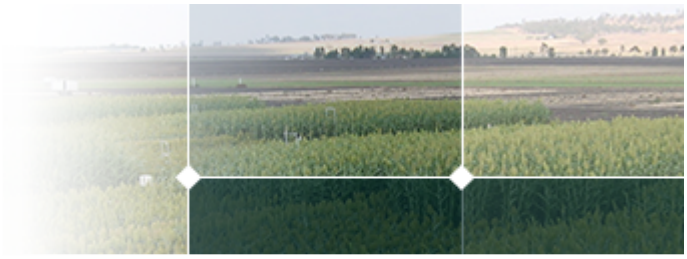


0.0.0.0

The Dymex Pseudo Aphid Tutorials

This is a direct copy of the Dymex tutorial and development of a pseudo Aphid model:

Dymex v2. Insect Tutorial. https://www.hearne.software/getattachment/0ca7c91f-dbe6-4765-8207-4e1b3f1591f2/Insect-Tutorial.aspx. Accessed 1/4/2019.

Tutorial_1_1_PseudoAphid_AgeBasedDevt (Dymex tutorial 1 Introduction)

Tutorial_1_2_PseudoAphid_AgeBasedDevtResults (exploring options to graph results in ApsimX using results from Tutorial_1_1)

Tutorial_2_1_PseudoAphid_ConstantMetTempDevt (Dymex tutorial 2 Temperature-dependent Development)

Tutorial_2_2_PseudoAphid_ConstantTempDevt (exploring other options in ApsimX to set a constant temperature)

Tutorial_2_3_PseudoAphid_MeanFieldTempDevt (Dymex tutorial 3 Changing to Field Temperatures)

Tutorial_3_1_PseudoAphid_DegreeDayDevt (Dymex tutorial 4 Degree Days)

Tutorial_4_1_PseudoAphid_3HrlyDevt (exploring the ApsimX capability to simulate degree days based on three-hourly intervals)

Tutorial_5_1_PseudoAphid_TempDepMortality (Dymex tutorial 5 Mortality)

Tutorial_6_1_PseudoAphid_DensityDepMortality (Dymex tutorial 6 Density-dependent Mortality)

Tutorial_7_1_PseudoAphid_DrynessDepMortality (Dymex tutorial 7 Dryness Dependent Mortality)

Tutorial_8_1_PseudoAphid_DrynessDepMortalitySoil (Dymex tutorial 8 Dryness Dependent Mortality using Soil Moisture)

In Dymex tutorials 9 and 10, management operations are added to the simulations to spray the PseudoAphids. These tutorials have not been rebuilt here since ApsimX provides for spraying as part of manangement activities that can be applied in any ApsimX simulation.

Notes are from the Dymex manual and adjusted to describe the processes required in the APSIM population model builder.

Each tutorial builds upon the previous one.

Unlike Dymex, APSIM provides timing models, climate models, and reporting models so none of these need to be created by the user. However, the model set up described in this section is requred for this and successive tutorials.

** **

1 Tutorial PseudoAphid Age Based Development

This tutorial is the first in the series for modelling the lifecycle of a pseudoaphid. To follow along save the lifecycle example to a working directory on you computer. The simulation **Tutorial_1_0_PseudoAphid_Start** is an empty simulation ready for you to begin by the end of this tutorial you should have created the simulation **Tutorial_1_1_PseudoAphid_AgeBasedDevt**

1.1 Overview

The characteristics of the hypothetical species used in this tutorial were suggested by rose aphids and for convenience, the species is referred to as a ?Pseudo- aphid? in all further discussions: Pseudo-aphids have an adult stage, which they reach 10 days after birth. The adults live for 20 days (i.e. from birth to death is 30 days total). Since the pseudo-aphids are parthenogenetic, all offspring are female and there is no egg stage. The adults begin to produce their 5 progeny after 8 days in the adult stage. The juveniles are produced at a rate which, in a small population, would be equivalent to two births on day 1, four births on day 2, six births on day 3, and so on. The population is free of predators and disease, so all pseudo-aphids only die after 20 days as an adult. Food is assumed to be abundant and has no effect on the population other than to allow it to increase.

In Tutorial_1_1, ApsimX is set up to simulate an insect population for one year. A lifecycle is established that includes a single Juvenile stage that is transferred to an adult stage after ten days. The adult stage is set up with simple mortality and reproduction functions. In this tutorial, all adults die after twenty days. The reproduction function for adults allows all adults to reproduce at a fixed rate after a fixed time.

1.2 Configuring the simulation

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2016
	End simulation =	31/12/2016
Weather		
	Select met file =	Normal.met
Report		
	Report Variables	[Clock].Today
	Report Variables	[Clock].Today.DayOfYear
	Report frequency	[Clock].DoReport

Adding and building a report

Add a report to hold the results of your simulation if there is not one present.

Steps

1. Right click FIELD object,
2. Select **add Model > Report**

For most simulations, the change in results over time is of interest. To add a time component to your report,

In the **Report module** Type "[Clock]." *(be sure to include the full stop after the closing square bracket)*. Options will appear - select "Today" to report today's date. Another useful time component to add to your report is the day of the year. To so this in the Report module, Type "[Clock]." *(be sure to include the full stop after the closing square bracket)*. Options will appear - select "Today". and then type another "." full stop. More options will appear and scroll down to "DayOfYear" to report the Julian day.

To tell the model when to report type clock in the Reporting Requency section and select doReport This will do a daily report

[clock].DoReport

1.3 Adding a Lifecycle

Steps

1. Go to the FIELD object,
2. Right click and select **add Model > LifeCycle > LifeCycle**

The LifeCycle module will be added to your simulation under the Field

1.4 Adding a LifeCyclePhase (Juvenile)

Steps

1. Right click on the **LifeCycle** object
2. Select **add Model > LifeCycle > LifeCyclePhase**
3. Rename the LifeCyclePhase **Juvenile** by clicking on the objects name

Once the Juvenile LifeCyclePhase has been named its atributes need to be defined. These atributes can be defined by **Functions, Constants or Processes**

1.5 Mortality

AddModel	Function Type	Name	value
-	Constant	Mortality	0

Steps

1. Right click on the **Juvenile** object
2. Select **add Model > Functions > constant**
3. Rename constant **Mortlity** set value to **0**

1.6 Reproduction

AddModel	Function Type	Name	value
-	Constant	Mortality	0

Steps

1. Right click on the **Juvenile** object
2. Select **add Model > Functions > constant**
3. Rename constant **Reproduction ** set value to **0**

1.7 Development

AddModel	Function Type	Name	
-	MultiplyFunction	Development	

The pseudo Aphid develops chronologically and after 10 days becomes an Adult

---

AddModel	Function Type	Name	
-	LinearInterpolationFunction	TransferToAdult	

Steps

1. Right click on the **Development** object
2. Select **add Model > Functions > LinearInterpolationFunction**
3. Rename the function **TransferToAdult**

AddModel	Function Type	Name	Value
-	XYPairs	XYPairs	X = 0, 9.9, 10, 20
-			Y = 0, 0, 1, 1
-	VariableReference	XValue	[Juvenile].CurrentCohort.ChronologicalAge

For the function to work it needs a set of **XYPairs**

Steps

1. Right click on the **TransferToAdult** object
2. Select **add Model > Functions > XYPairs**

An x value or driving variable is required

Steps

1. Right click on the **TransferToAdult** object
2. Select **add Model > Functions > VariableReference**
3. Rename the variable **xValue** then specify an internal Plant variable by typing in the value Field.

The Variable we what is the chronological age of the current cohort for pseudo Aphids

Steps

1. Type **[Juvenile].** once you add the **.**
2. A selection box will pop up select **CurrentCohort**
3. Add a **.** and select **ChronologicalAge**

The Step function, which defines the Juvenile to Adult Stage Transfer, requires two parameters: the **ChronologicalAge** at which the juveniles become adults and the proportion of juveniles which actually become adults during that time step. These parameters are entered as **XYPairs**

1.8 Migration

Function Type	Name	value
Constant	Mortality	0

Steps

1. Right click on the **Juvenile** object
2. Select **add Model > Functions > constant**
3. Rename constant **Migration ** set value to **0**

1.9 ProgenyDestination

LifeCycle Function	Name	Option	Value	
ProgenyDestinationPhase	ProgenyDestination	LifeCycle Added	Lifecycle	
-	-	LifeCyclePhase Added		

The **ProgenyDestinationPhase** describes where progeny go in some cases, such as viruses, the progeny may be part of two lifecycles. In this case they remain in the one lifecycle so the progeny will be added to the current lifecycle

Steps

1. select **LifeCycle** from the dropdown list
2. Since there is no reproduction in this phase the lifecycle phase section is left blank

Document your model

It is good practice to describe your functions and why you have used them. This can be done by adding a memo field to the function that describes where the data came from and what papers you used

Steps

1. Right click on **TransferToAdult > add Model > Memo**

Memos can be added as markdown text see Examples Tutorials to see a summary of the markdown features or prepare your memos in a markdown specific text editor.

_____

1.10 Adding a LifeCyclePhase (Adult)

Steps

1. Right click on the **LifeCycle** object
2. Select **add Model > LifeCycle > LifeCyclePhase**
3. Rename the LifeCyclePhase **Adult** by clicking on the objects name

1.11 Mortality

As the pseudo-aphids all need to die as adults, a **Mortality** process needs to be set. There are three types of Mortality process: **Continuous**, **Establishment** and **Exit**.

Continuous mortality operates throughout the duration of the LifeStage and is the most common type of mortality.

However, certain species pass through life cycle stages where considerable mortality occurs when the organism tries to gain a foothold in its new stage (eg. for ticks attaching to an animal, a large proportion are rejected during their initial attempts to attach). Organisms such as this will require an **Establishment Mortality"** process (which acts only during the move to the new stage) in addition to, or instead of, the Continuous Mortality process.

The **Exit mortality** acts only on those individuals that are leaving the LifeStage to transfer to the next stage.

The pseudo-aphid requires only the **Continuous mortality** process. Since the adults all die at the end of a fixed time period, their mortality-inducing variable is **Chronological Age**, with the effect being defined by a **Step function**. The threshold value (i.e., the age of the adults at death) will be 20 and the constant value will be 1 (i.e, they all die on the 20th day). Note that the **Chronological Age** is a measure of the length of time an individual has stayed within that stage. The pseudo-aphids are dying at the end of a 30- day life span. However, since **Phenological Age** in the Adult stage refers only to the length of time an individual has spent in that stage, 20 days is the correct value of the parameter.

AddModel	Function	Name	Value
-	AddFunction	Mortality	
-	MultiplyFuncton	Mortality	
-	LinearInterpolationFunction	Mortality	
-	XYPairs	XYPairs	x= 0, 19.99, 20, 30
-			y= 0, 0, 1, 1
-	VariableReference	xValue	[Adult].CurrentCohort.ChronologicalAge

Steps

1. Right click on the **Adult** object
2. Select **add Model > Functions > AddFunction**
3. Rename AddFunction to **Mortality**
4. Right click on the **Mortlaity** object
5. Select **add Model > Functions > MultiplyFunction**
6. Rename AddFunction to **Mortality**
7. Right click on the **Mortlity** MultiplyFunction object
8. Select **add Model > Functions > Variable Reference**
9. Rename **VariableReference** to **CohortPopulation**
10. Right click on the **Mortlity** MultiplyFunction object
11. Select **add Model > Functions > linearInterpolationFunction**
12. Rename **linearInterpolationFunction** to **Mortlity**
13. Right click on the **Mortlity** linearInterpolationFunction object
14. Select **add Model > Functions > XYPairs**
15. Rename **XYPairs** to **XYPairs**
16. set xy values in the grid
17. Right click on the **Mortlity** linearInterpolationFunction object

18. Select **add Model > Functions > VariableReference**
19. Rename **VariableReference** to **XValue**
20. Set Variable Reference to chronological age

The step function called **Mortality** tells us what proportion of the population dies on day 20. In this case the step function is either 0 or 1

This is multiplied by the **CohortPopulation** to tell us how many individuals will die, in this case either none on day 19 (x0) or all of them on day 20 (x1)

1.12 Reproduction

The starting point for setting Reproductive parameters is a **Reproduction** function.

Three components are requred:

Fecundity(E),

Fecundity(R)

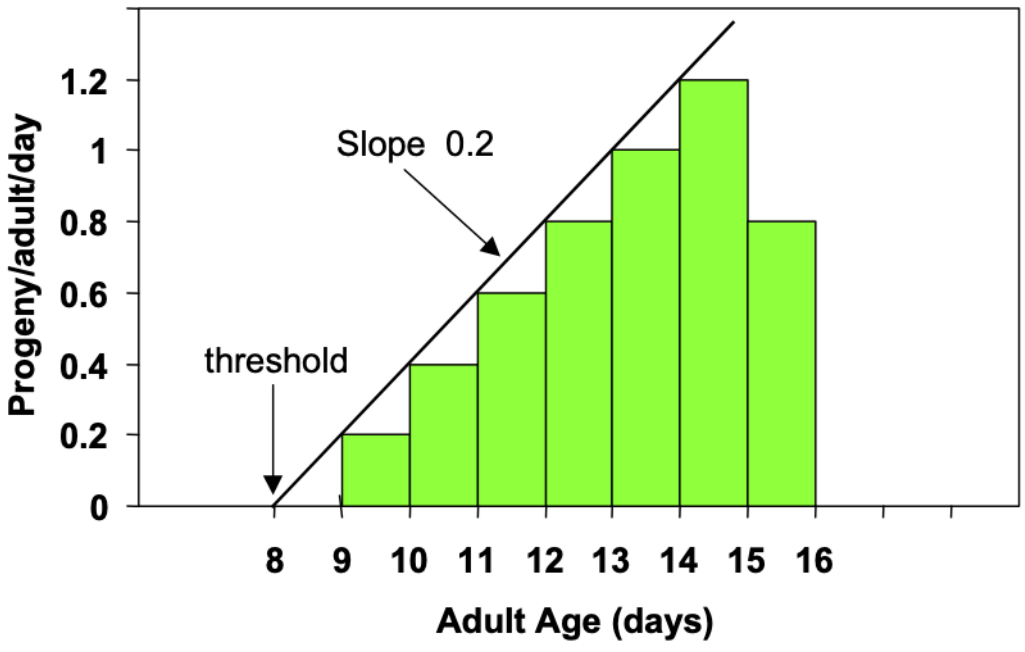
Progeny Production.

Fecundity is the total number of possible offspring that can be produced per individual. The first two components set this number in slightly different ways. **Fecundity(E) is an establishment process**, acting only when individuals first enter the adult stage. This is often the most appropriate way to set the fecundity for an insect i.e., a maximum number of potential offspring (which may depend on how well the insect has fed in the juvenile stages). We will use Fecundity (E) in this example.

Progeny Production determines the rate at which the progeny are produced over time. Note that the Fecundity does not take into account that half the population consists of males, and normally this will need to be modelled by adjusting the value of Fecundity by the sex ratio.

For the pseudo-aphid model, only the Fecundity(E) process is used, and a constant value of 5 was chosen. Fecundity will usually vary with environmental factors, but in this model of the pseudo-aphid, Fecundity will not alter.

Progeny Production Rate defines the rate at which the offspring are produced. For example, some species produce all their progeny at once; others produce batches of offspring over short periods of time separated by intervals, while others may steadily increase production of offspring over a period of time and then decrease the production rate gradually or rapidly back to zero. The Pseudo-aphid of our example is parthenogenetic (only females are present) and so adjustment of the **Progeny Production** by the sex ratio is not a consideration.



While the fractional number of progeny do not apply to an individual pseudo-aphid, they are readily applicable to populations. An important point to remember is that although the pseudo-aphid's progeny production graph

shows that the actual fecundity is exactly equal to the potential fecundity, shown by the area under the curve (shaded), this is in general not so. It is so only because of the deliberately limited nature of this hypothetical organism. The effects of Mortality invariably reduce the actual fecundity.

AddModel	Function	Name	Value
-	MultiplyFunction	Reproduction	
-	VariableReference	VariableReference	[Adult].CurrentCohort.Population
-	MinimumFunction	ProgenyRate	
-	LinearAfterThresholdFunction	ProgenyProduction	x property = [Adult].CurrentCohort.ChronologicalAge
-			x value Trigger = 8
-			Gradient = 0.2
-	Constant	Fecundity	5

Steps

1. Right click on the **Adult** object
2. Select **add Model > Functions > MultiplyFunction**
3. Rename MultiplyFunction to **Reproduction**
4. Right click on the **Reproduction** object
5. Select **add Model > Functions > VariableReference**
6. Rename VariableReference to **VariableReference** set value to **[Adult].CurrentCohort.Population**
7. Right click on the **Reproduction** object
8. Select **add Model > Functions > MinimumFunction**
9. Rename MinimumFunction to **ProgenyRate**
10. Right click on the **ProgenyRate** object
11. Select **add Model > Functions > LinearAfterThresholdFunction**
12. Rename LinearAfterThresholdFunction to **ProgenyProduction**
13. Set values to
14. x Property= **[Adult].CurrentCohort.ChronologicalAge**
15. x value Trigger= **8**
16. Gradient= **0.2**
17. Right click on the **ProgenyProduction** object
18. Select **add Model > Functions > Constant**
19. Rename Constant to **Fecundity** set value to **5**

1.13 Development

Development in the adult phase for this model is covered by the reproduction module. this reproduction is a place holder for future work

AddModel	Function Type	Name	value
-	Constant	Development	0

Steps

1. Right click on the **Adult** object
2. Select **add Model > Functions > constant**
3. Rename constant **Development ** set value to **0**

1.14 Migration

AddModel	Function Type	Name	value
-	Constant	Mortality	0

Steps

1. Right click on the **Juvenile** object
2. Select **add Model > Functions > constant**
3. Rename constant **Migration ** set value to **0**

1.15 ProgenyDestination

LifeCycle Function	Name	Option	Value
ProgenyDestinationPhase	ProgenyDestination	LifeCycle Added	Lifecycle
-	-	LifeCyclePhase Added	Juvenile

Steps

1. select **LifeCycle** from the dropdown list
2. The offspring from the reproductive process will be added to the Juvenile LifecyclePhase

1.16 Set Inital Population

Before we can run the simulation we need an initial population to Start

Description	Values
Select the type of infestation event	OnStart
The name of organisum that infests the zone	LifeCycle
The LifeCyclePhase of the organism when it arrives	Juvenile
Date of infestation (dd-MMM)	
Date of infestation (dd-MMM)	
Chronological Age of Immigrants (days)	0
Physiological Age of Immigrants (0-1)	0

We have set the inital population physiological age to 0 This is unrealistic, but for inital model development and testing, having the organisims magically appear at age zero makes checking the population dynamics easier. When simulating a population this value should be changed to a more realistic average population developmental age.

Steps

1. Right click on the **Field** object and select **infestation**
2. Rename to **InitialInfestationJuvenile**
3. Right click on **InitialInfestationJuvenile** object and **add Model > function > Constant**
4. Rename to **NumberOfImmigrants**
5. Set value to **1**

1.17 Running the Simulation

To run your simulation, right click on the **Simulation name** and select "**Run APSIM**".

1.18 Results

The results from your simulation will appear in the Report model. If there are no resultes check that you have added a reporting frequency.

1.19 Reporting

Reporting	Reporting variable	
1	[Clock].Today	
2	[Clock].Today.DayOfYear	
3	[LifeCycle].TotalPopulation	
4	[LifeCycle].Juvenile.TotalPopulation	
4	[LifeCycle].Adult.TotalPopulation	

Reporting	Reporting variable	
5	[LifeCycle].Adult.Mortalities	

Steps

1. Go to the **Report module** and Type "[LifeCycle]." *(be sure to include the full stop after the closing square bracket)*.
2. Options will appear - select "Juvenile" and then type another "." full stop.
3. More options will appear and scroll down to "TotalPopulation" to report the number of individuals within the Juvenile LifeStage (you will end up with **[LifeCycle].Juvenile.TotalPopulation**).

When examining the results in this simple model you can see how the juvenile population remains constant until it is transferred to the adult population. Then after 8 days the juvenile population starts to increase. If you look at day 30 the inital adult dies.

looking at the results this way is ok but as we get mor complex it will become more difficult

1.20 Ploting Results

Steps

1. Right click on the **Field** object and select **add model> Graph**
2. Rename to **TotalPopulation**
3. Right click on **TotalPopulation** and **add model> series**
4. Rename to **TotalPop**
5. select series and set **Data source** to **Report**
6. set **x:** to **Clock.Today.DayOfYear**
7. set **y:** to **LifeCycle.TotalPopulation**
8. set **Type:** to **scatter**
9. set **Line Type:** to **Solid**
10. set **Marker type:** to **None**
11. repeat steps 5:10 for AdultPopulation, Juvenile population and Adult Mortality
12. Change the clock date to 28/02/2016 and re run the simulation to see a more dynamic view

```

```

Add some additional reporting variables to see the dynamics of cohorts

Reporting
[Clock].Today
[Clock].Today.DayOfYear
[LifeCycle].TotalPopulation
[LifeCycle].Juvenile.TotalPopulation
[LifeCycle].Adult.TotalPopulation
[LifeCycle].Juvenile.CohortCount
[LifeCycle].Adult.CohortCount
[LifeCycle].Juvenile.Populations
[LifeCycle].Juvenile.PhysiologicalAges
[LifeCycle].Adult.Populations
[LifeCycle].Adult.PhysiologicalAges
[LifeCycle].Juvenile.Graduates
[LifeCycle].Adult.Graduates
[LifeCycle].Juvenile.Mortalities
[LifeCycle].Adult.Mortalities

end

1.21 Tutorial PseudoAphid Age Based Development Results

The simulation Tutorial-1-1-PseudoAphid-AgeBasedDevt is the starting point of this tutorial

1.21.1 Examining the output

Run APSIM

The exponential shape of the growth curves makes identifying what was goin on difficult.

Change the length of the simulation to see what is happening.

Set **Clock End Date to 31/3/2016**

Run APSIM

Often it is easier to look at one piece of data per plot Make 3 new plots one to hold **Juvenile** data one to hold **Adult** data and one to hold **Adult.Mortalities**

Steps

1. Right click on the **Field** object and select **add model> Graph**
2. Rename to **AdultPopulation**
3. Right click on **AdultPopulation** and **add model> series**
4. Rename to **AdultPop**
5. select series and set **Data source** to **Report**
6. set **x:** to **Clock.Today.DayOfYear**
7. set **y:** to **LifeCycle.Adult.TotalPopulation**
8. set **Type:** to **scatter**
9. set **Line Type:** to **Solid**
10. set **Marker type:** to **None**

To make the next figure

Right Click on the AdultPopulation figure and select copy write click on the field and select paste Change names and Y value to **LifeCycle.Juvenile.TotalPopulation**

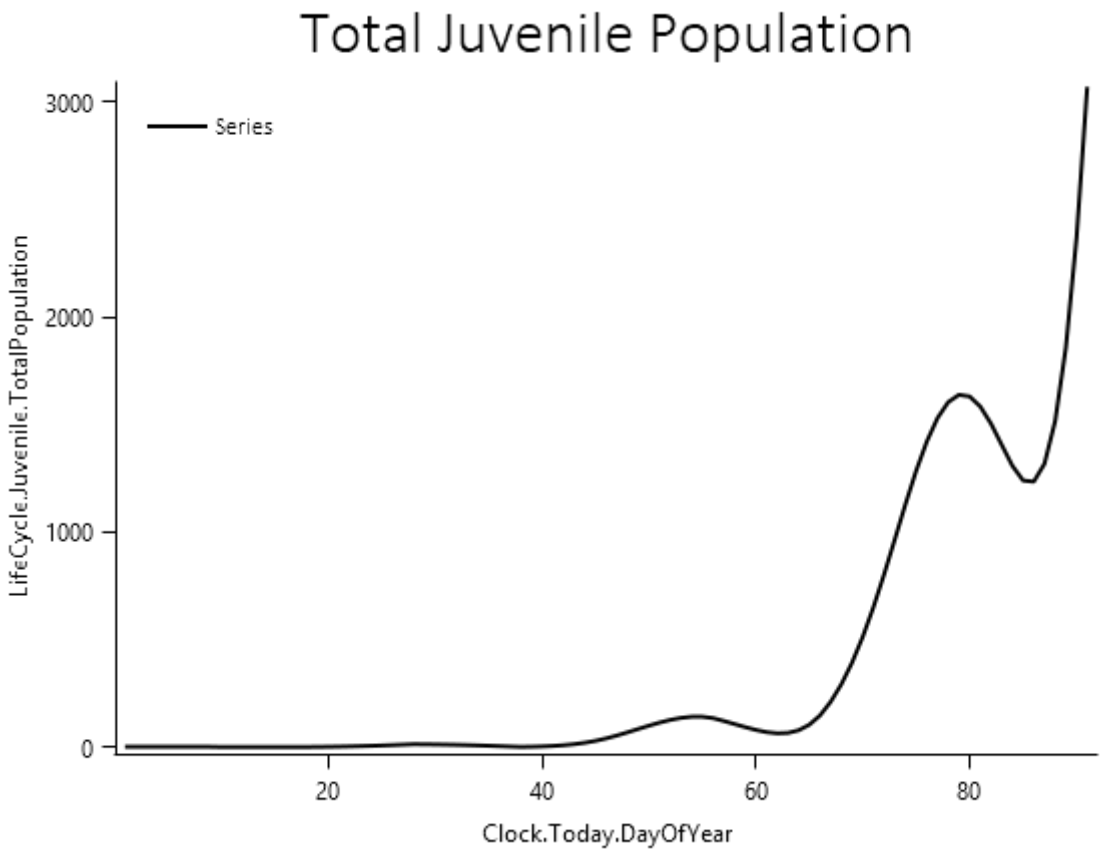
repeat process to make the adult mortality figure

Right Click on the AdultPopulation figure and select copy write click on the field and select paste Change names and Y value to **LifeCycle.Adult.Moralities**

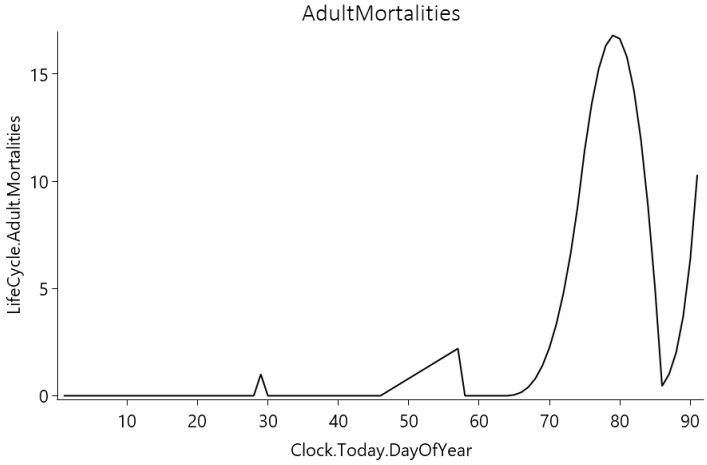
1.21.2 View all Graphs together

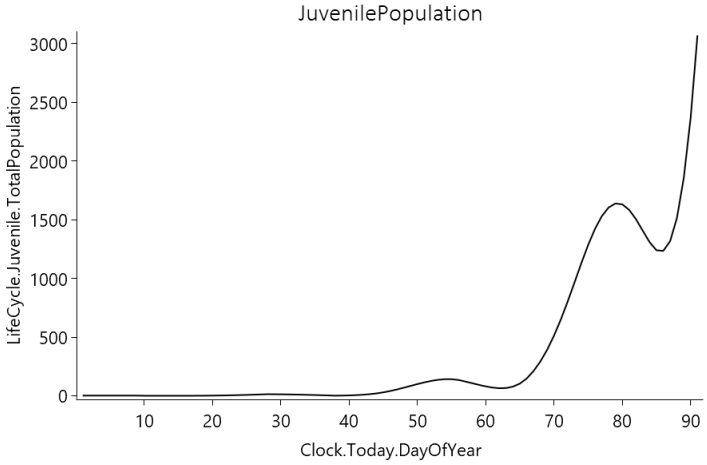
Steps

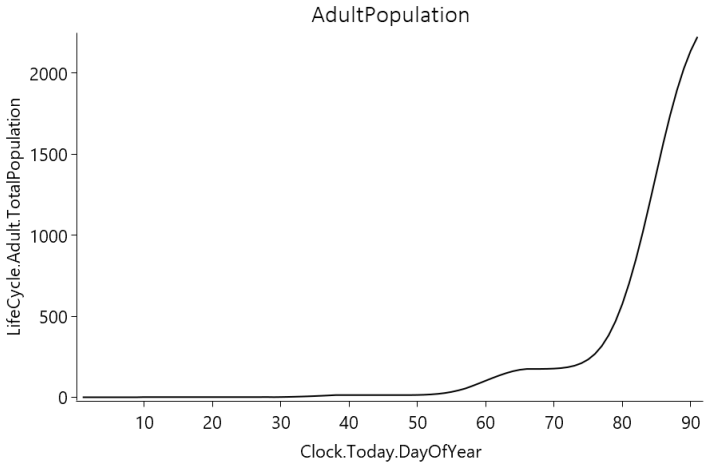
1. Right click on the **Field** object and select **add model > Folder**
2. Rename to **Results**
3. copy and paste each Graph into the Folder
4. click on the folder to see all Graphs at once



1.22 Results







2 Tutorial-PseudoAphid ConstantTempDevt

The simulation Tutorial-1-2-PseudoAphid-AgeBasedDevtResults is the starting point of this tutorial

Chronological and Physiological Age

In Tutorial 1_1 and 1_2, the pseudo-aphid model used chronological age to determine the sequence of development and reproduction. Chronological age is unsatisfactory as the only controlling influence on an insect's LifeCycle because the insect's physiological development (and by implication its reproduction and mortality) is often independent of chronological age and is generally controlled to a large extent by temperature. This tutorial presents an example of how APSIM-LifeCycle can model the pseudo-aphid's development from juvenile to adult when the LifeCycle development is based on physiological age and the rate of that development becomes temperature dependent. Since physiological age now enters all further tutorial models, it is defined below:

Physiological age measures the state of development of an individual with its units generally stated as a proportion (or percentage) of completed development. As an example, the birth of the pseudoaphid could be scaled to 0 and its transition to adult scaled as 1. Since insect development is generally temperature dependent, accumulation of physiological age is usually non- uniform.

2.1 The pseudo-aphid and Temperature

Since the pseudo-aphid's LifeCycle is ‘well known from published papers’, the effects of temperature on development are available and are presented in table form:

2.1 Pseudo Aphid Life table

Temperature (°C)	No. of days to developto adult
10	No Development
15	20
20	12
25	8
30	5

The model used in Tutorial_1_2 remains essentially intact, but the transition from juvenile to adult (currently determined by Chronological Age) now becomes dependent upon Physiological Age, which in turn is dependent upon temperature. All juveniles still become adults when they reach the required Physiological Age. To preserve simplicity of the model, the PseudoAphids are maintained under temperature controlled incubator conditions.

The results of Table 2-1 can be amended to display rate of development per day. This is done by calculating the reciprocal of the number of days taken to develop to adult (which assumes that the value “1' represents the physiological age of an adult). For example, suppose an organism takes 50 days to develop from egg to adult; its rate of development per day would therefore be 0.02 (ie. 0.02 * 50 = 1). Table 2.2 shows the results for the pseudo-aphid.

2.2 Pseudo Aphid Life table with rate of development

Temperature (°C)	No. of days to developto adult	Rate of Developmentper day
10	No Development	0
15	20	0.05
20	12	0.08
25	8	0.125
30	5	0.2

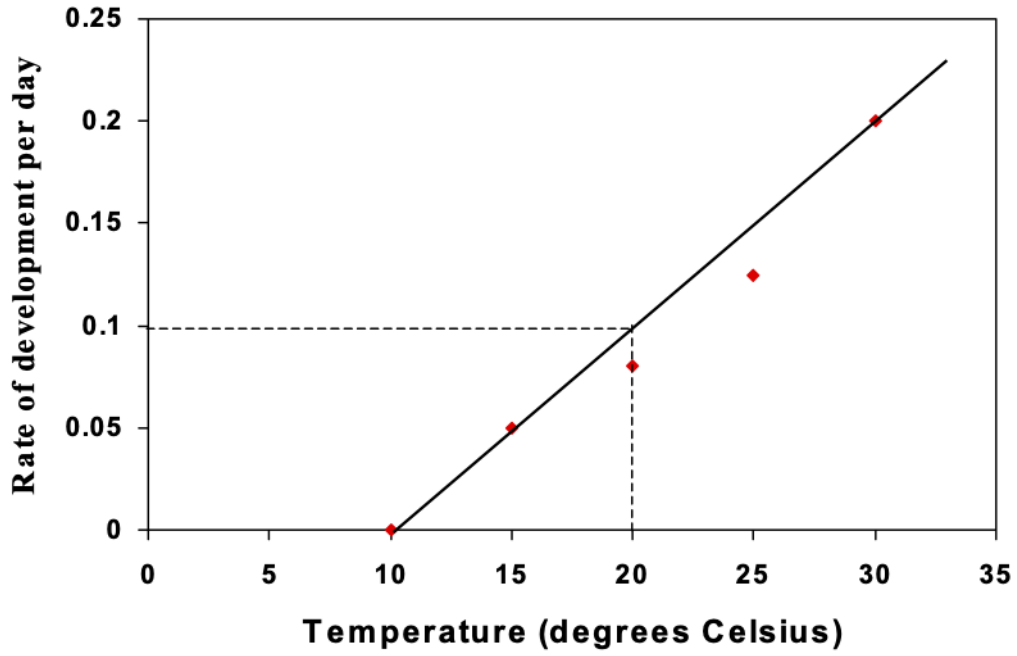


Figure 2-1 Rate of development of juvenile Pseudo-aphids

An inspection of Figure 2-1 shows that the development threshold is 10°C, while the slope of the graph is quite close to 0.01. These values will be used for this tutorial.

2.1.1 Simulation set up

This tutorial requires the following simulation set up.

This simulation takes place in a controlled temperature lab so the weather file has been changed to remain at a constant 20°C

2.1.2 Clock

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2016
	End simulation =	31/12/2016
Weather		
	Select met file =	20°C.met

2.1.3 Modifying Development

AddModel	Function	Name	Value
-	AddFunction	Development	
-	LinearAfterThresholdFunction	DevelopmentRate	x property = [weather].MeanT
-			x value Trigger = 10
-			Gradient = 0.01

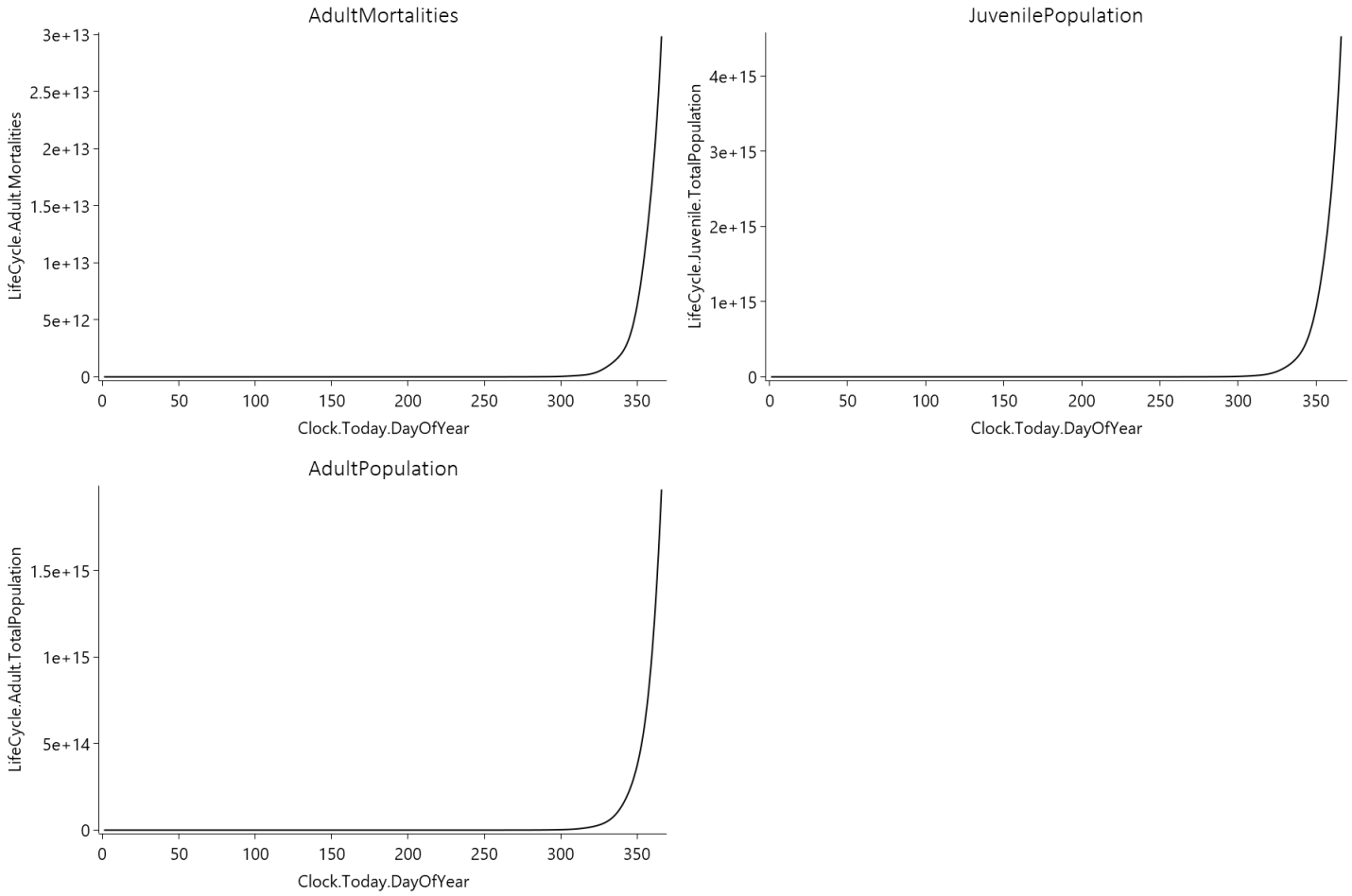
Steps

1. Expand **Juvenile** Node

2. Delete **Development**
3. Right click on the **Juvenile** object
4. Select **add Model > Functions > AddFunction**
5. Rename AddFunction to **Development**
6. Right click on the **Development** object
7. Select **add Model > Functions > LinearAfterThresholdFunction**
8. Rename LinearAfterThresholdFunction to **DevelopmentRate**
9. Set values to
10. x Property= **[weather].MeanT**
11. x value Trigger= **10**
12. Gradient= **0.01**
13. Select **Development** and use control up Arrow to move it above **Reproduction**

Run APSIM

2.2 Results



2.3 Tutorial-PseudoAphid_ConstantTempDevt

The simulation Tutorial-2-1-PseudoAphid_ConstantMetTempDevt is the starting point of this tutorial

The phenology of diseases, pathogens and insects respond to temperature and much of the research is conducted in controlled temperature incubators and so the ability to simulate a constant temperature is important. Changing the Weather file is one way to do this, however an alternative simpler way is to add a constant **Incubation temperature**.

2.3.1 Simulation set up

Return the clock and met file to what they were in tutorial 1.2

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2016

APSIM Object	Description	Values
	End simulation =	31/12/2016
Weather		
	Select met file =	Normal.met

Steps

1. Right click on **Your simulations name ie Tutorial_2_2**
2. Select **add Model > Functions > constant**
3. Rename AddFunction to **IncubatorTemp**
4. Use **control** and up arrow to position it below **Weather**

2.3.2 Modify Development

Modifying the model to set Juvenile development in response to constant incubator temperature

Steps

In the LifeCycle, expand **Juvenile > Development > DevelopmentRate.** Set **DevelopmentRate** function set: Xproperty to [IncubatorTemp] X value trigger to 10 Gradient to 0.01

Run APSIM

2.4 Tutorial-PseudoAphid_ConstantTempDevt MeanDailyTemp

The simulation Tutorial-2-2-PseudoAphid_ConstantTempDevt is the starting point of this tutorial

2.4.1 Simulation set up

Return the clock and met file to what they were in tutorial 1.2

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2016
	End simulation =	31/12/2016
Weather		
	Select met file =	Normal.met

Steps

Delete The **IncubatorTemp** object or right click on the **IncubatorTemp** and unselect the **Enable** check box

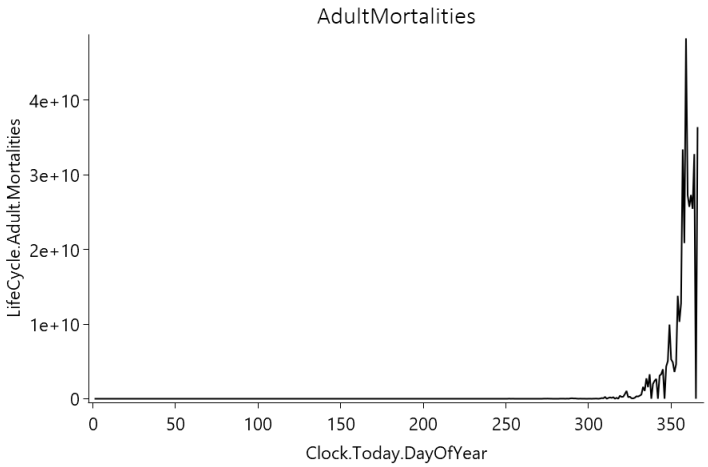
2.4.2 Modify Development

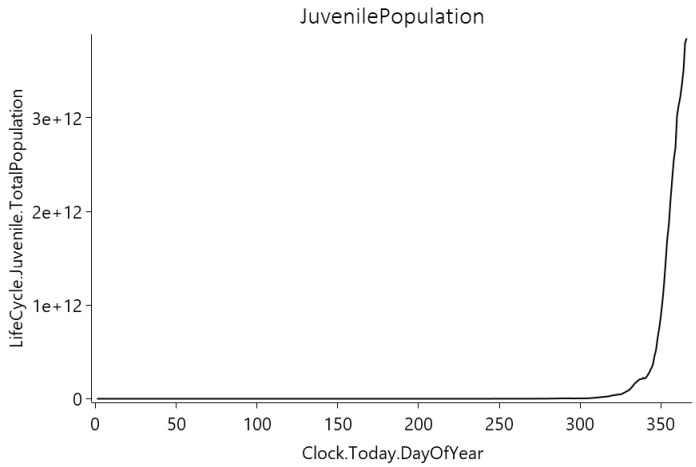
now change the driving variable for temp development

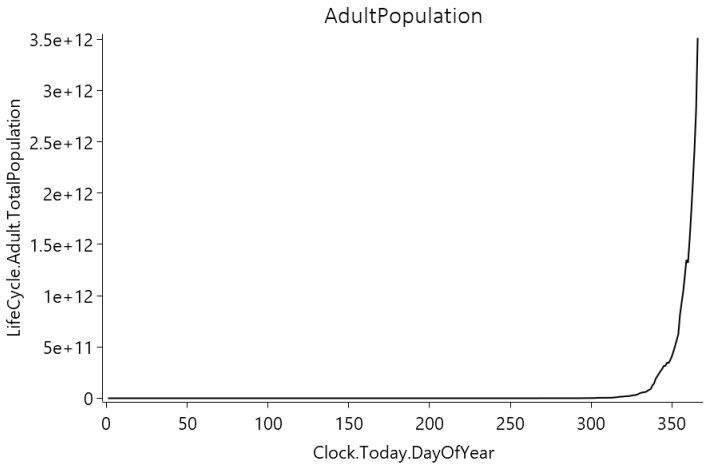
Steps

In the LifeCycle, expand **Juvenile > Development > DevelopmentRate.** Set **DevelopmentRate** function set: Xproperty to [weather].MeanT X value trigger to 10 Gradient to 0.01

2.5 Results







3 Tutorial_PseudoAphid Temperature Based Development Day Degrees

The simulation Tutorial-2-3-PseudoAphid_MeanFieldTempDevt is the starting point of this tutorial

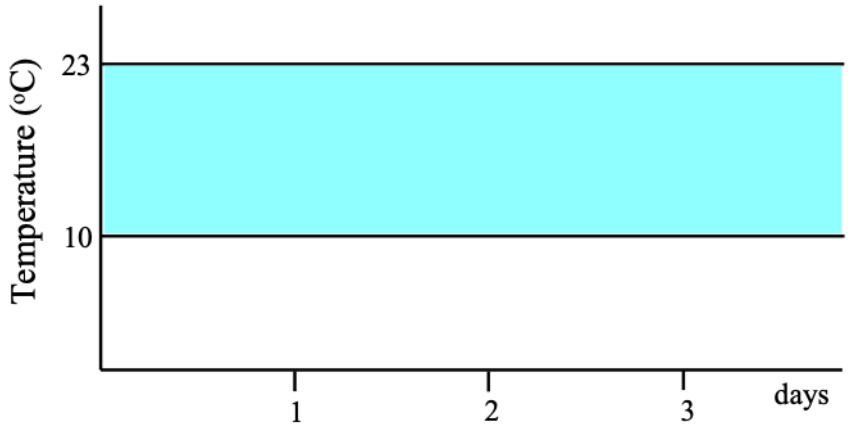
3.1 Degree Days

3.1.1 Introduction

In the Tutorial_2_3, development of the PseudoAphid was linked to temperatures, with actual field temperatures used as input. An average temperature was calculated from the daily minimum and maximum, and this was applied to the development process to drive the increase in Physiological Age. This approach is unsatisfactory in that it does not fully reflect the actual conditions the organisms experienced in the field. A simple example will make this clear: Minimum and maximum temperature values of 8°C and 12°C give an average temperature of 10°C, but so do 0°C and 20°. In the model of Tutorial 2, both these would give the same increase in development, yet to the insect in the field, they are likely to have quite different effects. The later temperature regime would result in a substantial part of the day well above the developmental threshold of 10°C, and is likely to result in more development than the former regime. This concept (termed the ‘day-degree’ concept) is described and applied to the model in this tutorial.

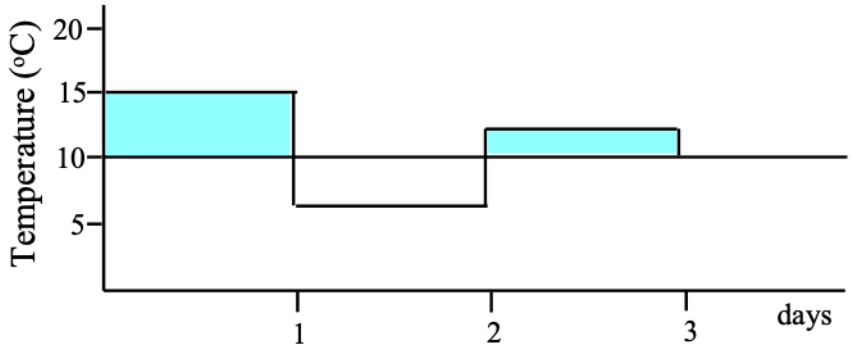
3.1.2 The ‘Degree Day’ Concept

The pseudo-aphid has been given a development threshold of 10°C, which implies that once the temperature rises above that threshold, development proceeds. Suppose that the pseudo-aphid was existing under ideal conditions in an incubator, where the temperature was maintained at a steady 23°C. This situation is illustrated in Figure 3.1, where the area between the developmental threshold and the actual temperature is shaded. The shaded area represents the number of ‘degree-days’ available to the aphid’s development. The name ‘degree-days’ is derived from the fact that the size of the area is obtained by multiplying the height of the shaded rectangle (in degrees) by the width (a number of days). Each day has a temperature that is 13°C above the threshold of 10°C, thus the number of degree days accumulated over a 3-day period would be 13 x 3 = 39.



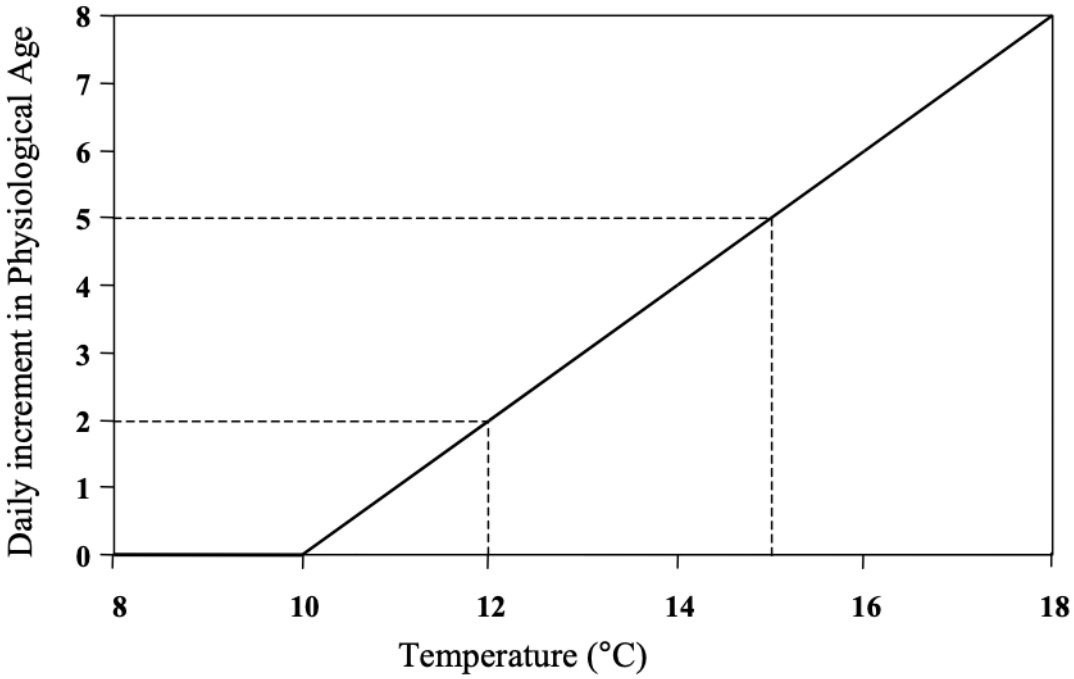
3.1.3 Figure **Accumulation of “Day-degrees” at a constant temperature**

Suppose now that the temperature is different each day, but constant within the day. Three successive days, with temperatures of 15°C, 6°and 12°C are shown in Figure 4.2. Again, the number of degree-days that have accumulated is calculated by summing the area under the temperature curve (and above the threshold temperature line). In this case, it is 5 + 0 + 2 = 7 degree days. The temperature on the second day is below the developmental threshold, and its contribution to the sum is zero. Note that the average temperature for the three days is 11°C, which would yield only 3 degree-days over the 3 days if it were applied directly.



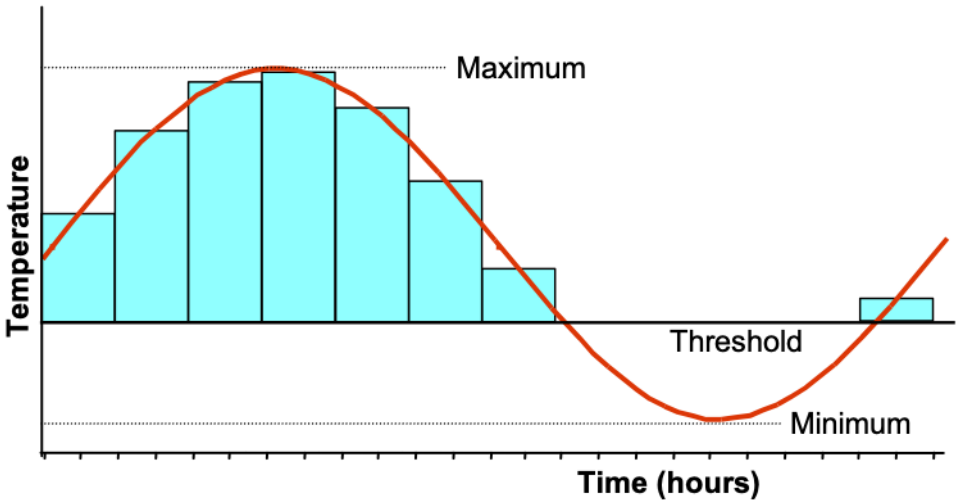
3.1.4 Figure **Accumulation at day-degrees at different temperatures**

An important point to note is that determining accumulated degree-days in this way is exactly equivalent to specifying development in terms of “Physiological Age”, with “Physiological Age” a function of temperature using the “Linear above Threshold” function. This is easily seen by comparing Figure 4.2 with the graph below (Figure 4.3), where the slope of the function is set to 1.



3.1.5 Calculating Degree Days Using the Circadian Cycle

When using actual field temperatures, the situation is more complex, as the temperature varies between the day's minimum and maximum in some cyclical way each day. APSIM uses the daily maximum and minimum temperatures of the meteorological database to fix the "crest" and "trough" limits of a sinusoidal, circadian cycle of temperatures (Figure 4). To calculate the number of degree- days, the same procedure as in the above examples is used. The sine curve is divided into 8 segments (each of 3 hours). A rectangle (of width 3 hours) is created for each of the 8 segments, with the base on the threshold temperature and the midpoint of its top edge on the sine curve (Figure 4). The area of this rectangle is used as the degree-days accumulated for those 3 hours, and all 8 areas are added to obtain the total degree-days for the whole day. The sinusoidal approximation should be adequate for most purposes that APSIM-LifeCycle is used for, and will be used for the Pseudo-aphid model.



3.1.6 Figure 4.4. Circadian cycle between daily maximum and minimum temperatures, and the 3-hourly degree-day summation

The current Pseudo-aphid model already implicitly uses degree-days for determining development (since the "Linear above Threshold" function is used to determine the accumulation of Physiological Age). The model will now be modified so that a circadian curve of temperature is used for driving the development.

3.1.7 Simulation set up

Return the clock and met file to what they were in tutorial 1.2

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2016
	End simulation =	31/12/2016
Weather		
	Select met file =	Normal.met

3.1.8 Add a function to calculate day degrees for development

Many models in APSIM use thermal time to describe different processes. There a number of ways to convert max and minimum temperature to sub daily temperatures. In this example we will use houly temperature to calculate our average temperature In many other APSIM models 3-hourly temperatures are used.

3.1.9 Add an HourlyInterpolation Function

The Thermal time function will be added to the lifecycle node so all life stages can easily use it

Steps

1. Right click on **LifeCycle**
2. Select **add Model > Functions > HourlyInterpolation**
3. Rename HourlyInterpolation to **ThermalTime**
4. Use **control** and up arrow to position it above juvenile
5. set method used to agregate sub daily temperature function to **average**

This function will interpolate the information that comes to it from variables below it and will either sum or average these values for this example set the thermalTime to **Average**

3.1.10 Add an interpolationMethod

Different interpolation methods exist within APSIM the **HourlySinPpAdjusted** method:

calculating the hourly temperature based on Tmax, Tmin and daylength At sunrise (th = 12 - d/2), the air temperature equals Tmin. The maximum temperature is reached when th equals 12 + p h solar time.The default value for p is 1.5 h. The sinusoidal curve is followed until sunset. Then a transition takes place to an exponential decrease, proceeding to the minimum temperature of the next day. To plot this curve correctly, we first need the starting point, the temperature at sunset (Tsset). or the ThreehourSin method commonly used in APSIM classic models and based on the work of Jones, C.A., Kiniry, J.R., 1986. A Simulation Model of Maize Growth and Development. Texas A&M University Press, College Station. https://doi.org/10.1046/j.1469-8137.2003.00717.x/full.

For this model we will use the hourly interpolation method

Steps

1. Right click on **ThermalTime**
2. Select **add Model > Functions > HourlySinPpAdjusted**
3. Rename HourlySinPpAdjusted to **interpolationMethod**

3.1.11 Add a Response variable

The Response variable is required to drive the interpolation function in this case it will be treated as a direct response, but it can also be used to define base and maximum temperaures but for this example our thermal times calculations will include all temperatures

Steps

1. Right click on **ThermalTime**
2. Select **add Model > Functions > XYPairs**
3. Rename XYPairs to **Response**
4. Set XYPairs

AddModel	Function	Name	Value
-	XYPairs	XYPairs	x= 0, 20, 50, 100
-			y= 0, 20, 50, 100

1. Use **control** and **up arrow** to position **Response** above the **InterpolationMethod**

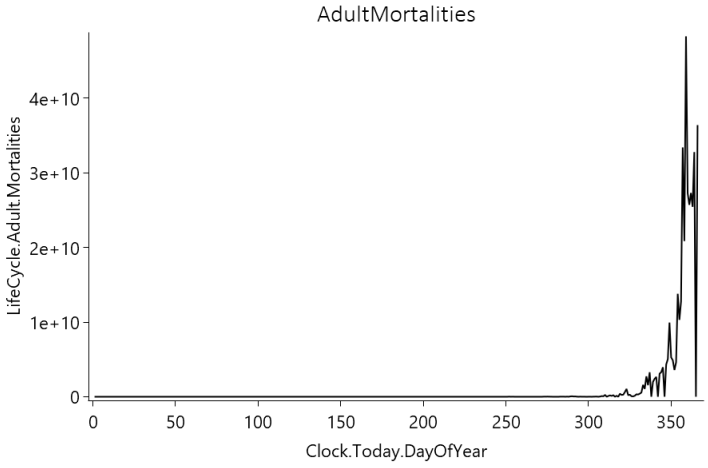
3.1.12 Set Development Rate

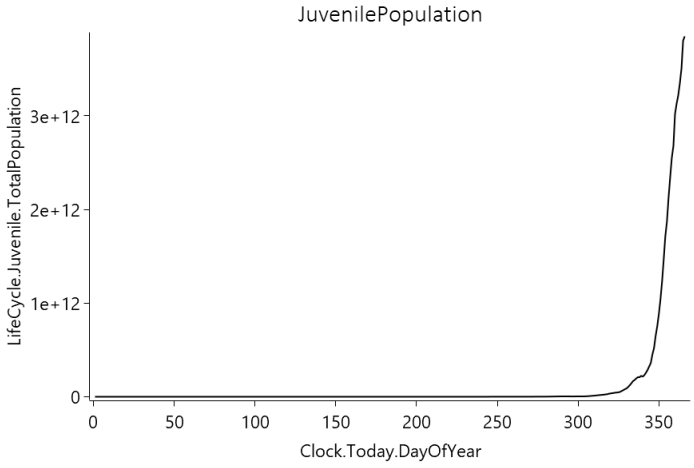
Change the Development rate to use thermal time

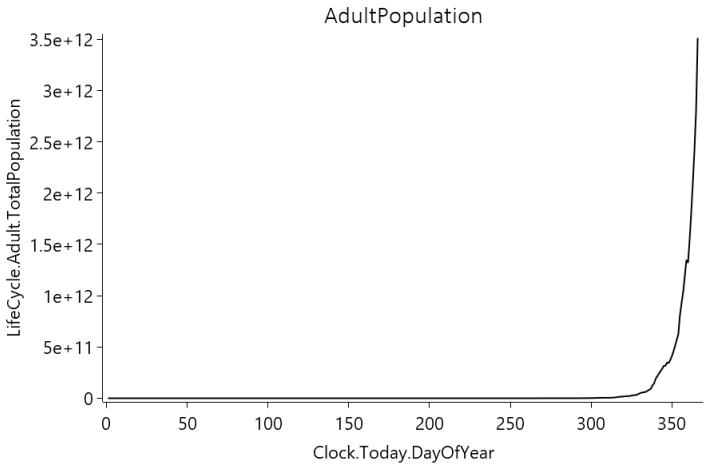
Steps

In the LifeCycle, expand **Juvenile > Development > DevelopmentRate.** Set **DevelopmentRate** function set: Xproperty to [ThermalTime] X value trigger to 10 Gradient to 0.01

3.2 Results







4 Tutorial_PseudoAphid Development Using hourly Development

Tutorial-3-1-PseudoAphid_DegreeDayDevt is the starting point of this tutorial

4.1 Modifying the Model to simulate development in response to hourly temperatures

4.1.1 Simulation set up

Return the clock and met file to what they were in tutorial 1.2

APSIM Object	Description	Values
Clock		

APSIM Object	Description	Values
	Start simulation =	01/01/2016
	End simulation =	31/12/2016
Weather		
	Select met file =	Normal.met

In this Tutorial we will add hourly development to the Juvenile development stage. To start we will remove the existing development process and then convert the thermal time process to control development

4.1.2 Modifying Juvenile Development

Steps

Right click on **DevelopmentRate** it is under **Juvenile > Development** Select **Delete** Right click on **ThermalTime** Copy and paste onto **DevelppmentRate** rename ThermalTime to **ThermalTimeDevelopmentDelete ThermalTime**

If we re-examine the Life tables we see an additional temperature the 34°C tempereature that stops development This is more realistic as most organisims will have an optimum range of temperatures for development. This envelope of temperature is often described as the cardinal temperatures for development. if we include this envelope below the hourly Interpoloation function only temperatures within the envelope will be used to calculate thermal time and development.

Pseudo Aphid Life table with rate of development

Temperature (°C)	No. of days to developto adult	Rate of Developmentper day
10	No Development	0
15	20	0.05
20	12	0.08
25	8	0.125
30	5	0.2
34	No Development	0

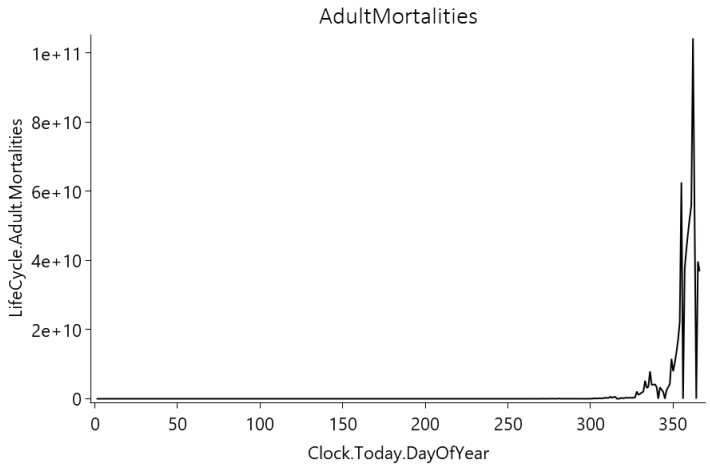
Based on our knowledge of the pseudo aphid we can now construct the cardinal temperatures for development within the XYPairs of the **Response**

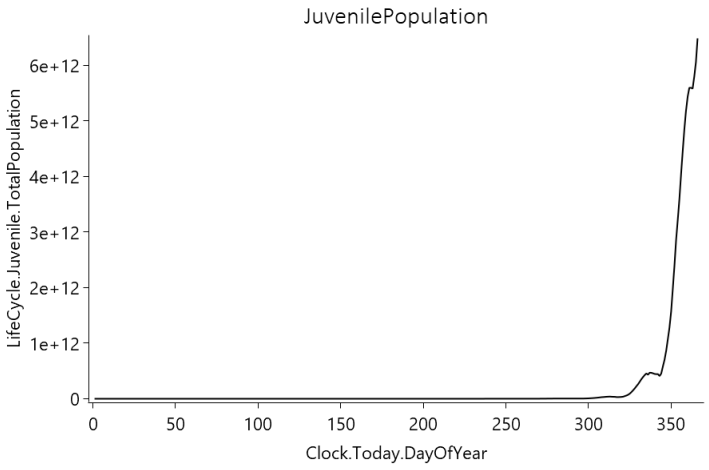
Response values

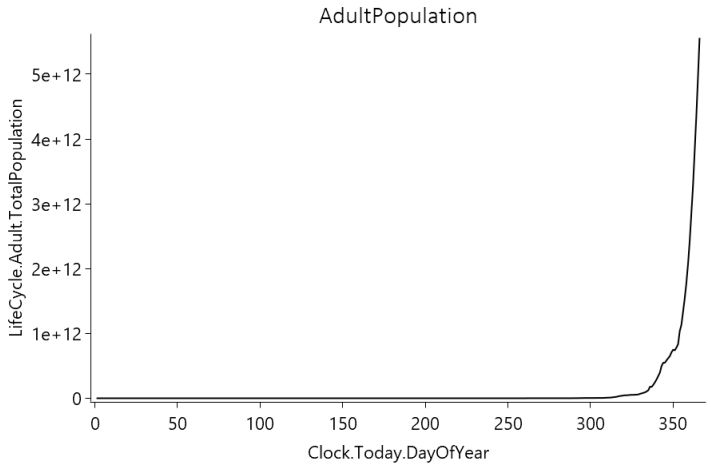
X	Y
0	0
10	0
20	0.1
30	0.2
34	0

Run APSIM

4.2 Results







5 Tutorial_PseudoAphid_Temperature Dependent Mortality

Tutorial-4-1-PseudoAphid_3hrlyDevt is the starting point of this tutorial

5.1 Introduction

In the current pseudo-aphid model, the only source of mortality is in the Adult stage, where adults die of “old age” (i.e., after they reach a specified Chronological Age). Mortality can be built into each lifestage and mortality can be linked to any suitable user-selected variable(s). In the field, aphids suffer mortality for a variety of reasons, predators, humidity, disease, temperature, etc. and all of these can be modelled. In this tutorial, temperature-driven mortality will be added to the pseudo-aphid model. Aphids are affected adversely by both excessively high and low temperatures. It will be assumed that each lifestage of the pseudo-aphid suffers a linear mortality in response to unfavourably high/low temperatures and that this will be the only additional cause of mortality apart from the currently existing adult death due to age.

5.2 Temperature-based Mortality

Assume that both the juveniles and adults have the same sensitivity to extremes of cold and heat. Further assume that we have data that indicates that mortality starts to occur when temperatures drop below 4°C and increases by 0.1 for each degree temperature drop below that value (i.e., at 3°C, there is 10% mortality per timestep, at 2°C, there is 20%, etc). The data also shows that mortality occurs above 34°C, and increase by 0.05 for every degree above that temperature (so that, for example, 20% of individuals die each timestep at 38°C due to heat stress). These responses will now be incorporated into the model.

Low temperature Mortality

X	Y
-6	1
-4	0.8

X	Y
-2	0.6
0	0.4
1	0.3
2	0.2
3	0.1
4	0

High temperature Mortality

X	Y
34	0
38	0.2
42	0.4
46	0.6
50	0.8
54	1

5.3 Juvenile Temperature Based Mortality

Steps

1. **Delete** Juvenile **Mortality** constant
2. Add an **AddFunction** to **juvenile** rename **Mortlity**
3. Add a **MultiplyFunction** to **Mortality** rename **LowTempMortlaity**
4. Add a **VariableReference** to **LowTempMortlaity** rename **CohortPopulation**
5. Add a **LinearInterpolationFunction** to **LowTempMortlaity** rename **LowTempMortlaity**
6. Add a **XYPairs** to LinearInterpolationFunction **LowTempMortlaity**
7. Add a **VariableReference** to LinearInterpolationFunction **LowTempMortlaity** rename **XVariable**
8. Set **XVariable** to [weather].MinT
9. set **XYPairs** to Low temp Mortality table
10. set CohortPopulation to [Juvenile].CurrentCohort.Population

Repeat for High temp mortality

1. Add a **MultiplyFunction** to **Mortlity** rename **HighTempMortlaity**
2. Add a **VariableReference** to **HighTempMortlaity** rename **CohortPopulation**
3. Add a **LinearInterpolationFunction** to **HighTempMortlaity** rename **HighTempMortlaity**
4. Add a **XYPairs** to LinearInterpolationFunction **HighTempMortlaity**
5. Add a **VariableReference** to LinearInterpolationFunction **HighTempMortlaity** rename **XVariable**
6. Set **XVariable** to [weather].MaxT
7. set **XYPairs** to High temp Mortality table
8. set CohortPopulation to [Juvenile].CurrentCohort.Population

5.4 Adult Temperature Based Mortality

The pseudo Aphid sensetivity to temperature is the same in the adult stage as the Juvenile stage. The adult stage already has agebased mortality that kills the population after 20 days as an adult to include temperature based mortality add the lowTempMortality (multiplyFunction) to the adult Mortlity (addFunction)

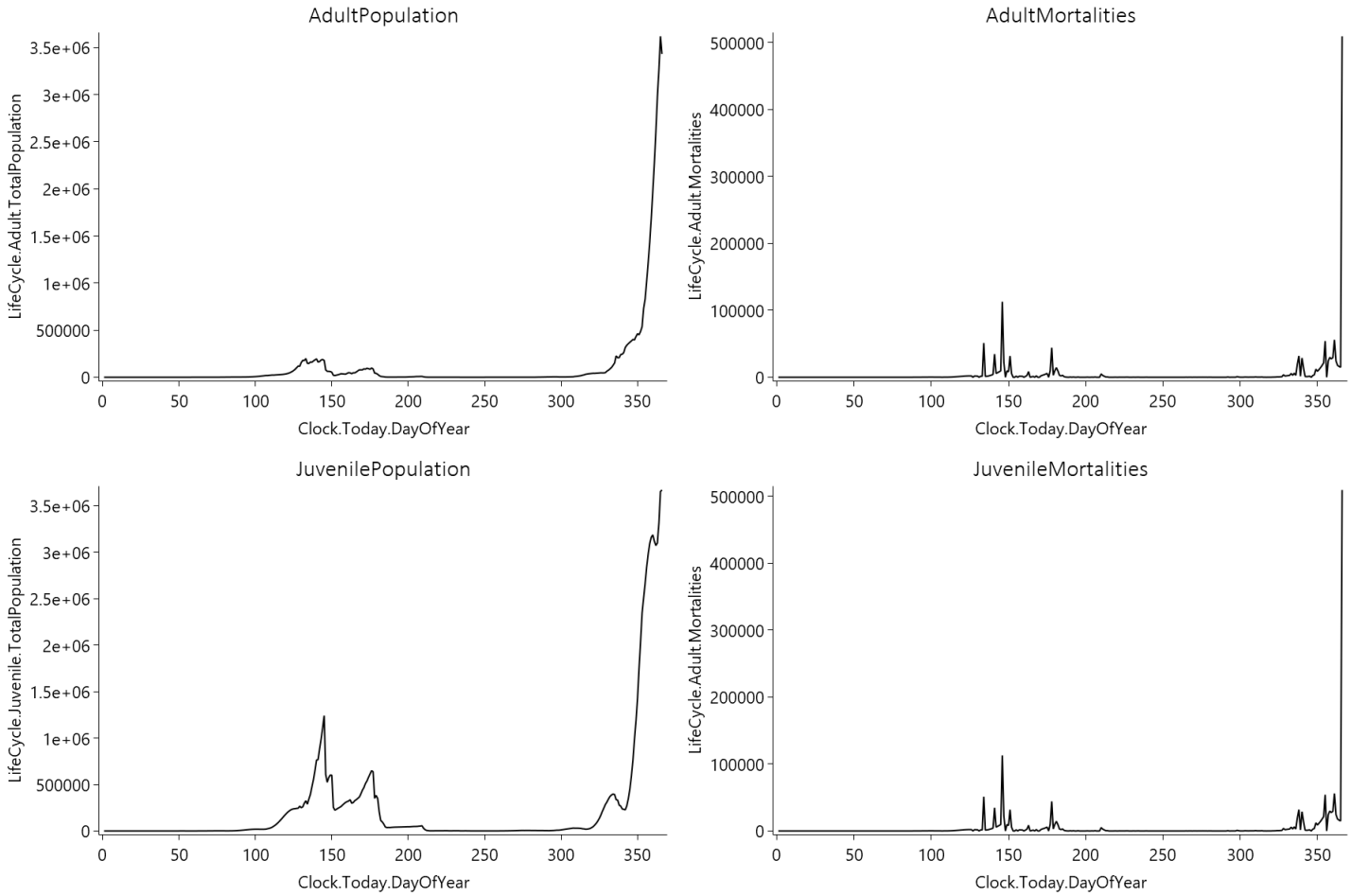
Steps

1. Right click on **LowTempMortlity** in the **Juvenile** LifePhase **copy**
2. Right Click on **Mortality** (AddFunction) **Adult** LifePhase and **paste**
3. Repeat for High temp.

5.5 Results

add a new result output Juvenile mortality add a new graph to the results folder see tutorial 1.2 for help

5.6 Results



6 Tutorial_PseudoAphid Mortality (density-dependent)

Tutorial-5-1-PseudoAphidMortality (TempDepMort) is the starting point of this tutorial

6.1 Introduction

Two variables have so far been used to control the pseudo-aphid's mortality: Chronological Age in the adults and temperature in both stages. In Tutorial 5, the effect of these mortality factors was explored. It was evident that population stability was not achieved – populations either increasing forever in suitable locations or becoming extinct in unfavourable locations. Much of this is due to the fact that an isolated population is being modelled and in fact populations will often die out in the field, to be re-established by immigrants later. However, many factors that cause mortality in populations are linked in some way to the density of the population, generally increasing the proportional mortality as the population density increases. These mortality factors (termed density-dependent) do help stabilize isolated populations.

6.2 Population Density and Mortality

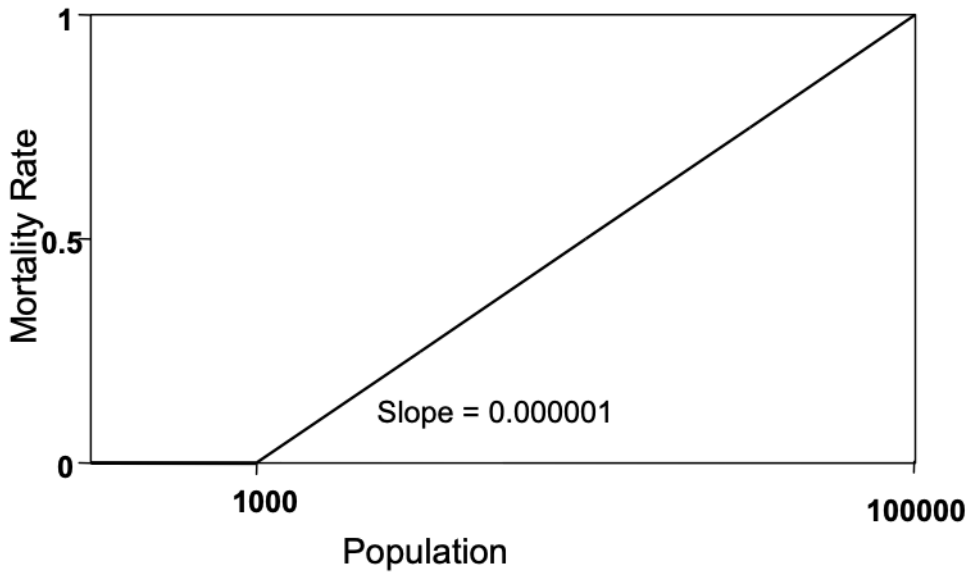
Population density is measured as the number of individuals per unit area. If for a simple case, the living area available for the population was fixed (eg. the number of pseudo-aphids present in a lucerne crop), population density would simply be a reflection of total population. This tutorial will assume such a situation so that an increase in population will automatically mean an increase in population density. Exactly how an increase in population density affects the population growth rate depends on the individual species' characteristics, but there are generally a few main causes of increased mortality rates in a larger population: crowding, food depletion, predation by attracted predators and increase of disease through rapid transmission of pathogens. Density dependent mortality is a stabilising influence on a population because it has a negative feedback effect. Put simply, the more the population grows, the greater will the rate of mortality become and the ensuing deaths will therefore reduce the population growth rate. As the population decreases, the mortality rate decreases also and the population growth rate increases once more. The population therefore tends to be cyclical with numbers sometimes exceeding the ability of the environment to sustain it but never quite falling away to extinction.

This is an area of modelling where data is often absent, or if available, difficult to interpret. It is usually evident that a population is fluctuating between some limits, but the causes are usually due to many factors and difficult to establish experimentally. When modelling the population, it may be necessary to lump many into one single mortality function, and "guess" at the function shape and parameter values, so that the "model" population

predicts the known field population. In that way, the model forms a hypothesis, which will stand or fall when tested with further data. As with all hypotheses, the model will lead to a more directed data collection effort and thus a better utilization of available resources. The improvement in understanding may then allow the previous, composite function to be split into component parts that represent the actual processes occurring better.

6.3 Defining the Mortality Rate

When a population is small, it can be assumed that mortality due to population density effects will be very low. For this tutorial, it will be proposed that no mortality due to population density effects occurs until a minimum of 1000 individuals is present (this might be the organism's population on a rose bush). The proposition further suggests that as the population increases, density dependent mortality rate increases in direct proportion until once the population reaches 100,000 individuals, the mortality rate becomes 1. This mathematical structure suggests a "Linear above Threshold" function. If the data is plotted (Figure 6-1) the slope can be calculated and is approximately equal to 0.000001. For this tutorial, it is assumed that the population on the x-axis is the population of the stage whose mortality rate is being calculated (i.e., adults affect adult density dependent mortality and juveniles affect juvenile mortality). This is obviously not realistic, and an exercise suggested at the end of the tutorial will propose a better model.



6.3.1 Juvenile Density dependent Morality

Steps

1. Add a **MultiplyFunction** to **Mortality** rename **DensityDependentMortality**
2. Add a **VariableReference** to **DensityDependentMortality** rename **CohortPopulation**
3. set to [Juvenile].CurrentCohort.Population
4. Add a **LinearAfterThreshodFunction** to **DensityDependentMortality** rename **DensityDependentMortality**
5. Set **XProperty** to [LifeCycle].Juvenile.TotalPopulation
6. set **XValueTrigger** to **1000**
7. set **Gradient** to **1000**

6.3.2 Adult Density dependent Morality

Steps

1. Add a **MultiplyFunction** to **Mortality** rename **DensityDependentMortality**
2. Add a **VariableReference** to **DensityDependentMortality** rename **CohortPopulation**
3. set to [Adult].CurrentCohort.Population
4. Add a **LinearAfterThreshodFunction** to **DensityDependentMortality** rename **DensityDependentMortality**
5. Set **XProperty** to [LifeCycle].Adult.TotalPopulation
6. set **XValueTrigger** to **1000**
7. set **Gradient** to **1000**

6.4 Simulation set up

Return the clock and met file to what they were in tutorial 1.2

Simulation Timing

APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2000
	End simulation =	31/12/2016
Weather		
	Select met file =	Toowoomba.met

6.5 Report

Add a Total Population Report and Figure

Run APSIM

6.6 Conclusion

Questions

How long does it take for the population to die out?

Change the Met file to Gatton.met

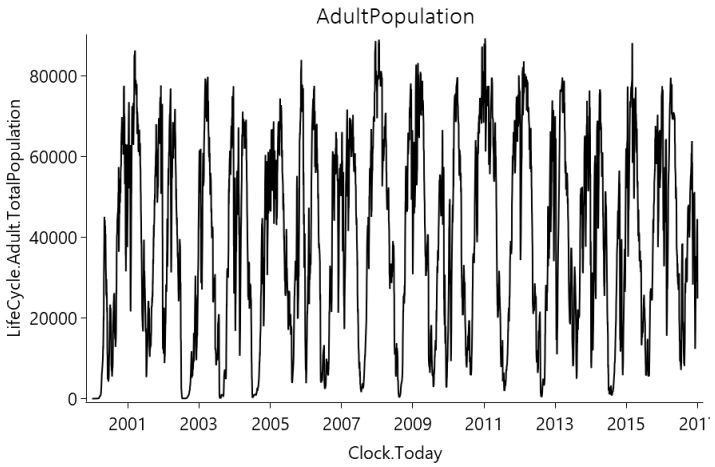
Does the Populations approach equilibrium in the long term

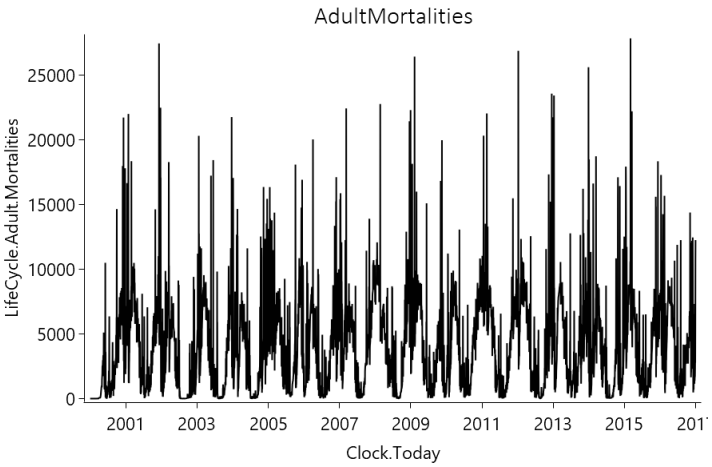
What is the main difference between the climates of Gatton and Toowoomba ?

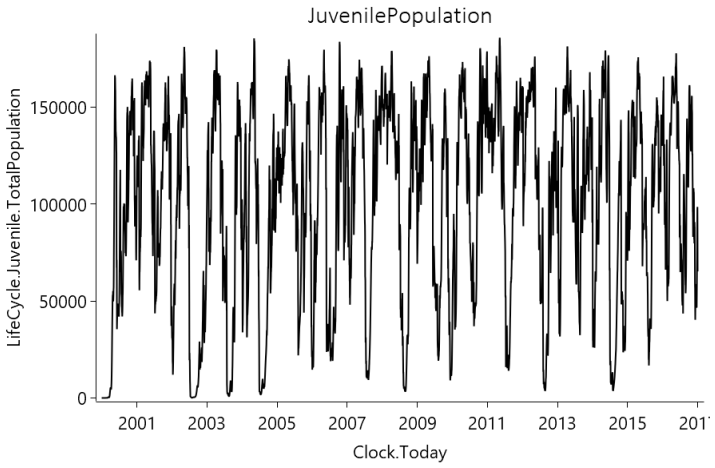
In tutorial 6 we have simulated the PseudoAphid population for a period of 17 years instead of the three-monthly or annual periods used in earlier tutorials.

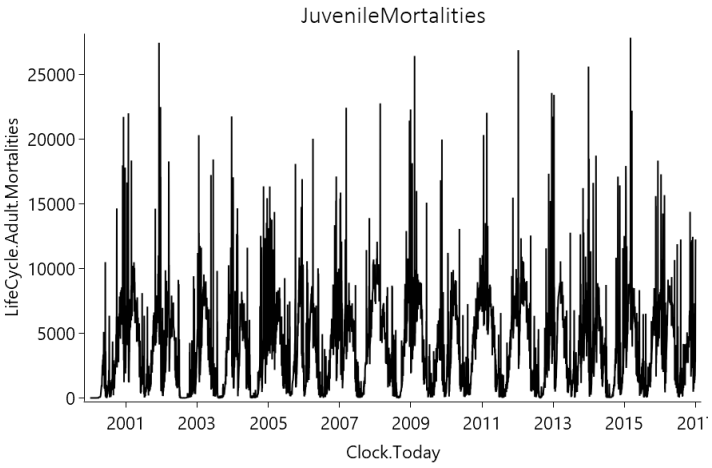
Note that while the adult and juvenile populations fluctuate seasonally, they do not increase exponentially and remain within boundaries.

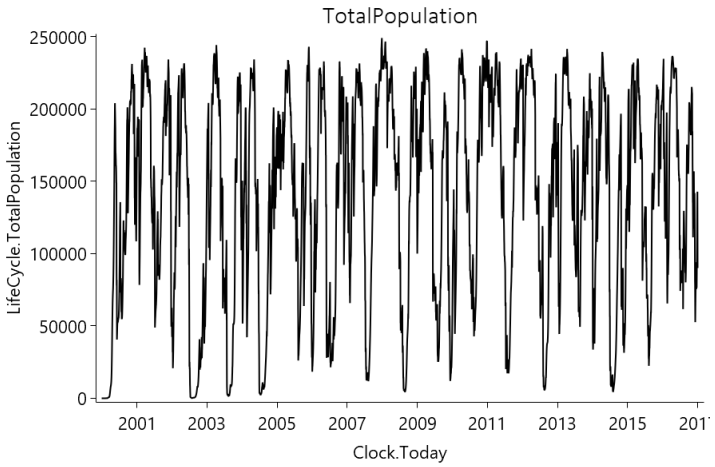
6.7 Results











7 Tutorial_PseudoAphid Mortality (dryness-dependent based on rainfall)

Tutorial-6-1-PseudoAphidMortality (DensityDepMort) is the starting point of this tutorial

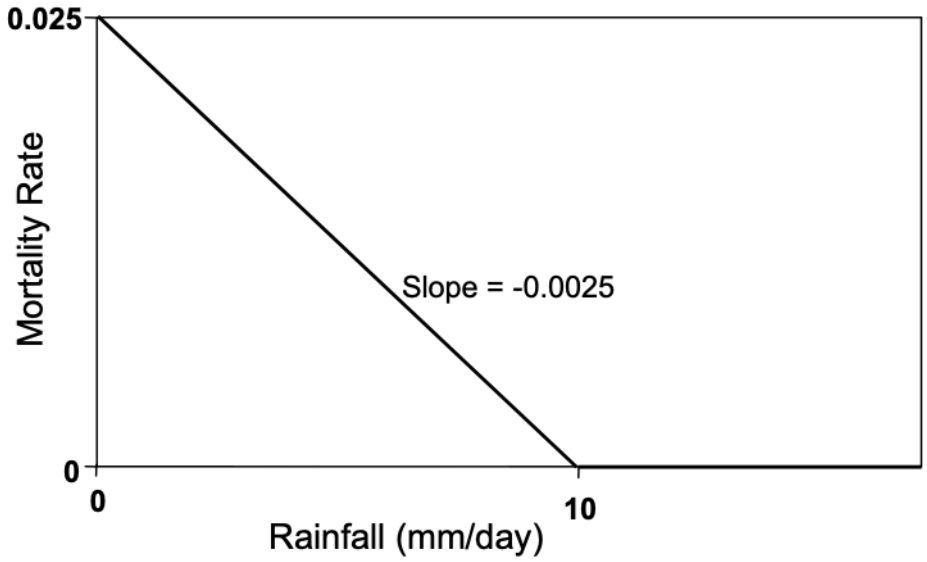
7.1 Introduction

All organisms are sensitive to the absence of water, and the pseudo-aphid is no exception. Survival can be adversely affected if it is too dry, and in this tutorial a first attempt is made to model this sensitivity to dryness using rainfall. In the next tutorial, a better way of modelling the effects of inadequate rainfall will be demonstrated.

Modelling Dryness Dependent Mortality

As rainfall decreases, the soil dries out to produce plant stress and this in turn affects pseudo-aphid mortality. In this tutorial, this will be modelled very simply, by assuming that there is a particular level of rainfall each day that is required for no mortality due to dryness to occur. If insufficient rain falls during any one day, mortality on that day is linearly dependent on how far the rainfall is below that threshold. The relationship is illustrated in Figure

7-1. A Threshold of 10 mm\day is chosen – rainfall above that level produces no mortality. The slope of the function is selected so that 25 consecutive days without rainfall would produce an accumulated mortality of about 0.5.



7.1.1 Figure Mortality rate as a function of Rainfall

Hence,

(1 - m)25 = 0.5, where m = mortality/day

i.e., 25 x ln(1- m) = ln(0.5)

i.e., m = (1-e)(ln(0.5/25) = 0.027 (approximately)

Using 0.025, which is close enough for our purpose, fixes the slope at –0.0025 (the "-" sign is required as the slope is negative, with decreasing rain increasing the mortality rate).

7.1.2 Juvenile dryness dependent Morality

Steps

Add a **MultiplyFunction** to **Mortality** rename **DrynessMortality** Add a **VariableReference** to **DrynessMortality** rename **CohortPopulation** set to [Juvenile].CurrentCohort.Population Add a **LinearInterpolationFunction** to DrynessMortality rename **DrynessMortality** Add XYpairs to the LinearInterpolationFunction **DrynessMortality** set x = (0, 10) y = (0.025, 0) Add a **VariableReference** to the LinearInterpolationFunction **DrynessMortality** rename **XValue** Set **XProperty** to [weather].Rain

7.1.3 Adult dryness dependent Morality

Steps

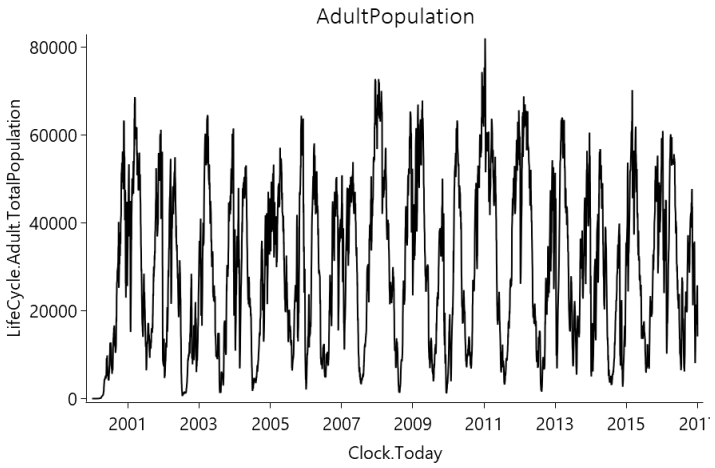
Add a **MultiplyFunction** to **Mortality** rename **DrynessMortality** Add a **VariableReference** to **DrynessMortality** rename **CohortPopulation** set to [Adult].CurrentCohort.Population Add a **LinearInterpolationFunction** to DrynessMortality rename **DrynessMortality** Add XYpairs to the LinearInterpolationFunction **DrynessMortality** set x = (0, 10) y = (0.025, 0) Add a **VariableReference** to the LinearInterpolationFunction **DrynessMortality** rename **XValue** Set **XProperty** to [weather].Rain

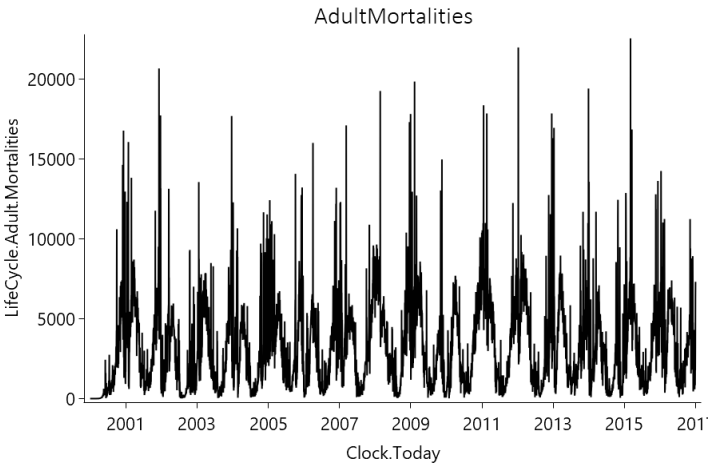
Simulation Timing

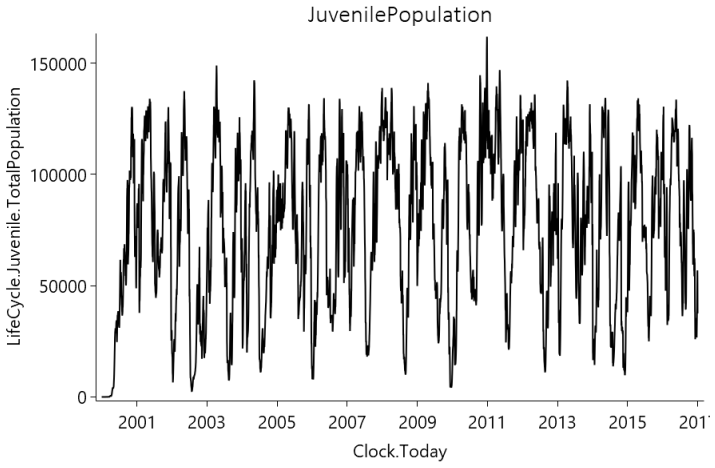
APSIM Object	Description	Values
Clock		
	Start simulation =	01/01/2000
	End simulation =	31/12/2016

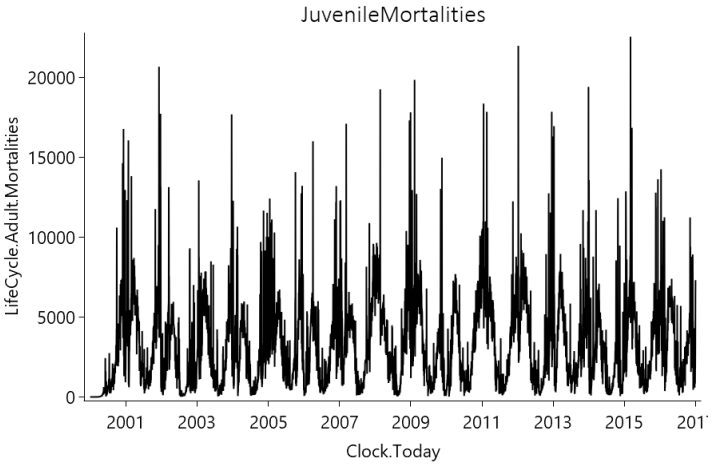
APSIM Object	Description	Values
Weather		
	Select met file =	Gatton.met

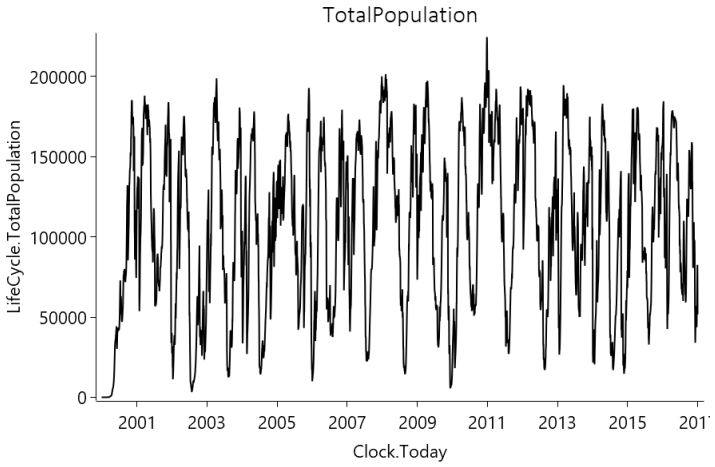
7.2 Results











8 Tutorial_PseudoAphid Mortality (dryness-dependent based on soil moisture)

Tutorial-7-1-PseudoAphidMortality (dryness-dependent based on rainfall) is the starting point of this tutorial

8.1 Introduction

Tutorial 7 implied that rainfall by itself may not be the best variable to use in modelling dryness dependent mortality. This is due to the fact that rainfall is rather erratic in nature, but its effects on insects are generally not direct. The soil and plant environment tends to integrate the effects of rainfall over time. In addition, there are the effects of atmospheric moisture (measured as relative humidity or saturation deficit) which may be more appropriate drivers of insect mortality.

Soil moisture is usually the dominant factor determining microclimatic conditions and the moisture content of vegetation. For an insect that lives in close contact with the plant (such as the Pseudo-aphid), soil moisture alone can provide a very useful indicator of the moisture conditions the insect is experiencing. ApsimX contains a soil module that is designed to simulate the changes in soil moisture with given rainfall and evaporation. The module can be described in detail (right mouse click on the Soil module and select "Include in Documentation". When documentation is generated (Simulations>Create documentation), then a detailed description of soil processes will be included in the documentation. This tutorial will illustrate the use of this module within the Pseudo-aphid model.

Soil Moisture in the Soil Module

The Soil module provides a number of output variables that reflects the simulated current state of the soil's moisture content in response to rainfall, evaporation, runoff, drainage and plant uptake (where plants are included in the simulation). The soil moisture will be scaled between 0 and 2, with a value of 0 indicating a dry soil at lower limit, 1 denoting a soil at drained upper limit, and 2 denoting a saturated soil.

The "Lifecycle" module previously used rainfall as the input variable that controlled dryness-dependent mortality (in Tutorial 7). This will now be changed so that soil moisture levels control this mortality. The parameters will be chosen so that dryness-dependent mortality commences to have an effect once the soil moisture level drops below drained upper limit.

8.1.1 Juvenile Soil Dryness dependent Morality

Steps

1. **modify** Juvenile > DrynessMortality > interpolationfunction **DrynessMortalityxValue** to [SoilWaterScale]
2. **modify** Juvenile > DrynessMortality > interpolationfunction **DrynessMortalityXYPairs** to

XY Pairs

X	Y	
0	1	
1	0	
2	0	

8.1.2 Adult Soil Dryness dependent Morality

Steps

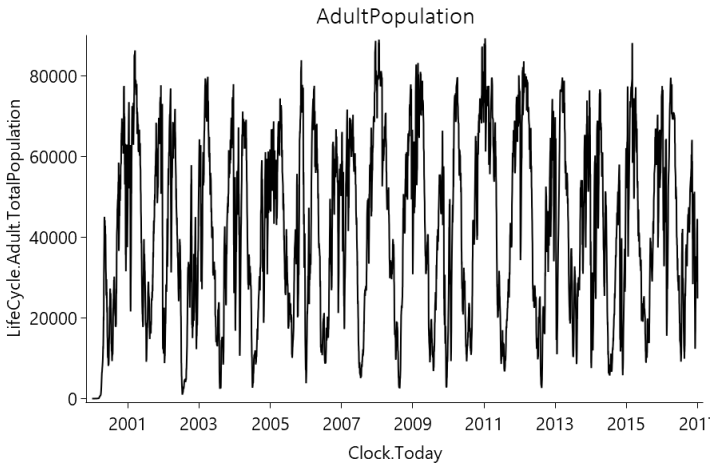
1. **modify** Adult > DrynessMortality > interpolationfunction **DrynessMortalityxValue** to [SoilWaterScale]
2. **modify** Adult > DrynessMortality > interpolationfunction **DrynessMortalityXYPairs** to

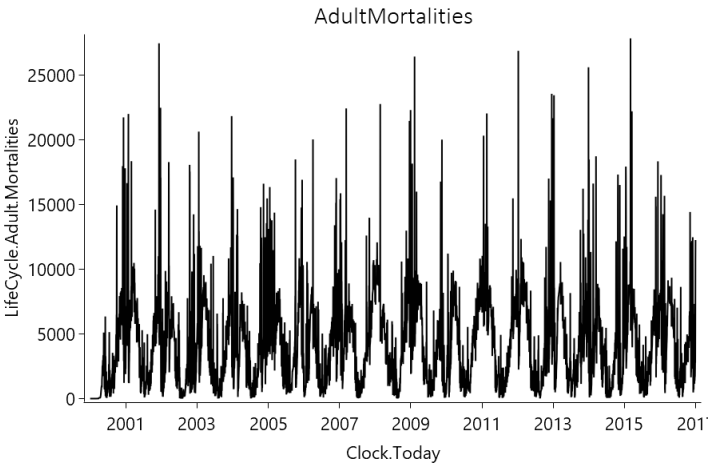
XY Pairs

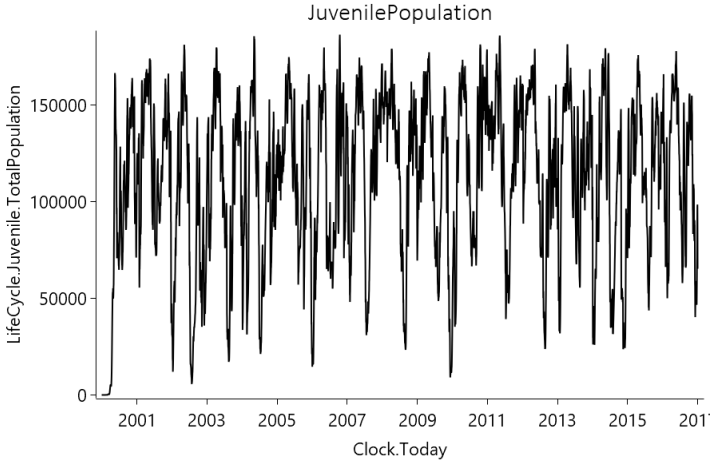
X	Y	
0	1	
1	0	
2	0	

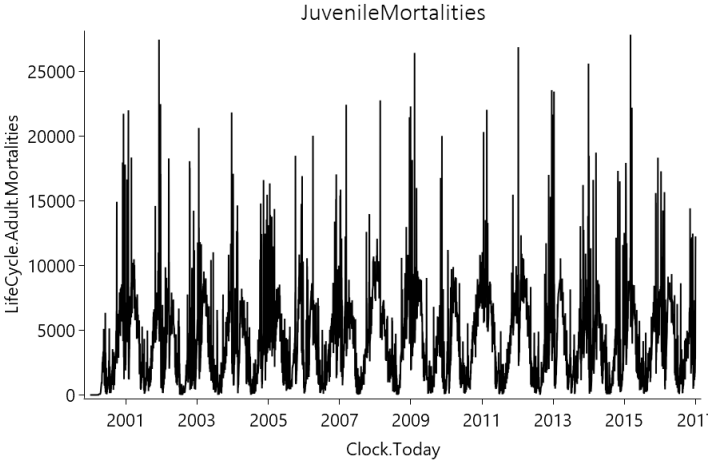
The fertiliser model

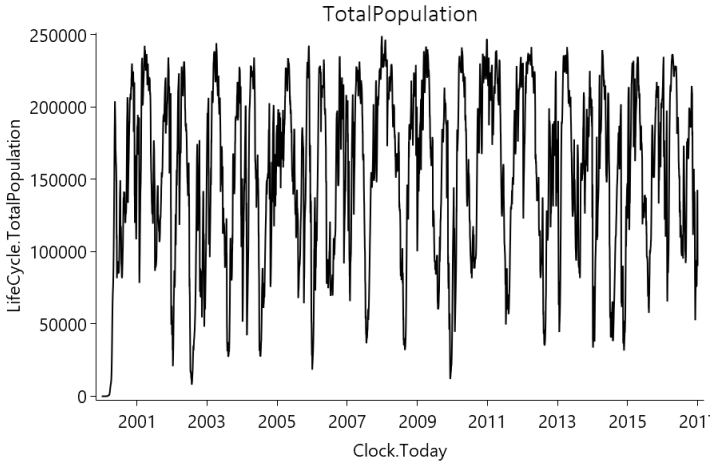
8.2 Results


AdultPopulation
LifeCycle.Adult.TotalPopulation
0
20000
40000
60000
80000
2001
2003
2005
2007
2009
2011
2013
2015
Clock.Today


AdultMortalities
LifeCycle.Adult.Mortalities
0
5000
10000
15000
20000
25000
2001
2003
2005
2007
2009
2011
2013
2015
Clock.Today


JuvenilePopulation
LifeCycle.Juvenile.TotalPopulation
0
50000
100000
150000
2001
2003
2005
2007
2009
2011
2013
2015
Clock.Today


JuvenileMortalities
LifeCycle.Adult.Mortalities
0
5000
10000
15000
20000
25000
2001
2003
2005
2007
2009
2011
2013
2015
Clock.Today


TotalPopulation
LifeCycle.TotalPopulation
0
50000
100000
150000
200000
250000
2001
2003
2005
2007
2009
2011
2013
2015
Clock.Today