

PROJ-H405

---

# Project in Electromechanical Engineering

Discrete-time time-varying delay models for the modelling of insect pest populations

---

*Author :*


Arthur Lots

*Teachers :*

Emanuele Garone

Luca Rossini

2023-2024



## Contents

<b>Introduction</b>	<b>1</b>
<b>1 Generalisation and representation</b>	<b>2</b>
1.1 Loss of information . . . . .	2
1.2 Solution . . . . .	3
1.2.1 DatasetOverview . . . . .	3
1.2.2 Condition 1-15 . . . . .	3
1.2.3 Individual-LifeHistory . . . . .	4
1.2.4 PopulationDynamics . . . . .	5
1.2.5 CohortGrouping . . . . .	6
1.2.6 LifeTables-Statistics . . . . .	8
1.2.7 Plots . . . . .	8
<b>2 Data processing</b>	<b>10</b>
2.1 MATLAB code . . . . .	10
2.2 Data acquisition . . . . .	11
2.2.1 Importing dataset . . . . .	11
2.2.2 Impulse responses . . . . .	11
2.2.3 Whole life cycle . . . . .	12
<b>3 Optimisation</b>	<b>15</b>
3.1 Cohort grouping . . . . .	15
3.2 Step 1: Calculating population dynamics in WLF . . . . .	16
3.2.1 Population dynamics to cohort grouping . . . . .	16
3.2.2 Cohort grouping to population dynamics . . . . .	17
3.3 Step 2: Optimisation of population dynamics . . . . .	17

3.4 Results . . . . . 18

4 Conclusion 20

5 Acknowledgment 21

Life tables are essential tools used by entomologists to study the population dynamics and life history traits of insect species. These tables explore a species by describing its biology from a quantitative and qualitative point of view. They provide valuable insights into the development, reproduction, and mortality rates of insect populations. They provide a method for understanding the population dynamics of insect species, which is crucial for various fields including pest management, conservation biology, and ecological research.

In the first chapter of this report, an important problem with the representation of these life tables will be considered: The lack of a standard in data collection and a great confusion among the different representations. [Ros+24] Also, a loss of information is often present because of not supplementing the raw dataset. A generalisation of data processing will be proposed, using an easy to manage excel file.

In the second chapter, the general representations from this excel file will be used to compose useful data for users. Using the impulse responses of each stage, an attempt will be made to compose the whole life cycle. Using regular data processing, it will be shown that the original plots of the insects life stages can be composed. For this, a MATLAB script is composed.

The third and final chapter of this report will explain what should be done if an insects life cycle could not be noted down individually during tests, but only in cohort grouping. This is representing another loss of information for which extra measurements should be made (instead of repeating the whole experiment individually). These involve solving an optimisation problem to estimate the individual insect lifes, starting from a cohort grouping dataset.

All files and code discussed in this report are publicly available and can be found in the GitHub repository <https://github.com/LotsAr/PROJ-H405>. Any other information can be requested directly to the corresponding author (Arthur Lots) at [arthur.lots@vub.be](mailto:arthur.lots@vub.be).

## 1.1 Loss of information

Life tables are composed by placing different eggs laid on the same day in climactic chambers. Each egg will be monitored at a constant frequency and the life stages of the insect will be noted down. For these first two chapters, it will be supposed this is happening individually (each insect is assigned a number).

Life tables are standardised, but that standardisation might not be complete. One should not only report the mean values and standard errors to describe development times, since this is supposing a standardised Gaussian distribution. Yet, this distribution is only valid if each individual is identical. Because of intraspecific genetic variability of a species, this may not be assumed. As an example, two insects might reach adulthood at the same time. One of them might stay longer in the egg stage, yet less long in the larva stage than the second insect. This is a natural way that is ensuring the prosecution of the species even if an extreme event happens. This is not yet proven, but is an impression from previous experiments [Alt06].

In Figure 1.1, the experimental data of the *Corcyra cephalonica* larvae at 26°C [Ros+21] is compared with the Gaussian distribution composed by the mean and standard error of the dataset [Ros+24]. It can clearly be seen that they do not coincide well. This may lead to wrong interpretations of the insects life cycle. Thus may be concluded that reporting only the average and standard error in the current life tables does not always describe the actual data reliably and should be accompanied by the raw dataset.

Other advantages of sharing the raw dataset are the possibility of carrying out meta-analyses and time saving. Firstly, meta-analyses can improve distributions within a single species by using data from various laboratories, thereby verifying similarities and differences in behavior among organisms within the same taxonomic groups. Currently, such comprehensive analyses are not common due to limited data accessibility. Furthermore, compiling life tables is a lot of work and high cost process, often supported by public agencies dedicated to advancing scientific knowledge. This is another reason to make the data publicly available.

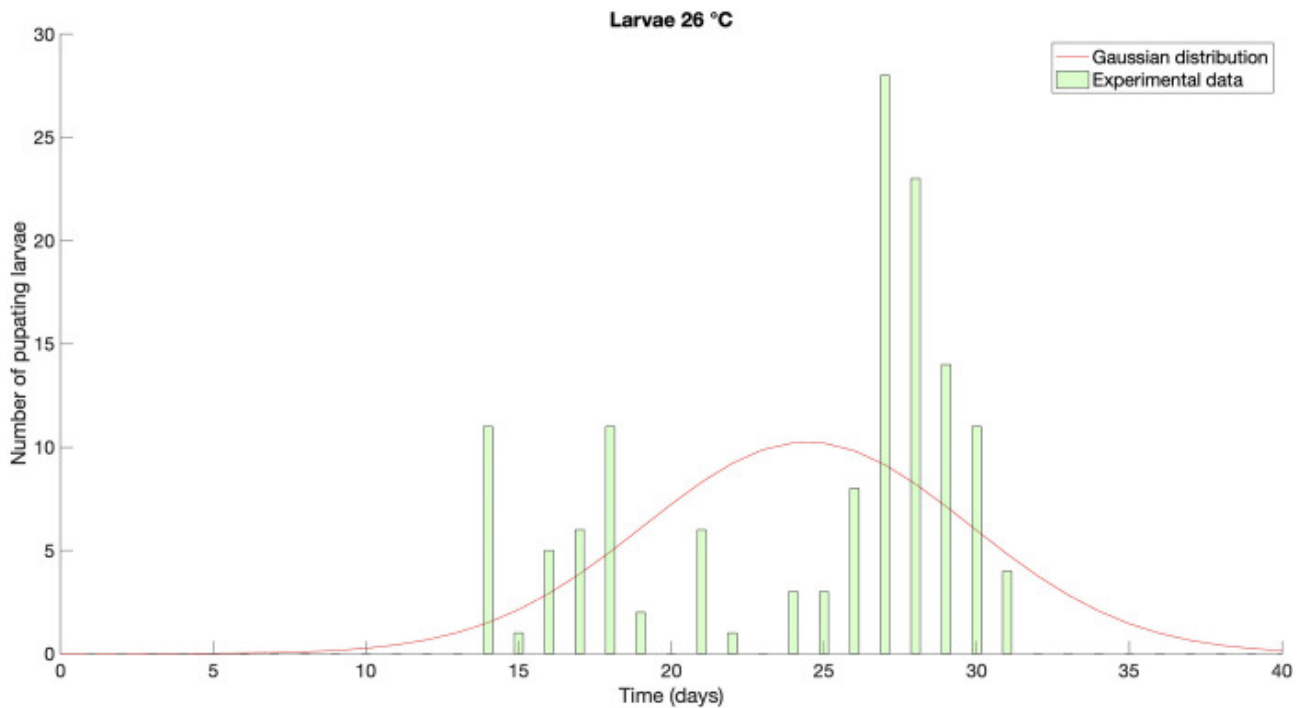


Figure 1.1: Gaussian distribution obtained by the average development time and the standard deviation of the experimental data, vs. raw experimental dataset [Ros+24]

## 1.2 Solution

To encourage scientists to follow the experimental method explained in the previous section, a simple to use excel file was developed that can convert the raw dataset into different representations that could be needed by the entomologist. This file consists of different tabs that will contain information about the dataset, plots, data collection and analysis of the species. These tabs will be explained in detail below.

### 1.2.1 DatasetOverview

In the first sheet, "DatasetOverview", a general overview of the experiments must be given by the user. In this sheet the species and different considered conditions are described. This way, the experiment could be easily reproduced if ever necessary. It also gives extra information to the reader for eventual meta-analyses, as discussed before. Afterwards, all life stages of the insect have to be given a label. This is to improve readability in the data. If a stage is not used, it can be left blank. An example can be seen in Figure 1.2.

### 1.2.2 Condition 1-15

There are 15 condition-sheets, each one representing a different condition as described in the DatasetOverview. The experimental condition depends on the purpose of the experimentation. The most common trials, for instance, are searching to explore the response of the species to different constant temperatures, diets, different constant relative humidities etc. In these sheets, the raw data is noted down. This is the first, and most obvious representation of data. Each day (or each time the measurement is taken), it is noted

Life stages	Label
Egg	E
Larva 1	L1
Larva 2	L2
Larva 3	L3
Larva 4	L4
Larva 5	L5
Larva 6	L6
Pupa	P
Adult males	AM
Adult females	AF
Deads	D

Figure 1.2: An example of the different life stages and their labels

which life stage a specimen is in. For this purpose, the labels given to the stages are used. This increases the speed at which the data collection process can happen. An example of raw data can be seen in Figure 1.3.

Specimen ID	Time (days) →																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	E	E	E	L1	L1	L1	L2	L3	L3	L4	L5	L5	L6	P	P	P	P	AM	AM	AM	AM	AM	D					
2	E	E	E	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
3	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	P	P	P	P	AM	AM	AM	AM	D
4	E	E	E	L1	L1	L1	L2	L3	L3	L4	L5	L5	L6	P	P	P	P	AM	AM	AM	AM	AM	D					
5	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
6	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
7	E	E	E	L1	L1	L1	L2	L3	L3	L4	L5	L5	L6	P	P	P	P	AM	AM	AM	AM	AM	D					
8	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
9	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L4	L5	L5	L6	P	P	P	P	AM	AM	AM	AM	AM	D	
10	E	E	E	L1	L1	L1	L2	L3	L3	L4	L5	L5	L6	P	P	P	P	AM	AM	AM	AM	AM	D					
11	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
12	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
13	E	E	E	L1	L1	L1	L1	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
14	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
15	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
16	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
17	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	P	P	AF	AF	D					
18	E	E	E	E	E	E	E	L1	L1	L2	L2	L3	L3	L3	L4	L4	L5	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
19	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	L6	L6	P	P	P	AM	AM	AM	AM	D
20	E	E	L1	L1	L1	L1	L2	L2	L2	L3	L3	L4	L4	L4	L5	L5	L6	L6	L6	L6	P	P	AF	AF	D			

Figure 1.3: An example of raw data noted: the life stage of each individual insect is written down each day

If a specimen died, the days afterwards nothing is noted anymore. For each condition, there are 365 columns representing 365 days (or one year). Also, up to 1000 rows, or specimens, can be noted. Each row is corresponding to one individual insect. The reason behind 365 days and 1000 insects is because it is wanted to ensure that this file is compatible to explore most of the insect species. Usually the life cycles can range from some days, up to one year.

### 1.2.3 Individual-LifeHistory

A second possible representation is showing the individual life history of each insect. In this representation, it is presented how long every insect stayed in a life stage. When a specimen died during a stage, it didn't really complete it. For this reason, the length of this stage should not be used in averages and thus a difference is made between the stages it survived and the one it died in. For each condition, two different tables are constructed in the Individual-LifeHistory sheet of the excel file, as can be seen in Figure 1.4 and Figure 1.5.

These representations are calculated starting from the raw data. One needs to count how many days an insect is in a certain life stage, which can be done from the raw data in the condition-sheets. Important to

Development - Condition 1										
Insect	E	L1	L2	L3	L4	L5	L6	P	AM	AF
1	3	3	1	2	1	2	1	4	0	0
2	2	4	3	2	3	3	2	2	0	0
3	7	2	2	3	2	2	3	3	0	0
4	3	3	1	2	1	2	1	4	0	0
5	2	4	3	2	3	3	2	2	0	0
6	7	2	2	3	2	2	3	3	0	0
7	3	3	1	2	1	2	1	4	0	0
8	2	4	3	2	3	3	2	2	0	0
9	7	0	0	0	0	0	0	0	0	0
10	3	3	1	2	1	2	1	4	0	0

Figure 1.4: The development table of the individual life history of the raw data as seen in Figure 1.3. It is noted how much time a specimen needed to advance to its next life stage

Mortality - Condition 1										
Insect	E	L1	L2	L3	L4	L5	L6	P	AM	AF
1	0	0	0	0	0	0	0	0	6	0
2	0	0	0	0	0	0	0	0	0	3
3	0	0	0	0	0	0	0	0	4	0
4	0	0	0	0	0	0	0	0	6	0
5	0	0	0	0	0	0	0	0	0	3
6	0	0	0	0	0	0	0	0	4	0
7	0	0	0	0	0	0	0	0	6	0
8	0	0	0	0	0	0	0	0	0	3
9	0	2	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	6	0

Figure 1.5: The mortality table of the individual life history of the raw data as seen in Figure 1.3. It is noted how much time a specimen survived in a life stage before it died

note is that there is no information lost when using this representation. It is perfectly possible to go back to raw data from this representation by doing the inverse process.

#### 1.2.4 PopulationDynamics

In the sheet PopulationDynamics, a third representation is used. This is a differential representation since it can be seen for every day how many individuals went into a stage. Only the new ones are counted, not the total number of insects in that life stage. It should also be noted that this is in a relative timescale, not an absolute. This means that day 0 of a stage (except for the eggs) is actually the day on which the insect entered its previous stage. Using this reasoning, one might see that the first 12 new L1's in Figure 1.6 are at relative day 2. This means that 12 specimens only needed 2 days in the egg stage before entering the first larva stage. Again, following the same reasoning as before, there is another table for the mortalities. An example of this is shown in Figure 1.7.

This representation is useful because it shows the distribution of the development times of each life stage. From this distribution, you can understand which is the minimum development time for a given stage, namely the minimum time needed by the biological processes under the given conditions. With that, also the maximum time and when you expect the peak of the population in that stage can be seen. This is interesting for many control strategies of insects (pesticides, for instance). These are only effective on certain



Stage transition - Condition 1											
Day	Init - E	E - L1	L1 - L2	L2 - L3	L3 - L4	L4 - L5	L5 - L6	L6 - P	P - A	AM - D	AF - D
1	37	0	0	11	0	11	0	11	0	0	0
2	0	12	10	9	23	9	20	12	12	0	0
3	0	12	11	12	9	12	12	8	8	0	0
4	0	0	12	0	0	0	0	0	10	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	12	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

Figure 1.6: The stage transitions of the population dynamics of the raw data as seen in Figure 1.3. It is noted how many new individuals arrive in a stage using a relative timescale

Mortality - Condition 1											
Day	Init-D	E-D	L1-D	L2-D	L3-D	L4-D	L5-D	L6-D	P-D	AM - D	AF - D
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	1	0	12
4	0	0	1	0	0	0	0	0	0	8	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	10	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

Figure 1.7: The mortality of the population dynamics of the raw data as seen in Figure 1.3. It is noted how many individuals died in a stage using a relative timescale

stages. It means that if pesticide is sprayed when the population of that stage is not at the peak, it is not optimally used and probably not solving the problem of the pest at all.

However, a first loss of information is introduced here. As already said by the name, this representation only gives an idea about transitions of the whole population, not of individuals. It is not possible to go from this representation to the previous, nor to the raw data. Yet, since this is an often-used presentation of insect populations, it proves the point made in previous section section 1.1. It is still a very useful representation, but loses the information of the individual life history. For example, it is not possible anymore to see influences of a life stage on a following stage, since it is unknown which insect stayed for how long in these stages.

### 1.2.5 CohortGrouping

The last representation is to be seen in the CohortGrouping sheet. Unlike the PopulationDynamics, this is rather an integral representation. It shows another way of counting raw data, not individual, but in cohort (a cohort is a group of insects, usually of the same species, born within a defined period of time). This is

sometimes done if the insect species can not easily be separated. Regardless of that, it can be obtained from the raw individual data by counting how many insects are in a stage on each day. The table seen in Figure 1.8 is quite diagonal. This is usually the case, since the specimens progress in stages as the time goes by. It can be noted that the total number of insects in a row doesn't always add up to the starting number (here 37). This is because the mortalities are again kept out of the table. They are represented in another table: Figure 1.9.

Development - Condition 1										
Day	E	L1	L2	L3	L4	L5	L6	P	AM	AF
1	37	0	0	0	0	0	0	0	0	0
2	37	0	0	0	0	0	0	0	0	0
3	25	12	0	0	0	0	0	0	0	0
4	13	24	0	0	0	0	0	0	0	0
5	13	24	0	0	0	0	0	0	0	0
6	12	24	0	0	0	0	0	0	0	0
7	12	0	23	0	0	0	0	0	0	0
8	0	12	12	11	0	0	0	1	1	1
9	0	11	12	11	0	0	0	0	0	0
10	0	0	10	12	11	0	0	0	0	0
11	0	0	10	12	0	11	0	0	0	0
12	0	0	0	9	12	11	0	0	0	0
13	0	0	0	9	12	0	11	0	0	0
14	0	0	0	9	12	0	0	11	0	0
15	0	0	0	0	9	12	0	11	0	0
16	0	0	0	0	9	12	0	10	0	0
17	0	0	0	0	0	21	0	10	0	0
18	0	0	0	0	0	9	12	0	10	0
19	0	0	0	0	0	0	21	0	10	0
20	0	0	0	0	0	0	9	12	10	0
21	0	0	0	0	0	0	8	12	10	0
22	0	0	0	0	0	0	0	8	10	12
23	0	0	0	0	0	0	0	8	0	12
24	0	0	0	0	0	0	0	8	0	0
25	0	0	0	0	0	0	0	0	8	0
26	0	0	0	0	0	0	0	0	8	0
27	0	0	0	0	0	0	0	0	8	0
28	0	0	0	0	0	0	0	0	0	0

Figure 1.8: The development of the cohort grouping of the raw data as seen in Figure 1.3. Each day is noted how many specimens are in a certain stage.

While from the individual representation we can directly obtain the cohort dataset, the opposite process is not easy. This is because again, there is a loss of information: the identity of the specimens and thus the individual life history is lost. This problem will be discussed in chapter 3, as it is one of the crucial points of this report.

Mortality - Condition 1										
Day	E	L1	L2	L3	L4	L5	L6	P	AM	AF
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0
10	1	2	0	0	0	0	0	0	0	0
11	1	3	0	0	0	0	0	0	0	0
12	1	3	0	0	0	0	0	0	0	0
13	1	3	1	0	0	0	0	0	0	0
14	1	3	1	0	0	0	0	0	0	0
15	1	3	1	0	0	0	0	0	0	0
16	1	3	1	0	0	0	0	0	0	0
17	1	3	1	0	0	0	0	1	0	0
18	1	3	1	0	0	0	0	1	0	0
19	1	3	1	0	0	0	0	1	0	0
20	1	3	1	0	0	0	0	1	0	0
21	1	3	1	0	0	0	0	1	0	0
22	1	3	1	0	0	0	1	1	0	0
23	1	3	1	0	0	0	1	1	0	0
24	1	3	1	0	0	0	1	1	10	0
25	1	3	1	0	0	0	1	1	10	12
26	1	3	1	0	0	0	1	1	10	12
27	1	3	1	0	0	0	1	1	10	12
28	1	3	1	0	0	0	1	1	10	12

Figure 1.9: The mortality of the cohort grouping of the raw data as seen in Figure 1.3. Each day is noted how many specimens in total died in a certain stage.

### 1.2.6 LifeTables-Statistics

In this sheet, a lot of statistical numbers are given to characterise the different stages of a condition. Using the mean and standard deviation, the Gaussian of each stage could be drawn. Although it was stated prior that this is not a good estimation, it is widely used in life tables articles. For this reason the numbers are still given, alongside with the kurtosis, skewness, mode, median and a survival rate for each stage. All those values provide an indication of the shape of the distribution. By using the excel file, users will thus not have to manually compute them anymore.

### 1.2.7 Plots

The last sheet of the excel file is showing a lot of plots. Each plot depicts the development or mortality of a stage transition. In this way, the 15 discussed conditions can be compared per stage. An example of such a

plot is given in Figure 1.10. It shows the egg to first larva transition for each of 15 conditions. Each condition is drawn in another color.

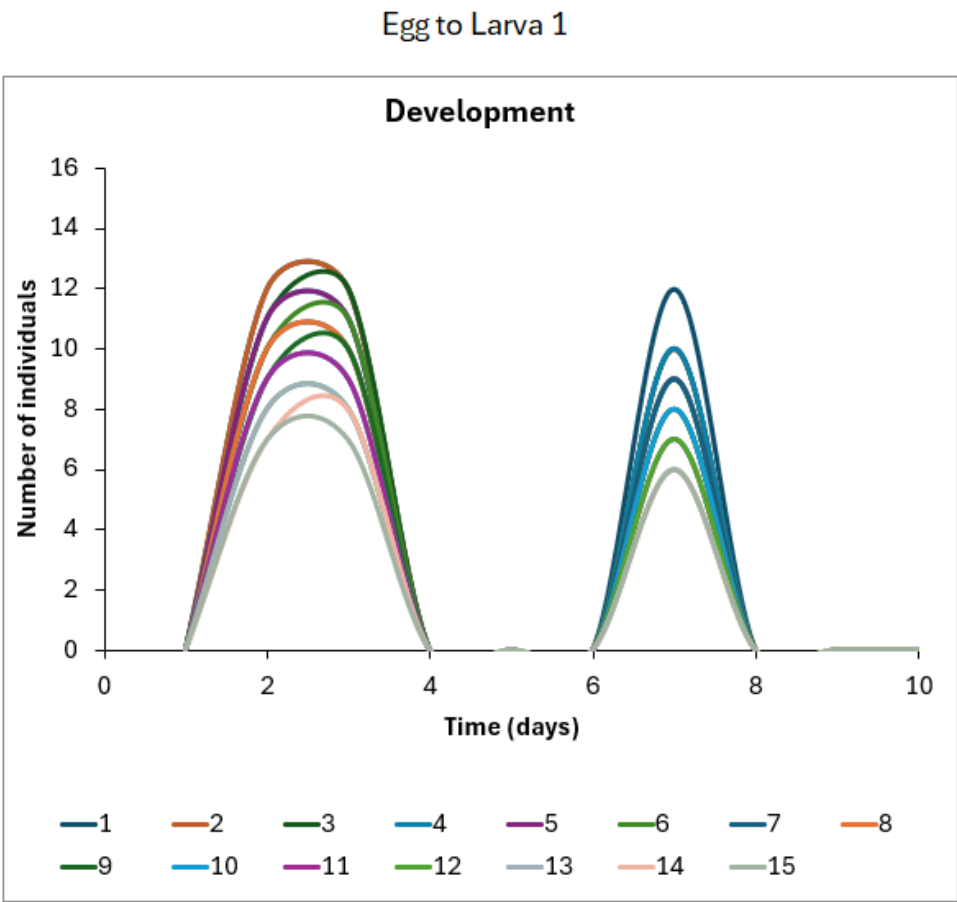


Figure 1.10: The development of the E to L1 transition for all 15 conditions.

In previous chapter a generalised excel file was discussed in which the different representations were composed from the raw data. From the population dynamics it is possible to obtain (by normalising) the impulse response of each stage. Accordingly, the stages can be composed to obtain the whole life cycle (using the time of oviposition as time zero) with convolutions using Equation 2.1.

$$W_{Ei} = W_{E(i-1)} * W_{(i-1)i} \quad (2.1)$$

In this formula, the first subscripts tells to which stage the second subscript is defined. Stage  $i$  is the one that needs to be computed and stage  $i - 1$  the one before. Here,  $W_{(i-1)i}$  is a typical row of the population dynamics and  $W_{Ei}$  would be the whole life cycle. However, Equation 2.1 is not verified. This is one of the main objectives of this report.

## 2.1 MATLAB code

To verify this formula, a MATLAB script was written. It collects the impulse responses from the excel file and is providing their composition. This operation is helpful because it is not fully proven that the composition of the impulse response corresponds to the WLF representation. With this tool, the analysis can be carried out for multiple species and might thus be used to prove it in the future. The script can be found in the repository and consists of four files.

- RunMe\_Main.m: This should be the only file to run
- DataAcquisition.m: In this file all variables are composed from the data out of the excel file (see section 2.2)
- Plots.m: Draws useful figures from the acquired data

- Functions.m: Holds the functions that are used in the other files

To work, the code should be placed in the same folder as the excel file. Then the names at the start of DataAcquisition.m should be changed accordingly to the name choices made by the user. Finally, RunMe\_Main.m could be started, which will consecutively call for DataAcquisition.m and Plots.m.

## 2.2 Data acquisition

The data acquisition from the excel file happens in DataAcquisition.m. To work properly, one should always replace the names at the start of the code by the ones used in its experiment. An example can be seen in purple in Figure 2.1.

```
%Replace these by the titles of the general excel-file
DataFile = 'Life tables test.xlsx';
IndividuallifeHistory = 'Individual-LifeHistory';
PopulationDynamics = 'PopulationDynamics';
CohortGrouping = 'Cohort-grouping';
```

Figure 2.1: An example of chosen names at the start of DataAcquisition.m in purple

Once this has happened, nothing should be changed anymore. Yet, for the sake of completeness, this chapter will discuss all important steps followed by the file. Important to note is that all variables will have names depending on the users choices in its experiments. As an example can be looked to the dataset of all pupae in the first determined conditions. If the user chose to call pupae 'P' and the first condition 'Cond1', the dataset will be called 'P\_Cond1'. Yet, if they used 'Pu' and 'C1', this will also change in the code variables to 'Pu\_C1'. Although this is a slight complication of the code, it is the easiest way for a user to locate their variables.

### 2.2.1 Importing dataset

The data is imported in the first part of the code. The imported representation is Population dynamics (see subsection 1.2.4) because this already uses a relative timescale. The data of individual life histories is lost, yet for this analysis it is not needed. As can be seen in Figure 2.2, the columns (life stages) are saved in variables called Stage\_Condition (such as E\_Cond1). Also, the total number of insects alive in a stage are saved as TotalStage\_Condition (such as TotalE\_Cond1). These will be used for normalization purposes later on.

### 2.2.2 Impulse responses

A next part of the code computes the impulse responses of the normalized data. This facilitates the understanding of how the system, described by a transfer function, responds to the input signal represented by the normalized data. The data is sampled and thus discrete. For this reason, as can be seen in Figure 2.3, the transfer function is also taken to be discrete:  $z$ . The sample time unit corresponds to the

```

% Importing dataset

% Real data from climatic chamber experiments
% Here the new individuals in each stage per day are reported
% The day zero is reset for each stage (scaled population)

%Create all stage variables and assign data
RawDataArray = cell(LenStageNames*length(CondNames), 2);
for k = 1:numel(CondNames)
    for i = 1:numel(StageNames)
        % Form variable name
        varName = [StageNames{i} '_' CondNames{k}];

        % Assign data to the dynamically created variable
        eval([varName ' = dataTable(:, i+(k-1)*14);']);

        % Total insects per condition (survived from egg to adult)
        % This is for checking purposes and to normalize data
        totalVar = ['Total' varName];
        eval([totalVar ' = sum(eval(varName));']);

        %For callback purposes:
        RawDataArray(i+(k-1)*LenStageNames, 1:2) = {eval(varName), eval(totalVar)};
    end
end

```

Figure 2.2: Code snippet that saves the columns of PopulationDynamics to Stage\_Condition and the total of a stage to TotalStage\_Condition

sampling time of the life tables experiment. The function 'Function.ImpResp' can be found in the functions.m file and won't be discussed in detail in the report. It is designed to analyze the impulse response of a system, given an input data array and a transfer function. FirstData represents the transfer function of the stage after processing the entire input sequence, while IR\_Data and IR\_Time represent the impulse response data and corresponding time values of this transfer function. These values are consecutively stored in variables called Out\_Stage\_Condition (Out\_E\_Cond1), IR\_Stage\_Condition (IR\_E\_Cond1) and Time\_Stage\_Condition (Time\_E\_Cond1).

### 2.2.3 Whole life cycle

The previous section calculated the transfer function and impulse responses from each separate stage. Yet, as introduced briefly in Equation 2.1, it is also interesting to view these of the complete life cycle, as in Figure 2.4. To calculate this, some of the previous variables are needed, such as the Total insects eggs (see subsection 2.2.1) and the transfer functions of stages (see subsection 2.2.2). Afterwards, the transfer function of the life cycle until a certain stage (numbered i) can be computed with the following formula:

```

% Second part - Calculation of the transfer functions and impulse reponses
% This first part concerns ONLY the data

% Definition of the transfer function - This is good for the whole
% code, not only for what follows strictly below

z = tf([1 0], [1], 1);

%Calculate Impulse responses

ImpulseDataArray = cell(LenStageNames*length(CondNames), 3);
for k = 1:numel(CondNames)
    for i = 1:numel(StageNames)
        % Form variable name
        IRName = ['IR_' StageNames{i} '_' CondNames{k}];
        TimeName = ['Time_' StageNames{i} '_' CondNames{k}];
        OutData = ['Out_' StageNames{i} '_' CondNames{k}];

        % Assign data to the dynamically created variable
        NormData = NormDataArray(i+(k-1)*LenStageNames, 1);
        [IR_Data, IR_Time, FirstData] = Functions.ImpResp(NormData{1}, z); %See Functions.m
        eval([IRName ' = IR_Data;']);
        eval([TimeName ' = IR_Time;']);
        eval([OutData ' = FirstData;']);

        %For callback purposes:
        ImpulseDataArray(i+(k-1)*LenStageNames, 1:3) = {eval(IRName), eval(TimeName), eval(OutData)};
    end
end

```

Figure 2.3: Code snippet that saves the impulse responses to IR\_Stage\_Condition and Time\_Stage\_Condition and the transfer function of a stage to Out\_Stage\_Condition

$$Tf_{Lifecycle}(i) = Tf_{Lifecycle}(i-1) * Tf_{Stage}(i)$$

$$\text{where } Tf_{Lifecycle}(1) = TotalEggs * Tf_{Stage}(1)$$

Afterwards, the impulse response of this transfer function (stored in LifeCycle\_Stage\_Condition) is computed and stored in IR\_LifeCycle\_Stage\_Condition and Time\_LifeCycle\_Stage\_Condition.

All together, these computed variables can now be used in whatever plots the user would like. Also, it is now possible to compute the WLF representation directly from the data, given that you have the information on the excel file. This supports further validation of the hypothesis of the composition of the impulse responses of the single stage.



```

% Multiplication of the transfer functions from the stage of interest
% to the lower ones!!
% We have to consider the total eggs enter in the life cycle at day
% zero, for this reason it is multiplied by the number of eggs!

% From REAL data
LifeCycleArray = cell(LenStageNames*length(CondNames), 3);
for k = 1:numel(CondNames)
    T = RawDataArray(1+(k-1)*LenStageNames,2); % Start value = TotalEgg of condition
    T = T{1}; % Get value out of cell
    for i = 1:numel(StageNames)
        % Form variable name
        LifeCycleName = ['LifeCycle_' StageNames{i} '_' CondNames{k}];
        IR_LC_Name = ['IR_LifeCycle_' StageNames{i} '_' CondNames{k}];
        Time_LC_Name = ['Time_LifeCycle_' StageNames{i} '_' CondNames{k}];

        %Calculations
        Out = ImpulseDataArray(i+(k-1)*LenStageNames,3); % Out = transfer function of stage
        T = T*Out{1}; % Example: LifeCycle_L1_C1 = TotalE_C1 * (OutE_C1 * OutL1_C1);
        [IR_Data, IR_Time] = impulse(T); % Impulse respons of the life cycle

        %Assign data to variables
        eval([LifeCycleName ' = T;']);
        eval([IR_LC_Name ' = IR_Data;']);
        eval([Time_LC_Name ' = IR_Time;']);

        %For callback purposes:
        LifeCycleArray(i+(k-1)*LenStageNames, 1:3) = {eval(LifeCycleName), eval(IR_LC_Name), eval(Time_LC_Name)};
    end
end

```

Figure 2.4: Code snippet that saves the impulse response of the life cycle to IR\_LifeCycle\_Stage\_Condition and Time\_LifeCycle\_Stage\_Condition and the transfer function to LifeCycle\_Stage\_Condition

### 3.1 Cohort grouping

In previous chapters, all representations (Individual life history, Population dynamics and Cohort grouping) were composed starting from the raw individual data. This is typically the case. However, for some species it is not possible to gather the data individually. For example, a cohort of insects might not be able to be reared individually, or might need the host plant as a feeding substrate. This second case makes it difficult to constrain the individuals during their life cycle. Its data can only be gathered in cohorts, by counting on each day how many insects are in which life stage. Thus, we are not starting from the individual dataset, but from a cohort grouping dataset.

This poses a new problem: How is it possible to get the population dynamics representation? The population dynamics is written in the relative timescale, which still holds a little information about the individuals. However, the cohort grouping is composed in WLF (time zero is the time of oviposition). Because of this, there is a loss of information going from the population dynamics to the cohort grouping and the inverse transformation is not unambiguously determined. It is thus asking for an optimisation.

The following sections will try to solve this optimisation problem by splitting it in two steps:

- Firstly calculating the population dynamics representation in the WLF timescale. This can be done without any need for optimisation (section 3.2).
- Afterwards, this representation will be used to optimise the population dynamics representation with relative timescales (section 3.3).

This method will be explained in detail and validated with real data [Ros+21].

## 3.2 Step 1: Calculating population dynamics in WLF

### 3.2.1 Population dynamics to cohort grouping

To determine how to go from cohort grouping to population dynamics in WLF, first it is needed to study the usual transformation from population dynamics to cohort grouping in detail. To make things clearer, an example shall be studied: In Table 3.1 an example of the population dynamics of an insects first two stages can be seen. As usual, only the new insects to the stage are noted (as mentioned before in subsection 1.2.4). Also note that, in contrary to the notation used in the excel file, this table uses WLF-notation. This means that time 0 is in this case the absolute oviposition time and not the relative time used in the excel. This is important to keep in mind for the further calculations. In Table 3.2, the same data can be seen in cohort grouping representation. For the sake of keeping things easy, it is assumed that the larvae aren't going to the pupa-stage just yet. Also note that, as can be seen in Table 3.3, on day 6 one insect has died in the egg stage. This is written as 'between egg and larva'  $M_{EL}$ .

Table 3.1: Population dynamics (WLF)

Day	Egg $W_E$	Larva $W_L$
1	37	0
2	0	0
3	0	12
4	0	12
5	0	0
6	0	12

Table 3.2: Cohort grouping

Day	Egg $C_E$	Larva $C_L$
1	37	0
2	37	0
3	25	12
4	13	24
5	13	24
6	0	36

Table 3.3: Population dynamics mortality

Day	$M_{EL}$
1	0
2	0
3	0
4	0
5	0
6	1

The current cohort grouping presentation is not very useful to work with, since the other two tables are looked at from a day-to-day basis. For this reason, the difference from day to day is computed:

$$\Delta C_E = \begin{bmatrix} 37 & 0 & -12 & -12 & 0 & -13 \end{bmatrix}$$

Now it can clearly be seen that:

$$\Delta C_E = W_E - W_L - M_{EL} \quad (3.1)$$

This can be written more generally as

$$\Delta C = AW - M \quad (3.2)$$

where  $A$  is a square  $N \times N$  matrix, where  $N$  is the number of life stages. It has 1's on its diagonal and -1's on its superdiagonal as seen below. This means that it will subtract the next population dynamics stage from the previous one.  $W$  is the matrix composed of the different population dynamics stages in WLF-notation as rows (#Stages x #Days) and  $M$  is the matrix composed of the mortality stages of population dynamics as

rows (#Stages x #Days)

$$A = \begin{bmatrix} 1 & -1 & 0 & & \\ 0 & 1 & -1 & \dots & \\ 0 & 0 & 1 & & \\ & \vdots & & \ddots & -1 \\ & & & 0 & 1 \end{bmatrix} \quad (3.3)$$

Using Equation 3.2, the cohort grouping representation can be composed using the development and mortalities of the population dynamics representation (in WLF)

### 3.2.2 Cohort grouping to population dynamics

Now that it is known how to compose the cohort grouping from the population dynamics using Equation 3.2, this formula can be inverted to do the opposite. Doing this and using the same notation as before, it can be easily proven that

$$W = A^{-1}(\Delta C + M) \quad (3.4)$$

where  $A^{-1}$ , the inverse matrix of  $A$ , is again a square matrix. Moreover, it is an uppertriangular matrix with only ones. of the same size as  $A$ .

## 3.3 Step 2: Optimisation of population dynamics

The WLF population dynamics representations is acquired from the cohort grouping. However, the proper representation of population dynamics is not scaled to the time of oviposition, but to its previous life stage. The previous calculated  $W$  can be noted as  $W_{Ei}$ , since it is scaled to the time of oviposition (Egg stage). For example,  $W_{EP}$  has already been calculated, while  $W_{LP}$  is still unknown. This last one is the proper representation of population dynamics. Using Equation 3.5,  $W_{LP}$  could be optimised since  $W_{EP}$  and  $W_{EL}$  are known. It should again be stressed that this equation is yet to be proven, but this might be a first step in doing this.

$$W_{EP} = W_{EL} * W_{LP} \quad (3.5)$$

This optimisation will minimise the error  $W_{LP}$  given by the cost function seen in Equation 3.6.  $W'_{LP}$  is representing the estimated  $W_{LP}$

$$J_{LP} = W_{EP} - (W_{EL} * W'_{LP}) \quad (3.6)$$

In doing this, some conditions should be respected: Mass conservation and positiveness. The mass conservation constraint holds as condition that the sum of all elements in  $W_{LP}$  should be equal to the number of insects alive. If an insect is alive, it should pass through the life stage and thus be counted in the population dynamics (once). The second constraint considers that every element of  $W_{LP}$  on itself should be positive or zero. As seen before, elements in population dynamics  $W$  can never be negative.

For other stages, the same methodology can be used. For now, a simple case will just be studied of an insect life that just consists of an Egg, Larva, Pupa and Adult phase. Using the previous calculations,  $W_{Ei}$  are already known and  $W_{LP}$  is estimated as  $W'_{LP}$ . The only unknown that remains for having a complete Population dynamics representation is  $W_{PA}$ . Again, this can be computed using the optimisation problem in Equation 3.7.

$$W_{EA} = W_{EP} * W_{PA} = W_{EL} * W'_{LP} * W_{PA} \quad (3.7)$$

As can be seen, there are two different strategies in this case. It is possible to optimise  $W'_{PA}$  from Equation 3.8, since  $W_{EA}$  and  $W_{EP}$  are deterministically computed. Another way is from Equation 3.9, where the same  $W_{EL}$  is used in combination with the optimised  $W'_{LP}$ .

$$J1_{PA} = W_{EA} - (W_{EP} * W'_{PA}) \quad (3.8)$$

$$J2_{PA} = W_{EA} - (W_{EL} * W'_{LP} * W'_{PA}) \quad (3.9)$$

Again, these cost functions are minimised with respect to the constraints of mass conservation and positivity. It is expected that Equation 3.8 would have better results than Equation 3.9, since  $W_{EP}$  is used directly from the deterministic model instead of computed from an estimated optimisation. However, if the model gives good results, they can both be used.

### 3.4 Results

Using real data gathered by [Ros+21], the claimed solution will be checked. The obtained results can be seen in Figure 3.1, Figure 3.2 and Figure 3.3.

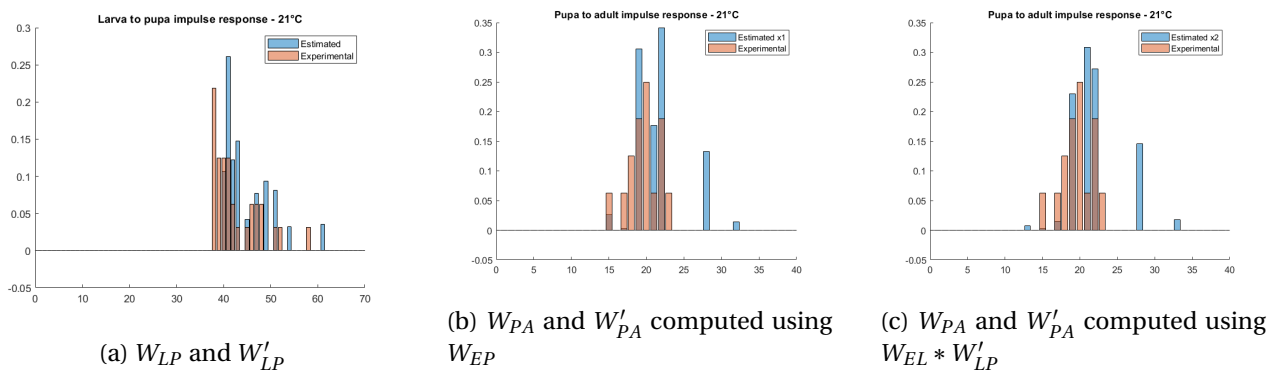
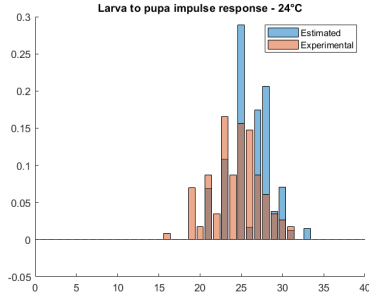
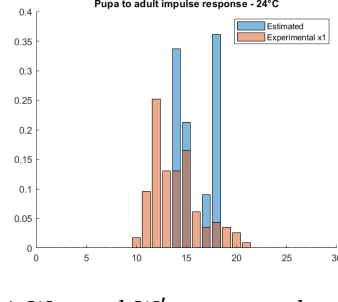


Figure 3.1: Estimations for the data at 21°C

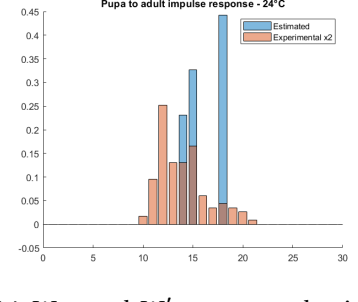
On subfigures (a) can  $W_{LP}$  and its optimised estimation  $W'_{LP}$  be seen. These estimations are quite close to the exact experimental outcome. On subfigures (b),  $W_{PA}$  is estimated as  $W'_{PA}$  using Equation 3.8 with  $W_{EP}$ . On subfigures (c), the same is done using Equation 3.9 using the convolution between  $W_{EL}$  and estimation  $W'_{LP}$ . It can be seen that this gives two different estimations. None of those come close to describing the original  $W_{PA}$ .



(a)  $W_{LP}$  and  $W'_{LP}$

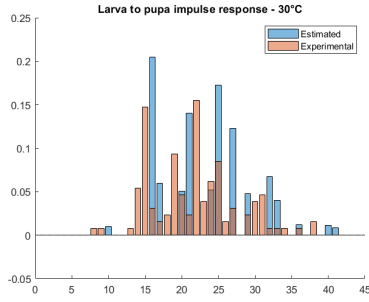


(b)  $W_{PA}$  and  $W'_{PA}$  computed using  $W_{EP}$

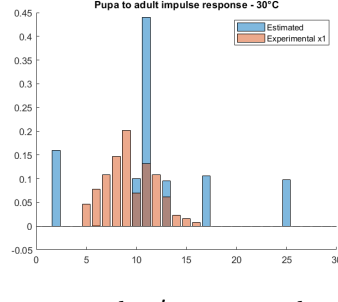


(c)  $W_{PA}$  and  $W'_{PA}$  computed using  $W_{EL} * W'_{LP}$

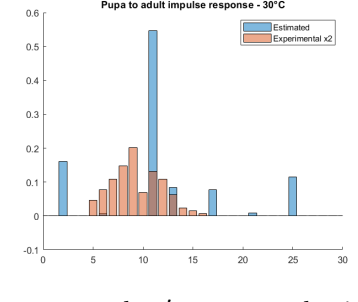
Figure 3.2: Estimations for the data at 24°C



(a)  $W_{LP}$  and  $W'_{LP}$



(b)  $W_{PA}$  and  $W'_{PA}$  computed using  $W_{EP}$



(c)  $W_{PA}$  and  $W'_{PA}$  computed using  $W_{EL} * W'_{LP}$

Figure 3.3: Estimations for the data at 30°C

Based on these results, it is not possible to say whether or not Equation 2.1 is actually proven. Using the optimisation of  $W_{LP}$ , it might be suggested that it in fact is. Although, the optimisation problems of  $W_{PA}$  would suggest otherwise. It would be advised to carry out this same analysis on larger datasets of which the impulse responses (population dynamics) and cohort grouping data is available. It might also be the case that the formulation only works for certain insect populations or life stages.

In the first chapter of this report, the lack of a standard in data collection has been handled. An easy to manage excel file has been proposed, converting raw individual datasets into three representations: Individual life history, population dynamics and cohort grouping.

Together with this file, a MATLAB script has been composed. In the second chapter, an attempt was made to compose the whole life cycle from the convolution of the impulse responses of each stage. It is now possible to compute the WLF representation directly from the data on the excel file. This could support further validation of this hypothesis.

The third and final chapter of this report started from data in cohort grouping. To obtain the population dynamics representation, the problem was split in two parts: A deterministic computation of population dynamics in WLF timescale and from there an optimised computation to the relative timescale. This involved the inverse problem of chapter 2. Based on results from real data, it can be concluded that the optimisation could be successful, although it is highly advised for further research.

## Acknowledgment

For this report, I am extremely grateful to Luca Rossini for supervising and contributing my work. His expertise on life tables and mathematical models helped increasing my understanding. Thanks to the shared dataset, the core of this analysis could be made clearer. Many thanks also to him and Emanuele Garone, who proofread and approved this report.



## Bibliography

- [Alt06] Altukhov, Yuri P. (2006). *Intraspecific Genetic Diversity*. Springer Berlin, Heidelberg. DOI: <https://doi.org/10.1007/3-540-30963-2>.
- [Ros+21] Rossini, Luca et al. (2021). “Life tables and a physiologically based model application to *Corcyra cephalonica* (Stainton) populations”. In: *Journal of Stored Products Research: Volume 91*.
- [Ros+24] Rossini, Luca et al. (2024). “Life tables in entomology: A discussion on tables’ parameters and the importance of raw data”. PhD thesis. ULB. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0299598>.