population size and the plastic trait has a short lag time.

That these phenomena are found might rely heavily on one single modelling implementation: the non-plastic individuals always have a partition of 0.5. This might limit the generality of our results. It has been observed that the average partition of the plastic individuals is mostly in the range 0.4–0.6. The extent to which this deviation from 0.5 makes the comparisons invalid is unknown.

In this research, the plastic trait investigated was only environmental dependent. Another possible plastic trait was egg quality of offspring. These maternal egg-size decisions are found *in vivo* to influence population dynamics even after three generations [1]. It would be interesting if this simulational setup would also yield similar observations.

This research suggests that phenotypic plasticity has an impact on population dynamics. Yet, it is not taken into account in current theoretical models. Some of these assume that a genotype produces a single phenotype and investigate the change in allele frequencies in populations, instead of investigating change of phenotype frequencies. My hypothesis would be that when modelling the soil mite system using this approach, the results could be identical from the ones produced in this research. The only thing I expect to be different is the time scale, as evolution acts slower than phenotypic plasticity, due to an assumed low mutation rate. When indeed a phenotypic plasticity setup appears to be the same as a genetic setup, the

simulation can be used to find out under which circumstances a distinction is most likely to occur. This enables to determine whether a change in phenotypes in a population is due to phenotypic plasticity or genetics. When this experimental setup is found in silico, the in vivo soil mite system could be used to test this. It will be important for future research to disentangle the mechanisms of genotypic and plastic change in populations.

Although phenotypic plasticity is ubiquitous in nature, it is neglected in most theoretical models. This research added this piece of realism into the model. The effects found might give conservation biologists an important indicator of extinction risk: when an inherently oscillating population is decreasing in CV might hint that an environmental parameter (e.g. death rate) has increased to a critical level. Knowing that when this parameter is increased even more, the population will go extinct, priorities in maintaining biodiversity can be shifted in conservation programs.

Incorporating knowledge of plasticity into population dynamics models will be a challenge, but our results suggest that to do so might substantially help in making better predictions of future population trends in at-risk species.

5 Acknowledgements

This Master's student project has been supervised by Dr Robert JH Payne from the School of Biological Sciences, Bristol. The project was done in collaboration with Professor Tim G Benton from the University of Leeds.

References

- [1] Benton TG, personal communication
- [2] Bulmer M. 1994. Theoretical Evolutionary Ecology. Sinauer Asociates, Inc. ISBN: 0878930787.
- [3] Ernande B, Dieckmann U. 2004. The evolution of phenotypic plasticity in spatially structured environments: implications of intraspecific competition, plasticity costs and environmental characteristics. *J. Evol. Biol.* 17:613-628.
- [4] Gamma E, Helm R, Johnson R, Vlissides J. 1995. Design patters. Addison-Wesley Professional. ISBN: 0201633612.
- [5] Greenman JV, Benton TG, Boots M, White AR. 2005. The evolution of oscillatory behaviour in age-structured species. *Am. Nat.* 166:68-78.

- [6] Houston AI, McNamara JM. 1999. Models of adaptive behaviour: An approach based on state. Cambridge University Press. ISBN: 0521655390.
- [7] Plaistow SJ, Lapsley CT, Beckerman AP, Benton TG. 2004. Age and size at maturity: sex, environmental variability and developmental thresholds. Proc. R. Soc. Lond. B 271: 919-924.
- [8] West-Eberhard, MJ. 2003. phenotypic plasticity and evolution. Oxford university press. ISBN: 0195122556 (pbk.).

A Formula overview

Parameter	Meaning
A, B, C, D, E	Constants that differ between every formula
a	Conversion efficiency from resources to body mass
b	Conversion efficiency from resources to reserves
c	Conversion efficiency from reserves to egg biomass
d_i	Death rate of i
f_i	Absolute resource intake of i . Feeding rate of i
g_i	Fraction of egg biomass invested per offspring
i	Index of an individual, $i = 0 \dots I_{tot} - 1$
m_i	Metabolism
n	Total number of individuals in population
n_i	Number of offspring produced by adult
p_i	Fraction of resources allocated to body size
q_i	Resources invested in egg biomass
r_i	Resources available to individual. Local resources.
s_c	Body size to reach adulthood
s_i	Body size
v_i	Reserves
x	Resources
w_i	Foraging success

Table 1: Overview of the parameter names

Population dynamics

The number of individuals in a population is described with the general equation 13.

$$\text{population size}(t+1) = \text{population size}(t) - \text{deaths} + \text{births} \quad (13)$$

There are two modes of modelling births. The first mode uses equation 14, which describes how many offspring are produced. From the number of offspring the amount of reserves from the egg biomass is calculated. The other mode uses equation 15, which first calculates the fraction of egg biomass per individual. Death is modeled the same in all simulation. Due to the two different reproduction modes, there are two formula's describing population size in time. The population size when the program is in the reproduction

mode using N can be described as follows:

$$n(t+1) = n(t) - \frac{1}{n(t)} \sum_{i=0}^{i=n-1} d_i + \sum_{i=0}^{i=n-1} n_i \mathcal{H}(s_i - s_c) \quad (14)$$

in which n denotes population size in number of individuals, d_i the chance individual i dies and n_i the number of offspring individual i creates. \mathcal{H} denotes a Heaviside function, which is one when its argument is bigger or equals zero, otherwise it is zero. In this case, it means that when an individual's size s_i is below the threshold size to become an adult, s_c , it will not reproduce. Note that 14 is only an approximation. This due to chance effects in the death chance d_i and the rounding-off of n_i to whole numbers. The population dynamics when the program is in the reproduction mode using G is very similar.

$$n(t+1) = n(t) - \frac{1}{n(t)} \sum_{i=0}^{i=n-1} d_i + \sum_{i=0}^{i=n-1} \frac{1}{g_i} \mathcal{H}(s_i - s_c) \quad (15)$$

It is similar to 14, except for that the number of offspring individual i produces is $\frac{1}{g_i}$. g_i being the fraction of egg biomass it invests per offspring. In this research, s_c is set to 10.

Resource dynamics

The resources are described with the general formula 16:

$$\text{resources(time + 1)} = \text{resources(time)} + \text{inflow} - \text{consumption} \quad (16)$$

This is equivalent to the formal formula 17:

$$x(t+1) = x(t) + R(x, t) - \sum_{i=0}^{i=n-1} f_i \quad (17)$$

Resources x are changed by resource inflow $R(x, t)$, which is dependent on the resources x themselves and time t . Resource inflow can be dependent on resources when e.g. the resources reproduce (yielding predator-prey dynamics) or resources are density dependent. Having the resources being dependent on time t enables the inflow to be seasonal. The consumption depends on the amount of individuals, n , and how much they consume f_i .

Individual's dynamics

An individual has a body size and a certain amount of reserves. It can invest absorbed resources in growth in body size and increase in reserves. Also, an individual can invest its reserves in egg production, but only when it is an adult. Generally, this modeled as:

$$\begin{aligned} \text{size(time+1)} &= \text{size(time)} + \text{resources invested in growth} \\ \text{reserves(time + 1)} &= \text{reserves(time)} + \text{resources invested in reserves-reserves invested in egg product} \end{aligned} \quad (18)$$

To describe this more formal:

$$s_i(t+1) = s_i + a.p_i.f_i \quad (20)$$

$$v_i(t+1) = v_i + b.(1-p_i).f_i - m_i - q_i \quad (21)$$

Resource intake, r_i , is invested in body size, s_i , and reserves, v_i . The increase in body size depends on the fraction of resources partitioned to body size increase, p_i , the resource intake f_i and a conversion factor a which determines the increase in body mass per unit resource. The change in reserves depends on resources, metabolism and the production of eggs. The reserve increase depends on resource intake f_i , the fraction of resources allocated to reserves, $1-p_i$, and a conversion factor b which determines the increase in reserves per unit resource. The decrease in reserves due to metabolic processes is called m_i . Reserves can also decrease by investment in eggs, q_i . In the computer program, a and b in 20 and 21 are called ‘alpha’ and ‘beta’ respectively.

Death rate per individual

The chance an individual dies, d_i , depends on body size s_i and reserves v_i .

$$\text{default} : D(s_i, v_i) = d_i = A + \frac{(1-A)D\left(\frac{s_i}{v_i+B}\right)^C}{1 + \left(\frac{s_i}{v_i+B}\right)^C} \quad (22)$$

The death rate has two parts: a constant death rate and a death rate dependent on the ratio between body size s_i and reserves v_i . The constant or background death rate is the first term, A . The second term goes to its maximum quickly when an individual has a high body size, s_i , but low reserves, v_i . This means that after an individual has produced offspring, its size will be big and its reserves depleted, making the probability it dies approximately 100%. In the box ‘Simplified mathematical system used’, c_d is equivalent to A in this equation. In this research, values for B , C and D were 1, 10 and 0.15 respectively.

Feeding rate per individual

When an individual has the r_i resources available, it also depend on its size s_i how much it actually consumes. This consumption is f_i . At this point of the research, it wasn’t necessary to implement such a size dependency.

$$\text{default} : F(r_i, s_i) = f_i = A.r_i \quad (23)$$

In this research A was set to one.

Fraction egg biomass per offspring for reproducing adult

The egg size produced by an individual g_i depends on the resource intake f_i , the amount of reserves v_i and the amount of biomass put into the eggs q_i . The fraction of this biomass invested per offspring only depends only on the first two. This equation is only used in the simulation's reproduction mode 'Fraction'. The amount of eggs an individual produces is $\frac{1}{g_i}$ rounded off.

$$\text{constant} : G(f_i, v_i) = g_i = A \quad (24)$$

$$\text{plastic} : G(f_i, v_i) = g_i = 1 - e^{-A(f_i - v_i)^2} \quad (25)$$

Where if $G_{\text{plastic}} < 0.001$, then $G_{\text{plastic}} = 0.001$

Metabolism per individual per time unit

The amount of reserves an individual has to use for metabolism m_i depends on its size s_i and its reserves v_i .

$$\text{constant} : M(s_i, v_i) = A \quad (26)$$

$$\text{default} : M(s_i, v_i) = A \cdot s_i \quad (27)$$

In the box 'Simplified mathematical system used', A in eq. 27 is equivalent to c_m .

Number of offspring produced for reproducing adult

The number of offspring produced, n_i , increases for higher consumption, f_i , and higher amount of reserves, v_i . This equation is only used in the simulation's reproduction mode 'Amount'. The amount of reserves are assigned to each offspring is the egg biomass investment of the parent, q_i , divided by the amount of offspring produces, n_i .

$$\text{constant} : N(f_i, v_i) = n_i = A \quad (28)$$

$$\text{plastic} : N(f_i, v_i) = n_i = A(D \cdot f_i + E \cdot v_i) + (1 - A)(BD + CE) \quad (29)$$

In 29, the constant A denotes the amount of plasticity: when A is zero, the individual behaves non-plastic. In this research, equation 31 with A always set to zero. The temp $(BD + CE)$ was set to 3.

Partition

The fraction an individual invests in body size, p_i is dependent on the resource intake f_i and the amount of reserves it has v_i .

$$\text{constant} : P(f_i, v_i) = A \quad (30)$$

$$\text{plastic} : P(f_i, v_i) = A \cdot \sqrt{\frac{\left(\frac{f_i}{C}\right)^B \cdot \left(\frac{v_i}{E}\right)^D}{1 + \left(\frac{f_i}{C}\right)^B \cdot \left(\frac{v_i}{E}\right)^D} + 0.5(1 - A)} \quad (31)$$

In equation 31, the constant A denotes the level of plasticity and is the same as c_p in the box 'Simplified mathematical system used'. Values used for B , C and D were 1, 0.1 and 2 respectively.

Net biomass investment in eggs

The investment of reserves into egg biomass q_i is dependent on its size s_i and its reserves v_i . If an individual is not an adult, i.e. its size s_i is smaller than the size an adult has s_c , then it produces no eggs, else it invests the amount q_i of its reserves into eggs. Either the investment in eggs biomass is constant or monotonically increases when the parent has a higher amount of reserves.

$$\text{constant} : Q(v_i) = q_i = \mathcal{H}(s_i - s_c)A \cdot v_i \quad (32)$$

$$\text{default} : Q(v_i) = q_i = \mathcal{H}(s_i - s_c) \cdot \frac{A \cdot v_i}{1 + v_i^B \left(\frac{C}{A}\right)} \quad (33)$$

In this research, equation 32 is used, with A set to one.

Resource inflow per time unit

Resources can either flow with a constant rate, the resources can be a reproducing prey or the resources can be seasonal.

$$\text{constant} : R(x, t) = A - B \cdot x \quad (34)$$

$$\text{logistic} : R(x, t) = Ax \left(1 - \frac{x}{B}\right) \quad (35)$$

$$\text{cosine} : R(x, t) = A + B \cdot \left(\cos\left(\frac{2\pi}{C} \cdot x + D\right)\right) \quad (36)$$

In this research, equation 34 was used, with A and B set to 100 and 0.001 respectively.

Foraging success

Resources can be distributed equally or unequally. The amount of resources an individual can maximally consume, r_i , depends on its foraging success, w_i , and the amount of resources each individual would get when having equal hunting success, $\frac{x}{n}$, in which x is the total amount of resources and n the population.

$$r_i = w_i \cdot \frac{x}{n}$$

Of course, there can be not more resources, x , extracted from the system than the sum of all resources possible for consumption r_i .

$$x \geq \sum_{i=0}^{i=n-1} r_i \quad (37)$$

There are two possible models for hunting success, w_i , in which 38 denotes equal hunting success, whereas 39 gives a linear deviation from equal hunting success.

$$\text{zero} : W(s_i, n) = w_i = \frac{1}{n} \quad (38)$$

$$\text{uniform} : W(s_i, n) = w_i = \left(1 + A(s_i - B)^2\right) C \left(1 - D + \frac{i}{n} \cdot 2.D\right) \quad (39)$$

$$\text{default} : W(s_i, n) = w_i = \left(1 + A(s_i - B)^2\right) C (1 + 2.D(\sigma - 0.5)) \quad (40)$$

in which σ denotes a random number $\in [0,1]$. The values of 39 are stored and shuffled afterwards. Otherwise, individual on index 0 would always receive the least resources. It is quicker than 40, as the latter performs one time-expensive random number operation per individual.

B Program overview

The program consists of 7 main classes, whose interactions are displayed in figure 13.

These classes are summarised here, after which a more details follow per class.

1. Sim4Interpreter: interprets the command-line input given by the user. If the user changes options or simulation parameters, it modifies OptionsFile to store their new value. If the user gave no argument, Experiment is executed.
2. OptionsFile: manages file I/O for the options file. Its values can be changed by Sim4Interpreter only. If Sim4Interpreter changes simulation parameters, it modifies SimulationFile. When an Experiment is executed, it is read by Experiment and Simulation.
3. SimulationFile: manages file I/O for the simulation parameters file. Its values can be changed by OptionsFile only. When an Experiment is executed, is is read by Simulation and FunctionsSingleton.
4. Experiment: manages the execution of one or more Simulations using the options specified by the user. It obtains these options by reading from OptionsFile.
5. Simulation: performs an individual-based simulation, generating file output. It first sets up the mathematical system in FunctionsSingleton by reading from SimulationFile. After this, a virtual population of SoilMites are let to interact with each other. Many parameters of the populations (and individuals) are measured and given to Experiment.

6. SoilMite: a virtual soil mite. It has the traits 'body mass' and 'reserves' (and some more for measurements). It reads FunctionsSingleton for the mathematical formula's modelling its behaviour.
7. FunctionsSingleton: embodies the mathematical system described in 'Formula Overview'. When a Simulation starts, it is set by Simulation. Then both Simulation and SoilMite read from it.

Sim4Interpreter

Interprets the command-line input given by the user. It only exists in the main function. If the user changes options or simulation parameters, it modifies OptionsFile to store their new value. If the user gave no argument, Experiment is executed. Figure 14 depicts the role of Sim4Interpreter in the main program flow.

When the user gave arguments, Sim4Interpreter interprets these. When the user entered something valid, there are three possibilities:

1. The user changed the name of the OptionFile he uses. If this file exists, the new filename is written to the initialization file (Sim4.ini).
2. The user changed an option. OptionsFile, already having read its own file, changes this value and stores it to disk.
3. The user changed a Simulation parameter. Then this is delegated to OptionsFile. OptionsFile, already having read its own file, knows which Simulation Files the user uses. OptionsFile then changes the relevant (or all) SimulationFile(s). Like OptionsFile, when a value is changed in a SimulationFile, it immediately stores this to disk.

OptionsFile

The class OptionsFile manages file I/O for the options file. Its values can be changed by Sim4Interpreter only. If Sim4Interpreter changes simulation parameters, it modifies SimulationFile. When an Experiment is executed, it is read by Experiment and Simulation.

Figure 15 shows the flowchart of reading and writing to OptionsFile. Whenever OptionsFile is created, it immediately reads its values from file. If any of its values are changed, it immediately saves these changes to disk. When the user wants to change a parameter in one or all SimulationFile(s), Sim4Interpreter delegates this task to OptionsFile. OptionsFile then changes the SimulationFile(s) the user is using. As OptionsFile, when SimulationFile is changed, it immediately saves this to disk.

SimulationFile

SimulationFile manages file I/O for the simulation parameters file. Its values can be changed by OptionsFile only. When an Experiment is executed, it is read by Simulation and FunctionsSingleton. Whenever SimulationFile is created, it immediately reads its values from file. If any of its values are changed, it immediately saves these changes to disk (figure 18).

Experiment

This class manages the execution of one or more Simulations using the options specified by the user. It obtains these options by reading from OptionsFile.

Experiment only has one OptionsFile and access to all SimulationFiles, where Simulation has one OptionsFile and one SimulationFile. This enables the user to run multiple Simulations with the same options (figure 17).

Simulation

Simulation performs an individual-based simulation, generating file output. It first sets up the mathematical system in FunctionsSingleton by reading from SimulationFile. After this, a virtual population of SoilMites are let to interact with each other. Many parameters of the populations (and individuals) are measured and given back to Experiment. Experiment will then plot and/or save these to disk (figure 18).

SoilMite

a virtual soil mite. It has the traits 'body mass' and 'reserves' (and some more for measurements). It reads FunctionsSingleton for the mathematical formula's modelling its behaviour.

But, there are two different behaviours of reproduction. One uses the formula G to calculate the fraction egg biomass per reproducing adult, after which it produces $\frac{1}{G}$ offspring. The other uses N to directly calculate the number of offspring to be produced directly. Instead of choosing the right reproduction mode every reproduction, this is done using polymorphism (figure 19).

FunctionsSingleton

embodies the mathematical system described in 'Formula Overview'. When a Simulation starts, it is set by Simulation. Then both Simulation and SoilMite read from it.

The class its Singleton pattern makes sure that there is only one instance of this class [4]. As the mathematical system needs to be flexible, every formula is embodied in a Strategy [4]. For example, the death rate d_i is embodied in StrategyDbase. If the death rate needs to be constant, the derived class StrategyDconstant is instantiated. The Strategy pattern prevents using state-dependent if-statements, by using polymorphism to define which formula needs to be evaluated (figure 20).

Special functions

Reproduction

There are two different SoilMites. The first, SoilMiteFraction using G to calculate the fraction of the invested egg biomass, from this calculating the number of offspring by rounding off $\frac{1}{G}$. SoilMiteAmount uses N to directly calculate the amount of offspring.

Algorithm 1 Reproduction of SoilMiteFraction

```
mFractionBiomassPerOffspringInitial = mpFunctions->G(mFeedRate, mReserves);  
nOffspringInitial = 1.0/mFractionBiomassPerOffspringInitial;  
nOffspring = round(nOffspringInitial);
```

Algorithm 2 Reproduction of SoilMiteAmount

```
nOffspringInitial = mpFunctions->N(mFeedRate, mReserves);  
nOffspring = round(nOffspringInitial);
```

After calculating the number of offspring, their share of egg biomass is calculated after which they are created.

Distribution of resources

The distribution of resources is performed by the class StrategyWbase, in the function distributeResources. This function takes a arguments the total amount of resources and a vector to be filled with the same size as population number. The derived class StrategyWzero fills this vector with an equal amount of resources.

StrategyWuniform is more complicated (algorithm 3) . StrategyWdefault fills the vector with w_i values in which every value is independently calculated. As $\sum w_i \leq x$, this was checked and when false, all values of w_i were scaled down equally.

Algorithm 3 Distribution of resources, the function distributeResources in FunctionsSingleton

```
if (popSize==1) { vectorResources[0]=totalResources; return; }

const double resourcesEqual = totalResources / popSize;
const double step = (2.0*S.A)/(popSize-1);
double fraction = 1.0-S.A;
for (unsigned int index=0; index<popSize; index=index+1)
{
    vectorResources[index] = resourcesEqual*fraction;
    fraction=fraction+step;
}
random_shuffle(vectorResources.begin(), vectorResources.end());
```

The higher-level distribution and intake is put into Simulation as follows:

Algorithm 4 Distribution of resources as performed by Simulation. Note the last line, that performs the measurement of the amount of resources.

```
void Simulation::changeEnvironment(time)
{
    mResources=mResources+mpFunctions->R(mResources,time);
    mVectorResources.resize(popSize);
    mpFunctions.stochasticiseResources(mResources,mVectorResources);
}

void Simulation::intake(table)
{
    for (i=0; i<popSize; i=i+1) mPop[i]->intake(mVectorResources[i]);
    foodIntake = getSumFeedRate();
    mResources=mResources->foodIntake;
    table.mResources.push_back(mResources); //Measurement
}
```

C Figures

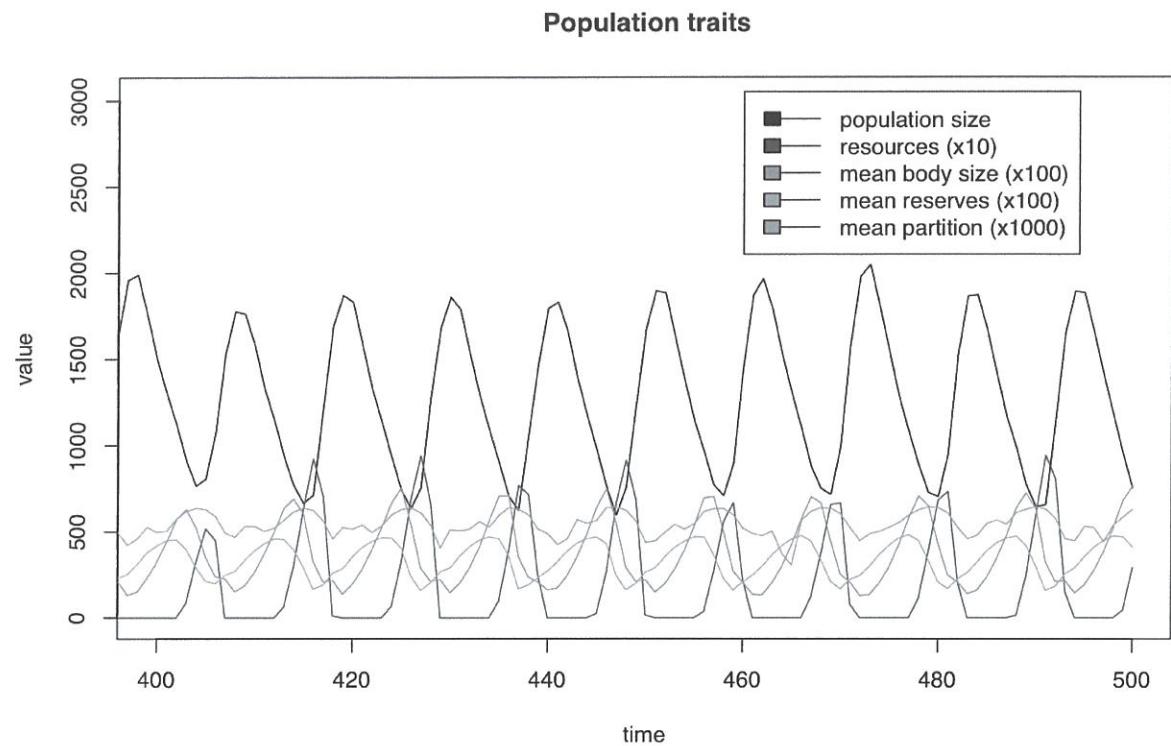


Figure 1: Population traits in time. $c_f = 0$, $c_d = 0.15$, $c_m = 0.0$, $c_p = 1$, $s_c = 10.0$

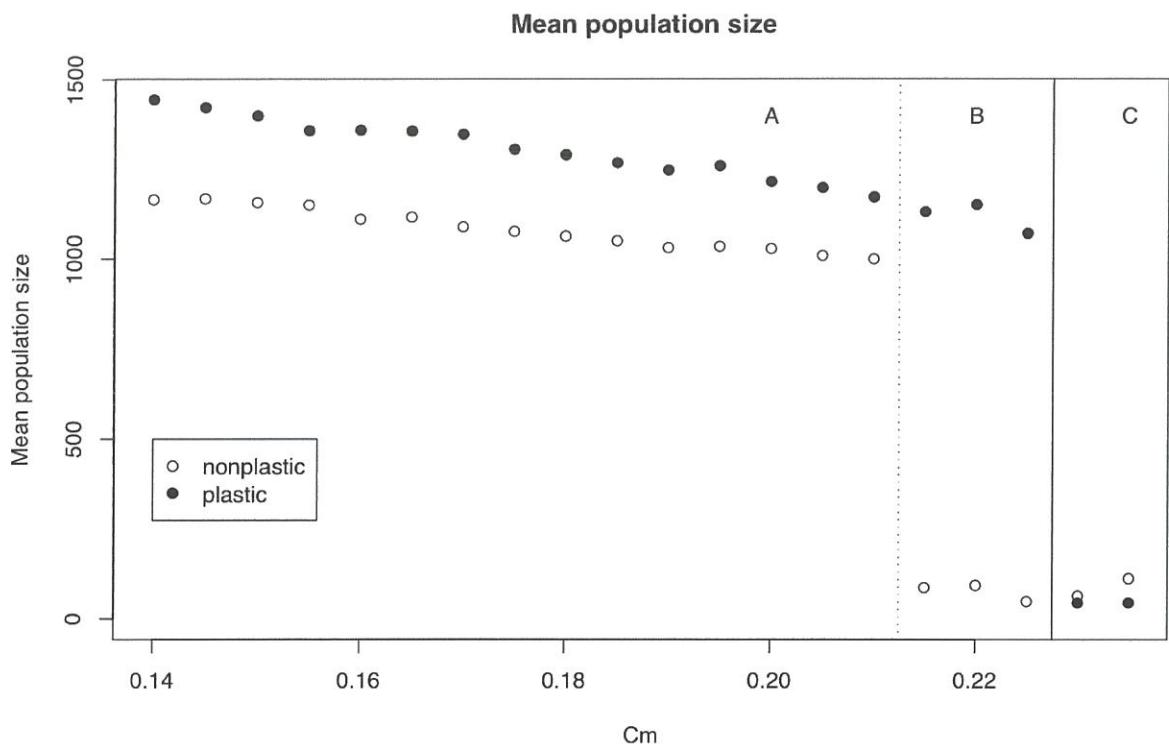


Figure 2: The effect of phenotypic plasticity on population extinction for different values of c_m . Right of the dotted and solid line non plastic and plastic populations go extinct respectively, after which the measured values lose relevance. $c_f = 0$, $c_d = 0.15$, $s_c = 10.0$.

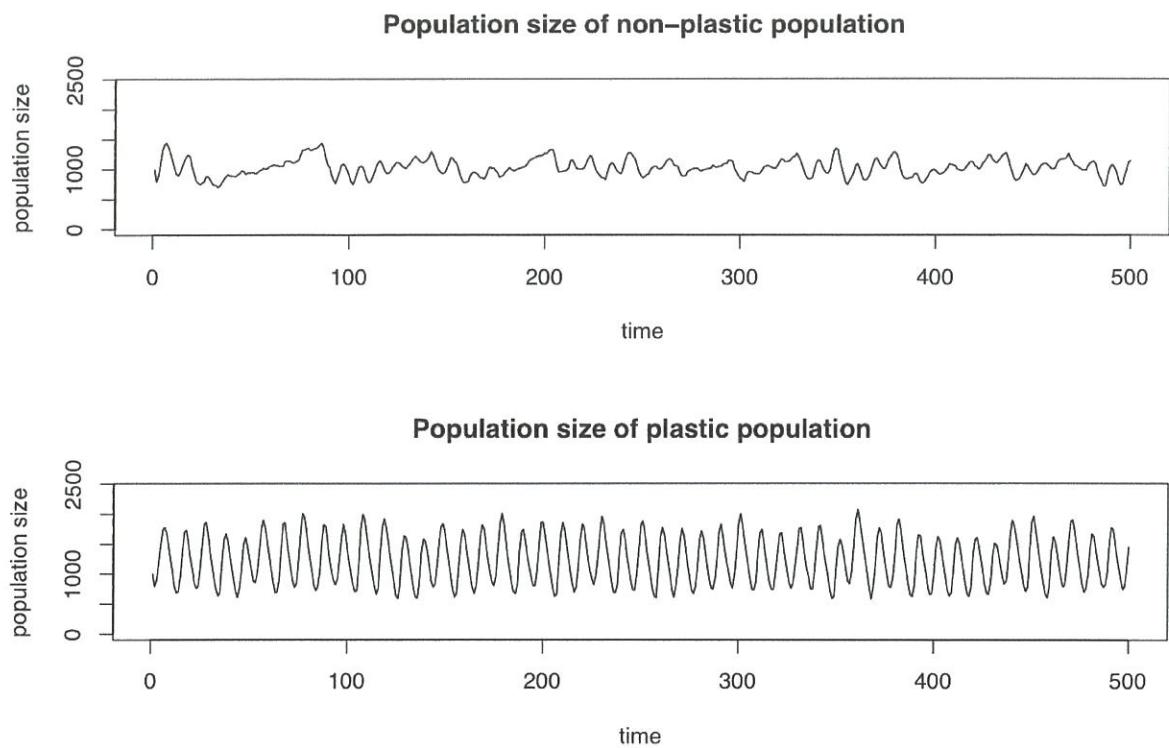


Figure 3: Timeplot for the point A indicated at figure 2. $c_f = 0$, $c_d = 0.15$, $c_m = 0.2$, $s_c = 10.0$.

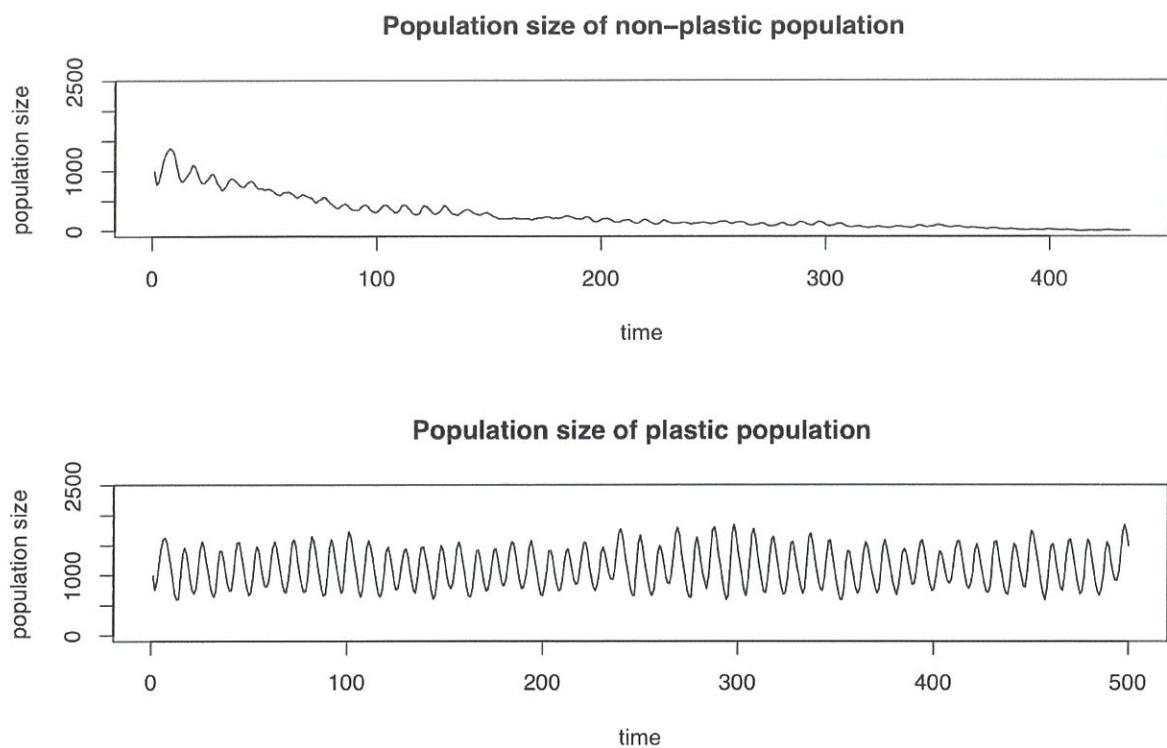
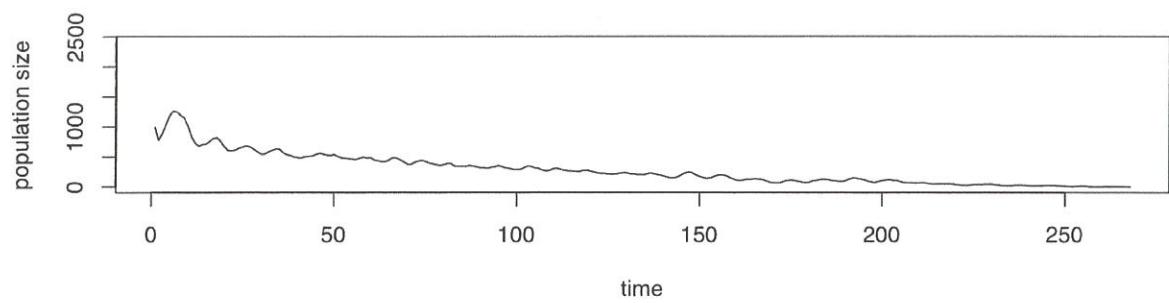


Figure 4: Timeplot for the point B indicated at figure 2. $c_f = 0$, $c_d = 0.15$, $c_m = 0.22$, $s_c = 10.0$.

Population size of non-plastic population



Population size of plastic population

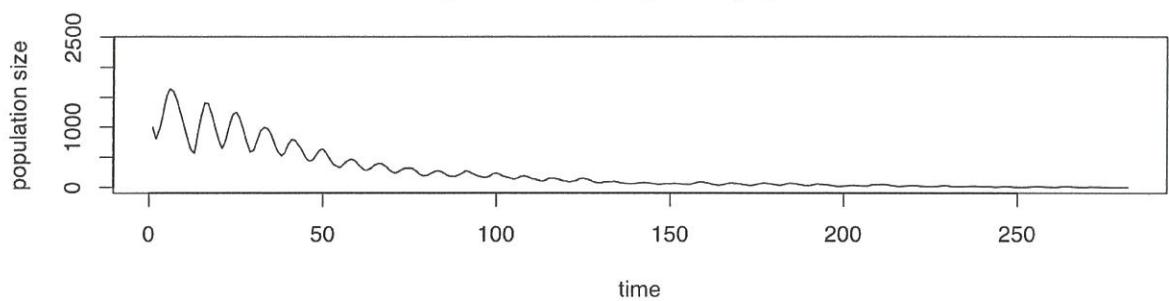


Figure 5: Timeplot for the point C indicated at figure 2. $c_f = 0$, $c_d = 0.15$, $c_m = 0.235$, $s_c = 10.0$.

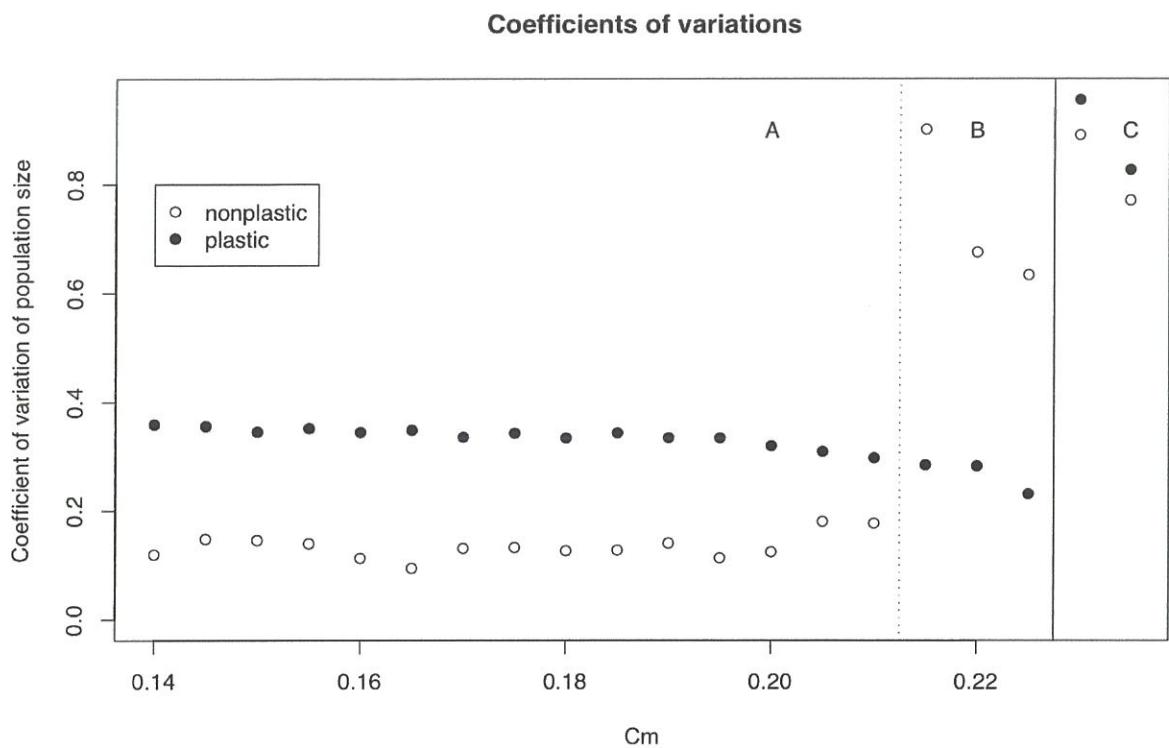


Figure 6: CV's for different values of c_m . Right of the dotted and solid line non plastic and plastic populations go extinct respectively, after which the measured values lose relevance. $c_f = 0$, $c_d = 0.15$, $s_c = 10.0$.

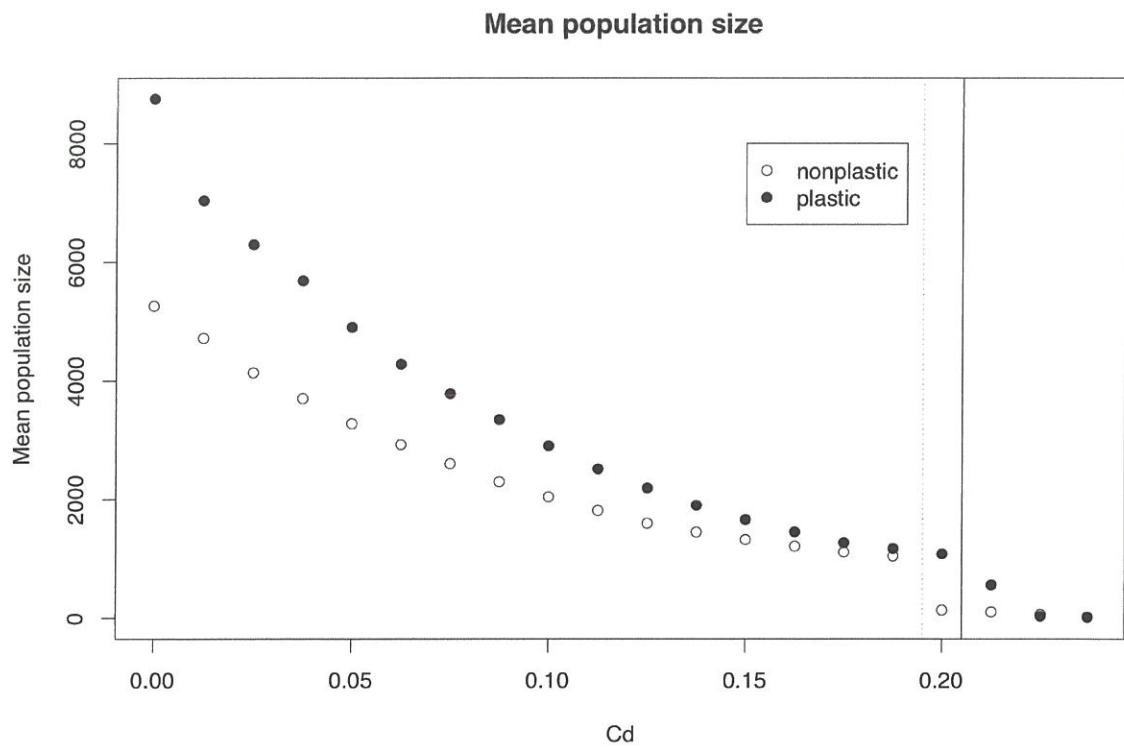


Figure 7: Mean population size for different values of c_d . Right of the dotted and solid line non plastic and plastic populations go extinct respectively, after which the measured values lose relevance. $c_f = 0.5$, $c_m = 0.1$, $s_c = 10.0$.

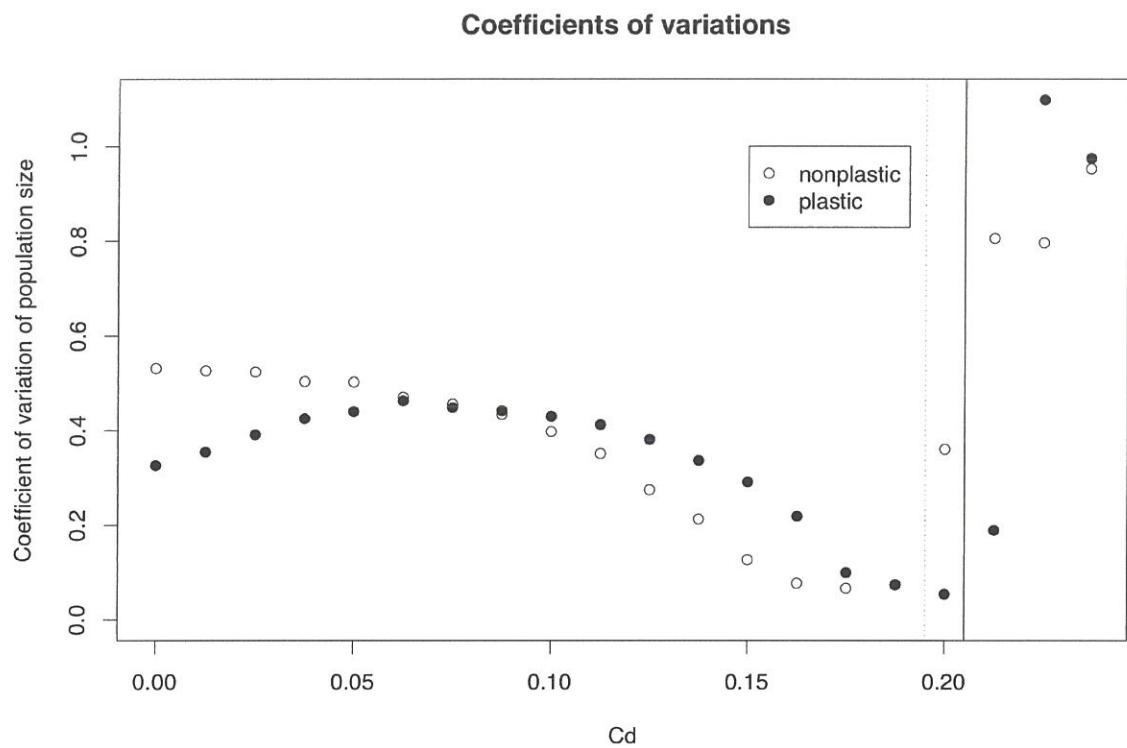
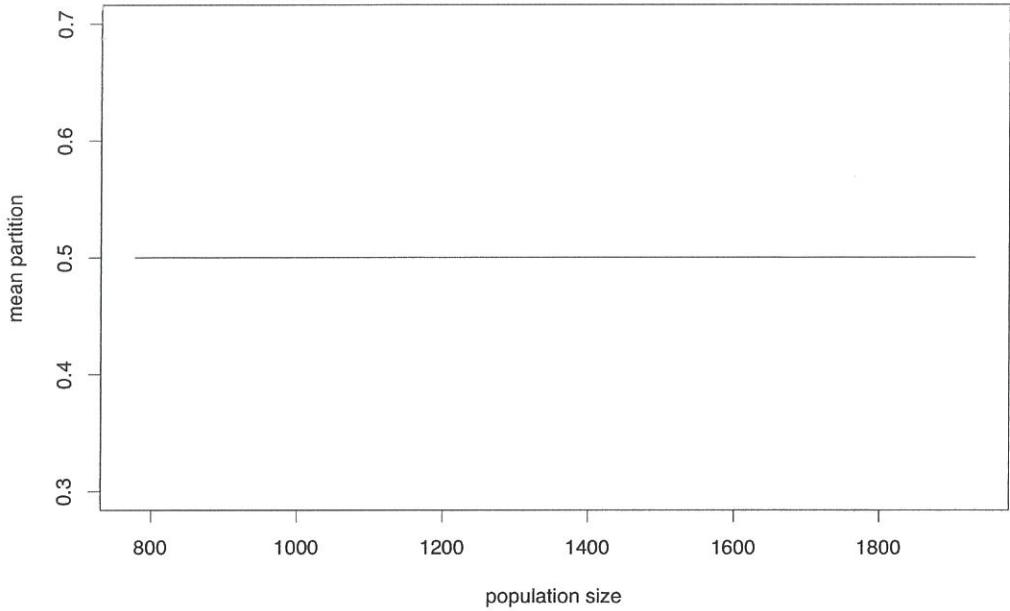


Figure 8: CV's for different values of c_d . Right of the dotted and solid line non plastic and plastic populations go extinct respectively, after which the measured values lose relevance. $c_f = 0.5$, $c_m = 0.1$, $s_c = 10.0$.

Relation between population size and mean partition without plasticity



Population size in time without plasticity

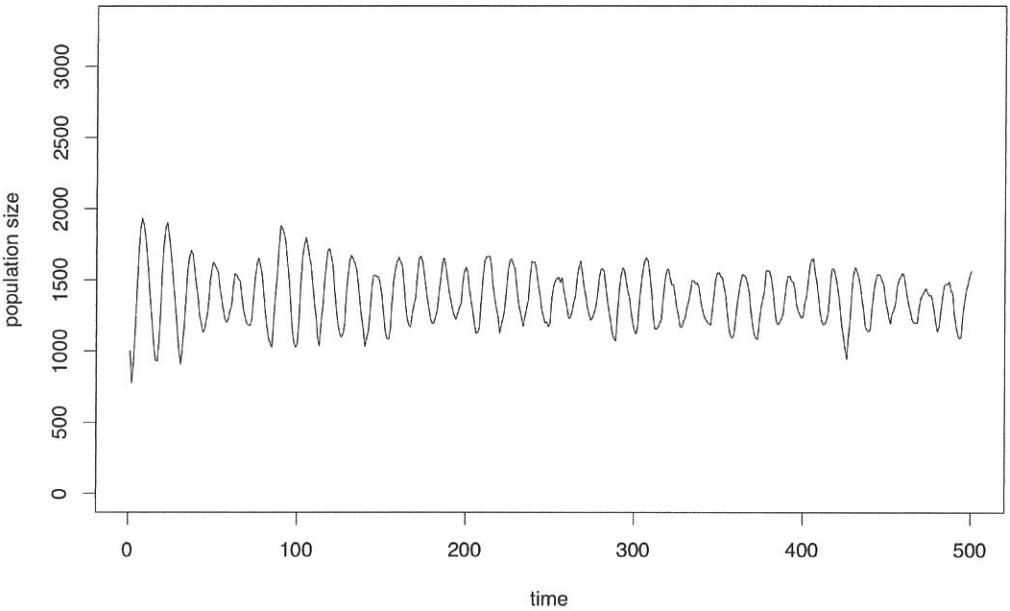


Figure 9: Phase plane (left) and time plot(right) of a non plastic simulation, i.e. the partition of resources to growth is constant. Therefore mean partition is independent of population size. $c_f = 0$, $c_d = 0.15$, $c_m = 0.08$, $c_p = 0$, $s_c = 10.0$.

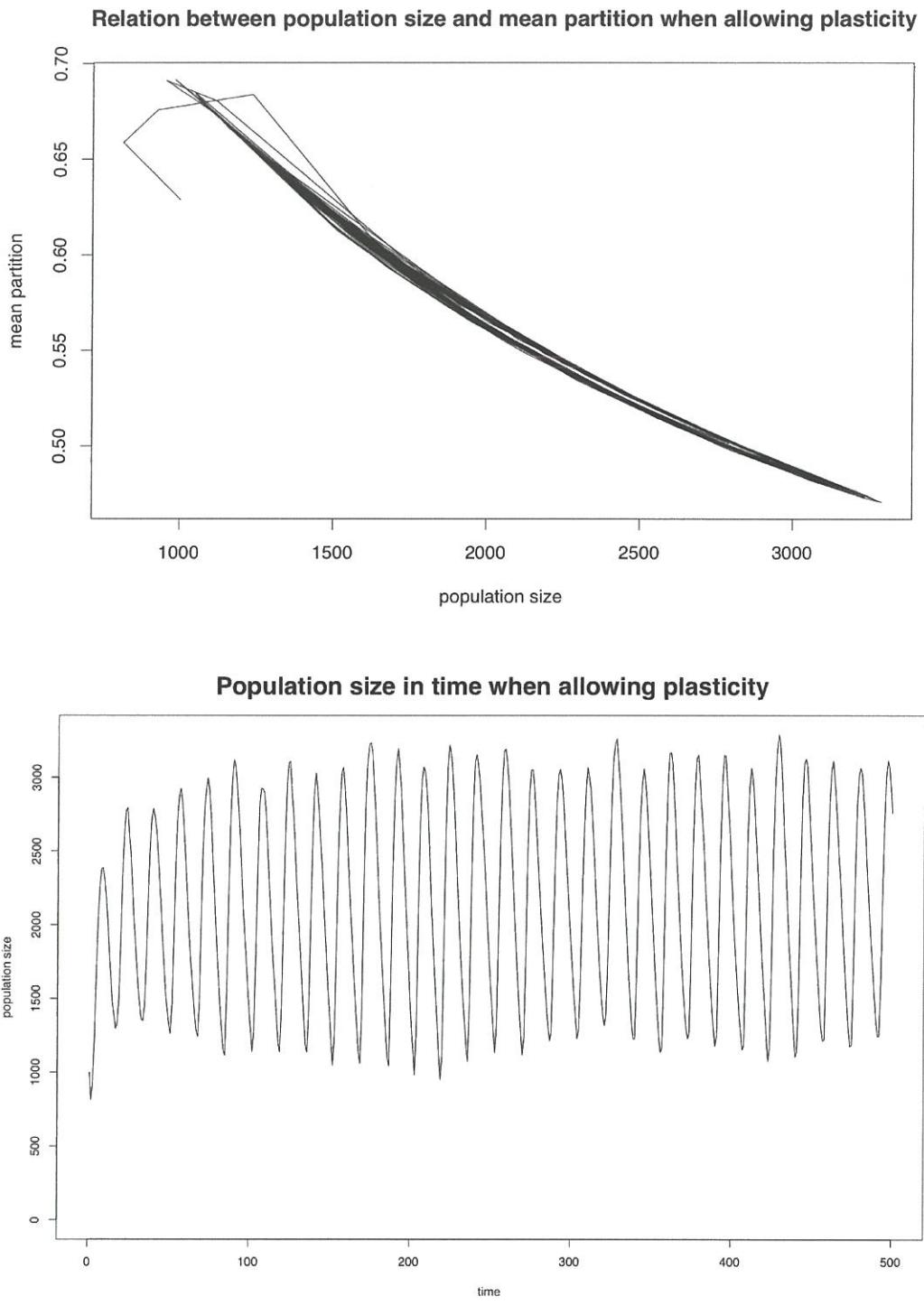
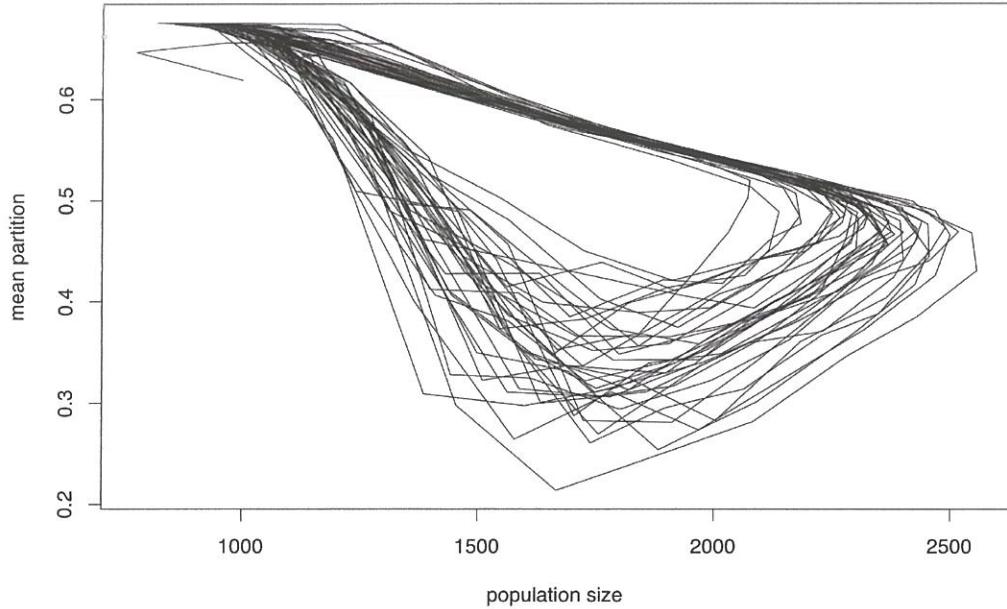


Figure 10: Left) Interaction between population size and partition. Right) the corresponding time plot.
 $c_f = 0$, $c_d = 0.15$, $c_m = 0.0$, $c_p = 1$, $s_c = 10.0$.

Relation between population size and mean partition when allowing plasticity



Population size in time when allowing plasticity

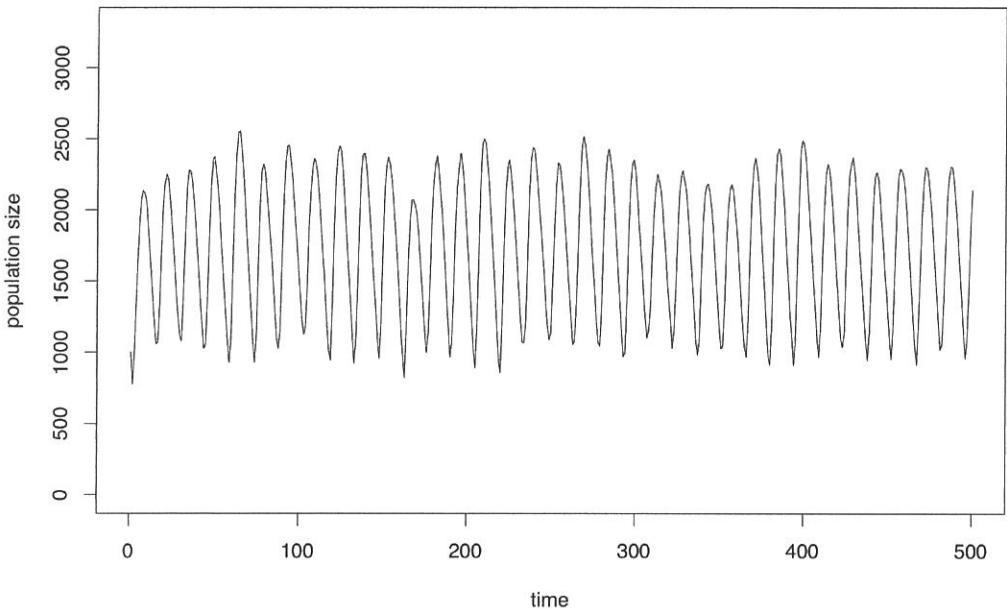
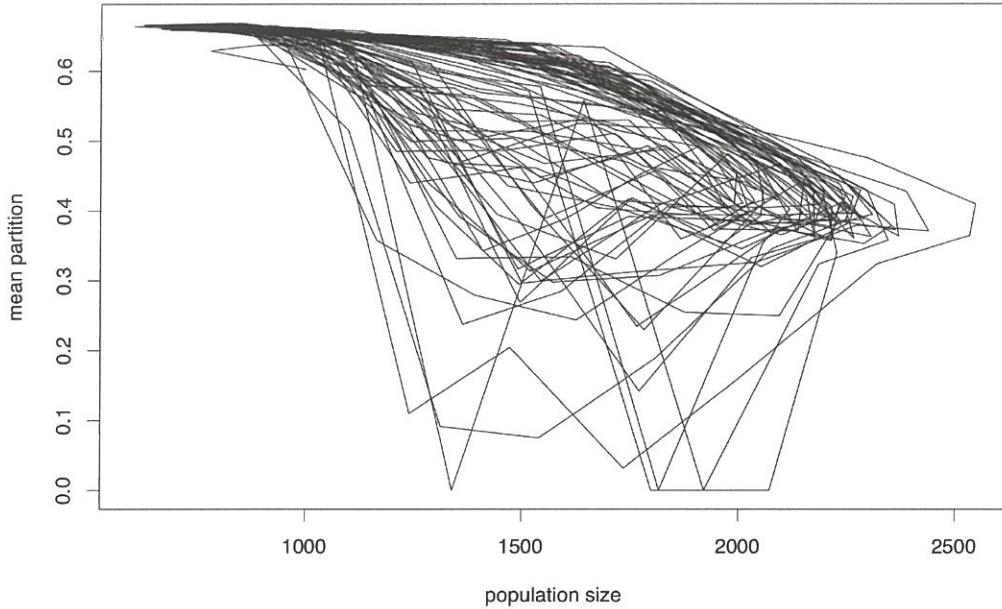


Figure 11: Left) Interaction between population size and partition. The trajectory is clockwise. Right) the corresponding time plot. $c_f = 0$, $c_d = 0.15$, $c_m = 0.08$, $c_p = 1$, $s_c = 10.0$.

Relation between population size and mean partition when allowing plasticity



Population size in time when allowing plasticity

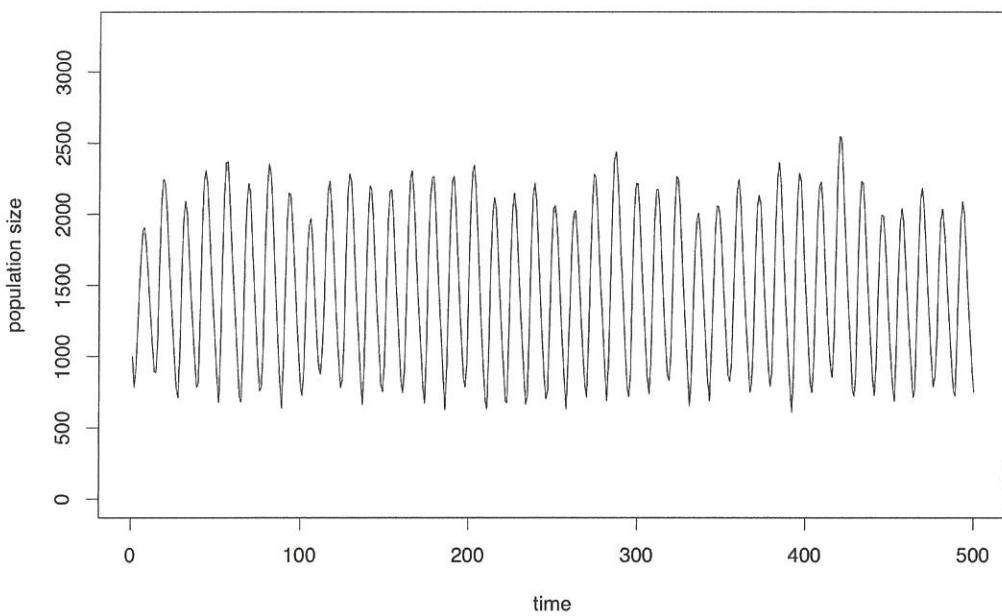


Figure 12: Left) Interaction between population size and partition. The trajectory is clockwise. Right) the corresponding time plot. $c_f = 0$, $c_d = 0.15$, $c_m = 0.13$, $c_p = 1$, $s_c = 10.0$.

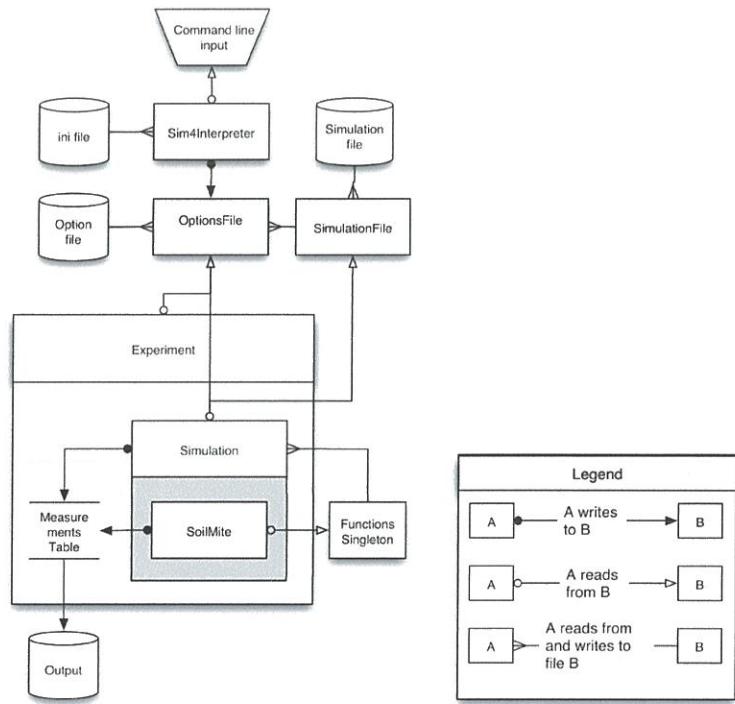


Figure 13: Program overview

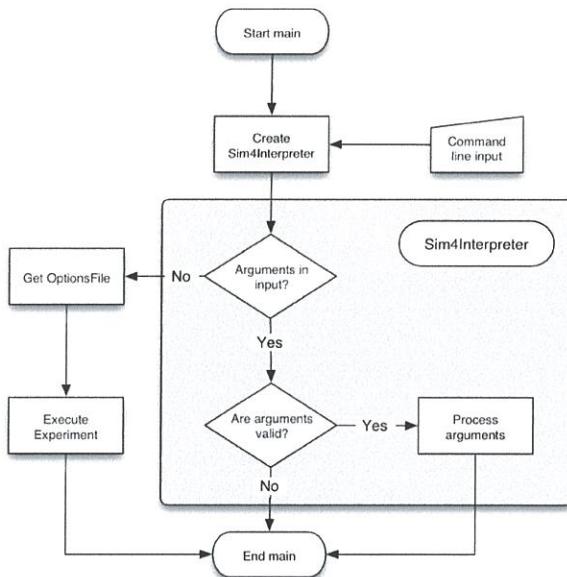


Figure 14: Sim4Interpreter flowchart

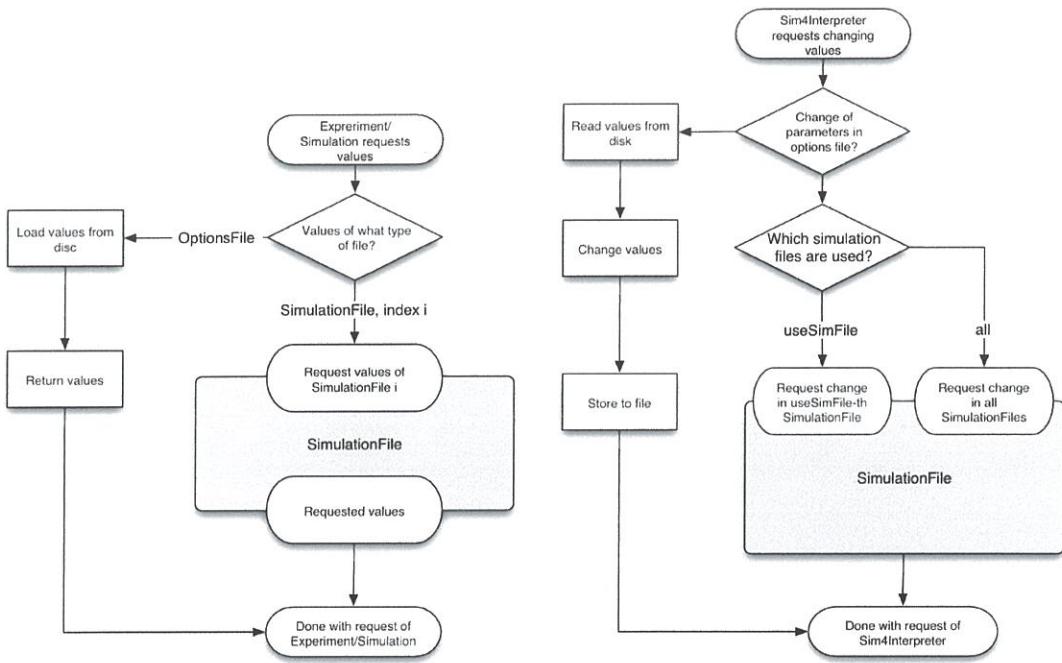


Figure 15: OptionsFile flowcharts. Left visualizes the reading of its data by Experiment and Simulation. Right denotes the modification on OptionsFile requested by Sim4Interpreter.

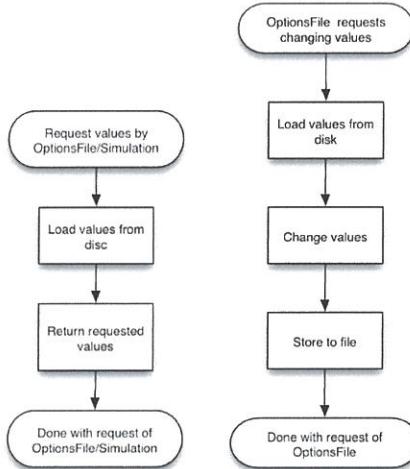


Figure 16: SimulationFile flowcharts. Left visualizes the reading of its data by OptionsFile/Simulation. Right denotes the modification on SimulationFile requested by OptionsFile.

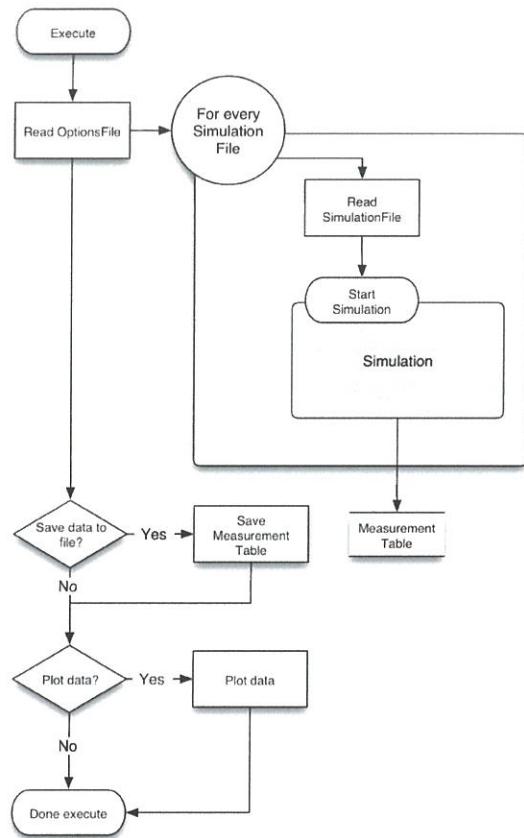


Figure 17: Flow chart of Experiment

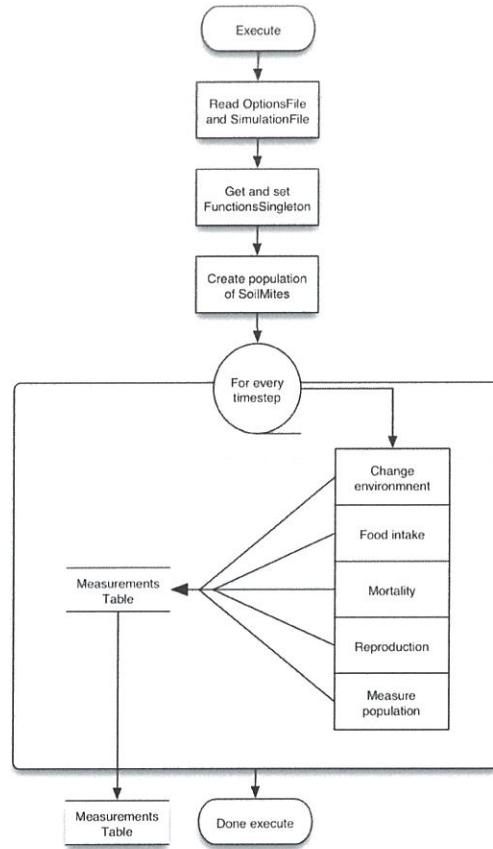


Figure 18: Simulation flowchart

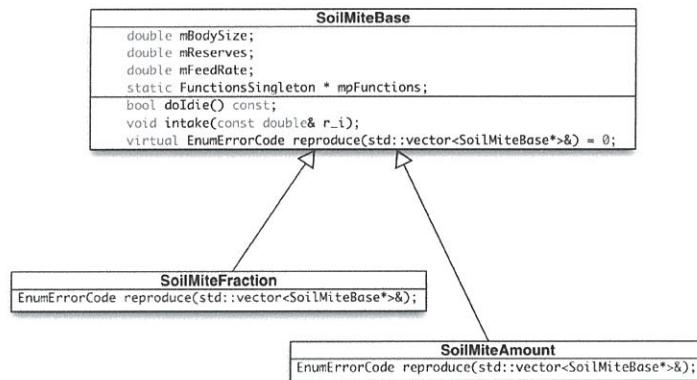


Figure 19: SoilMite UML. The most important member variables and functions.

```
FunctionsSingleton
static FunctionsSingleton * mpInstance;
std::auto_ptr<StrategyDbase> mpD;
std::auto_ptr<StrategyFbase> mpF;
std::auto_ptr<StrategyGbase> mpG;
std::auto_ptr<StrategyMbase> mpM;
std::auto_ptr<StrategyNbase> mpN;
std::auto_ptr<StrategyPbase> mpP;
std::auto_ptr<StrategyQbase> mpQ;
std::auto_ptr<StrategyRbase> mpR;
std::auto_ptr<StrategySigmaBase> mpSigma;
double mAlpha, mBeta, mGamma;

set...();
get...();
eval...();
```

Figure 20: FunctionsSingleton UML.