



## 1 The APSIM Peanut Model

The model consists of:

- \* a phenology model to simulate development between growth phases
- \* a structure model to simulate plant morphology
- \* a collection of organs to simulate the various plant parts
- \* an arbitrator to allocate resources (N, biomass) to the various plant organs

The model has been developed using the Plant Modelling Framework (PMF) of [Brown et al., 2014](#). This new framework provides a library of plant organ and process submodels that can be coupled, at runtime, to construct a model in much the same way that models can be coupled to construct a simulation. This means that dynamic composition of lower level process and organ classes (e.g. photosynthesis, leaf) into larger constructions (e.g. maize, wheat, sorghum) can be achieved by the model developer without additional coding.

The model is constructed from the following list of software components. Details of the implementation and model parameterisation are provided in the following sections.

### List of Plant Model Components.

Component Name	Component Type
Arbitrator	Models.PMF.OrganArbitrator
Phenology	Models.PMF.Phen.Phenology
Leaf	Models.PMF.Organs.SimpleLeaf
Grain	Models.PMF.Organs.ReproductiveOrgan
Root	Models.PMF.Organs.Root
Nodule	Models.PMF.Organs.Nodule
Shell	Models.PMF.Organs.GenericOrgan
Stem	Models.PMF.Organs.GenericOrgan
AflatoxinRisk	Models.Functions.BoundFunction
MortalityRate	Models.Functions.Constant

### 1.1 Arbitrator

#### 1.1.1 Arbitrator

The Arbitrator class determines the allocation of dry matter (DM) and Nitrogen between each of the organs in the crop model. Each organ can have up to three different pools of biomass:

- \* **Structural biomass** which is essential for growth and remains within the organ once it is allocated there.
- \* **Metabolic biomass** which generally remains within an organ but is able to be re-allocated when the organ senesces and may be retranslocated when demand is high relative to supply.
- \* **Storage biomass** which is partitioned to organs when supply is high relative to demand and is available for retranslocation to other organs whenever supply from uptake, fixation, or re-allocation is lower than demand.

The process followed for biomass arbitration is shown in the figure below. Arbitration calculations are triggered by a series of events (shown below) that are raised every day. For these calculations, at each step the Arbitrator exchange information with each organ, so the basic computations of demand and supply are done at the organ level, using their specific parameters.

1. **doPotentialPlantGrowth.** When this event occurs, each organ class executes code to determine their potential growth, biomass supplies and demands. In addition to demands for structural, non-structural and metabolic biomass (DM and N) each organ may have the following biomass supplies:

\* **Fixation supply.** From photosynthesis (DM) or symbiotic fixation (N)

\* **Uptake supply.** Typically uptake of N from the soil by the roots but could also be uptake by other organs (eg foliage application of N).

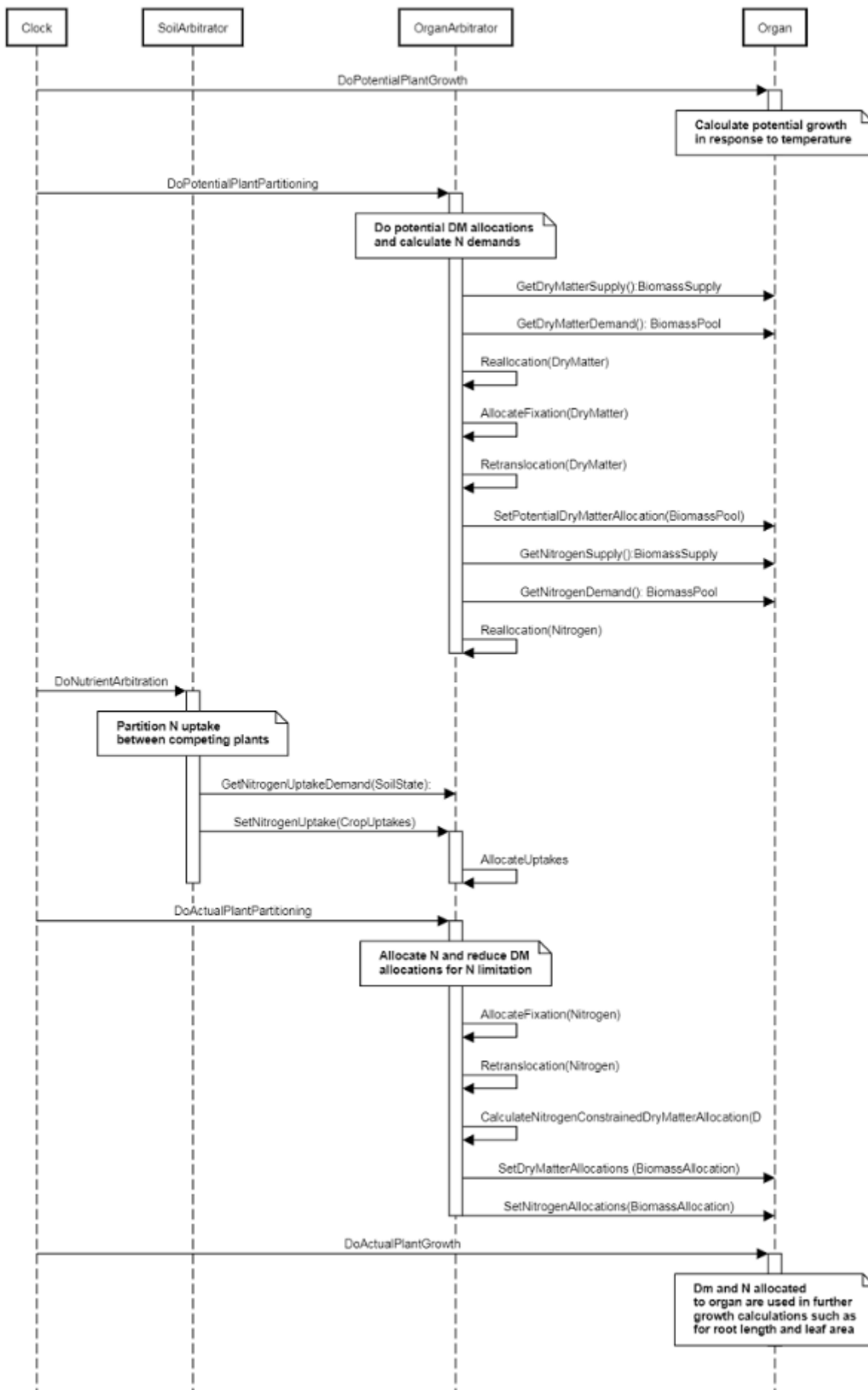
\* **Retranslocation supply.** Storage biomass that may be moved from organs to meet demands of other organs.

\* **Reallocation supply.** Biomass that can be moved from senescing organs to meet the demands of other organs.

1. **doPotentialPlantPartitioning.** On this event the Arbitrator first executes the DoDMSetup() method to gather the DM supplies and demands from each organ, these values are computed at the organ level. It then executes the DoPotentialDMAAllocation() method which works out how much biomass each organ would be allocated assuming N supply is not limiting and sends these allocations to the organs. Each organ then uses their potential DM allocation to determine their N demand (how much N is needed to produce that much DM) and the arbitrator calls DoNSetup() to gather the N supplies and demands from each organ and begin N arbitration. Firstly DoNReallocation() is called to redistribute N that the plant has available from senescing organs. After this step any unmet N demand is considered as plant demand for N uptake from the soil (N Uptake Demand).

2. **doNutrientArbitration.** When this event occurs, the soil arbitrator gets the N uptake demands from each plant (where multiple plants are growing in competition) and their potential uptake from the soil and determines how much of their demand that the soil is able to provide. This value is then passed back to each plant instance as their Nuptake and doNUptakeAllocation() is called to distribute this N between organs.

3. **doActualPlantPartitioning.** On this event the arbitrator call DoNRetranslocation() and DoNFixation() to satisfy any unmet N demands from these sources. Finally, DoActualDMAAllocation is called where DM allocations to each organ are reduced if the N allocation is insufficient to achieve the organs minimum N concentration and final allocations are sent to organs.



**Figure 1:** Schematic showing the procedure for arbitration of biomass partitioning. Pink boxes represent events that occur every day and their numbering shows the order of calculations. Blue boxes represent the methods that are called when these events occur. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.

## 1.2 Phenology

The phenological development is simulated as the progression through a series of developmental phases, each bound by distinct growth stage.

In the new model we simplified phenology by taking out stages that are not measurable (e.g. end of juvenile stage) and by adding new stages that are measurable (e.g. start pod). The new phenology follows the V/R staging system

### 1.2.1 ThermalTime

ThermalTime is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

PreEmergence has a value between Sowing and Emergence calculated as:

$$TT = [Phenology].PreemergentThermalTime$$

VegetativeGrowth has a value between Emergence and StartFlowering calculated as:

$$TT = [Phenology].VegetativeThermalTime$$

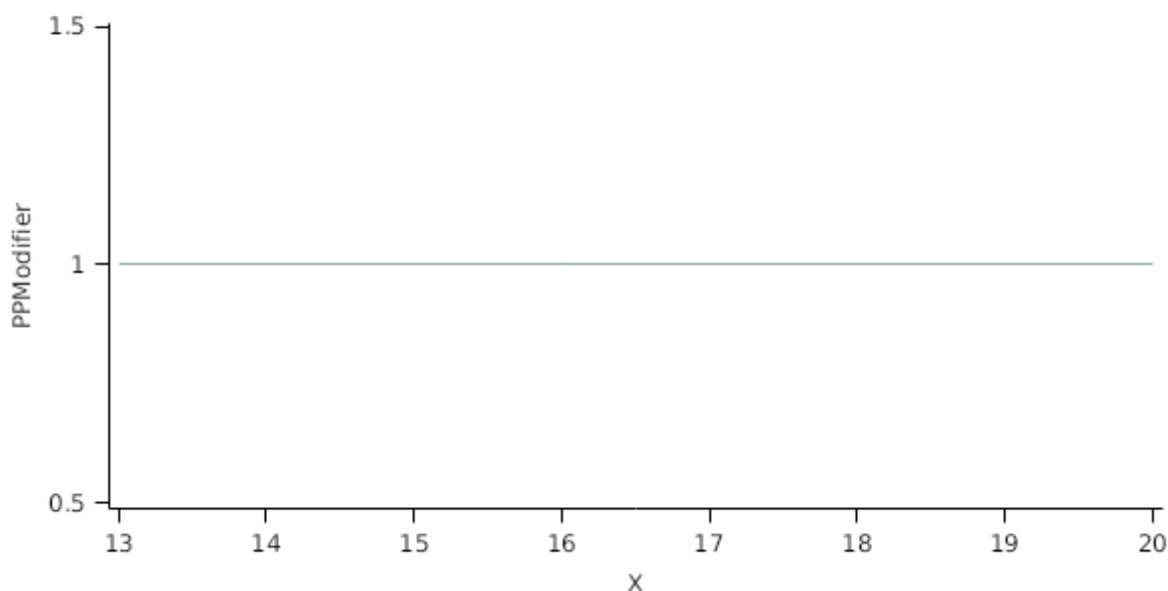
ReproductiveGrowth has a value between StartFlowering and HarvestRipe calculated as:

$$TT = [Phenology].ReproductiveThermalTime \times PPM Modifier$$

*PPModifier* is calculated using linear interpolation

X	PPModifier
13.0	1.0
16.0	1.0
20.0	1.0

### PPModifier



### List of stages and phases used in the simulation of crop phenological development

Phase Number	Phase Name	Initial Stage	Final Stage
1	Germinating	Sowing	Germination
2	Emerging	Germination	Emergence
3	Vegetative	Emergence	StartFlowering
4	EarlyFlowering	StartFlowering	StartPodDevelopment

Phase Number	Phase Name	Initial Stage	Final Stage
5	EarlyPodDevelopment	StartPodDevelopment	StartGrainFilling
6	EarlyGrainFilling	StartGrainFilling	EndCanopyDevelopment
7	MidGrainFilling	EndCanopyDevelopment	EndPodDevelopment
8	LateGrainFilling	EndPodDevelopment	EndGrainFill
9	Maturing	EndGrainFill	Maturity
10	Ripening	Maturity	HarvestRipe
11	ReadyForHarvesting	HarvestRipe	Unused

### 1.2.2 Germinating

The phase goes from sowing to germination and assumes germination will be reached on the day after sowing or the first day thereafter when the extractable soil water at sowing depth is greater than zero.

### 1.2.3 Emerging

This phase goes from germination to emergence and simulates time to emergence as a function of sowing depth. The *ThermalTime Target* for ending this phase is given by:

$$\text{Target} = \text{SowingDepth} \times \text{ShootRate} + \text{ShootLag}$$

Where:

$$\text{ShootRate} = 0 \text{ (deg day/mm),}$$

$$\text{ShootLag} = 70 \text{ (deg day),}$$

*SowingDepth* (mm) is sent from the manager with the sowing event.

Progress toward emergence is driven by thermal time accumulation, where thermal time is calculated as:

$$\text{ThermalTime} = [\text{Phenology}].\text{PreemergentThermalTime}$$

### 1.2.4 Vegetative

This phase goes from emergence to startflowering.

The *Target* for completion is calculated as:

$$\text{Target} = 390 \text{ (oD)}$$

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

### 1.2.5 EarlyFlowering

This phase goes from startflowering to startpoddevelopment.

The *Target* for completion is calculated as:

$$\text{Target} = 200 \text{ (oD)}$$

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

$$\text{Progression} = [\text{Phenology}].\text{ThermalTime}$$

### 1.2.6 EarlyPodDevelopment

This phase goes from startpoddevelopment to startgrainfilling.

The *Target* for completion is calculated as:

$$\text{Target} = 140 \text{ (oD)}$$

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

### **1.2.7 EarlyGrainFilling**

This phase goes from startgrainfilling to endcanopydevelopment.

The *Target* for completion is calculated as:

*Target* = FractionofGrainfilling x [Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue

FractionofGrainfilling = 0.05

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

### **1.2.8 MidGrainFilling**

This phase goes from endcanopydevelopment to endpoddevelopment.

The *Target* for completion is calculated as:

*Target* = FractionofMidToLateGrainfilling x MidToLateGrainfilling

FractionofMidToLateGrainfilling = 0.5

*MidToLateGrainfilling* = [Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue - [Phenology].EarlyGrainFilling.Target

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

### **1.2.9 LateGrainFilling**

This phase goes from endpoddevelopment to endgrainfill.

The *Target* for completion is calculated as:

*Target* = EntireGrainfillPeriod - [Phenology].EarlyGrainFilling.Target - [Phenology].MidGrainFilling.Target

EntireGrainfillPeriod = 1020

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

### **1.2.10 Maturing**

This phase goes from endgrainfill to maturity.

The *Target* for completion is calculated as:

Target = 45 (oD)

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

### **1.2.11 Ripening**

This phase goes from maturity to harvestripe.

The *Target* for completion is calculated as:

Target = 45 (oD)

*Progression* through phase is calculated daily and accumulated until the *Target* is reached.

*Progression* = [Phenology].ThermalTime

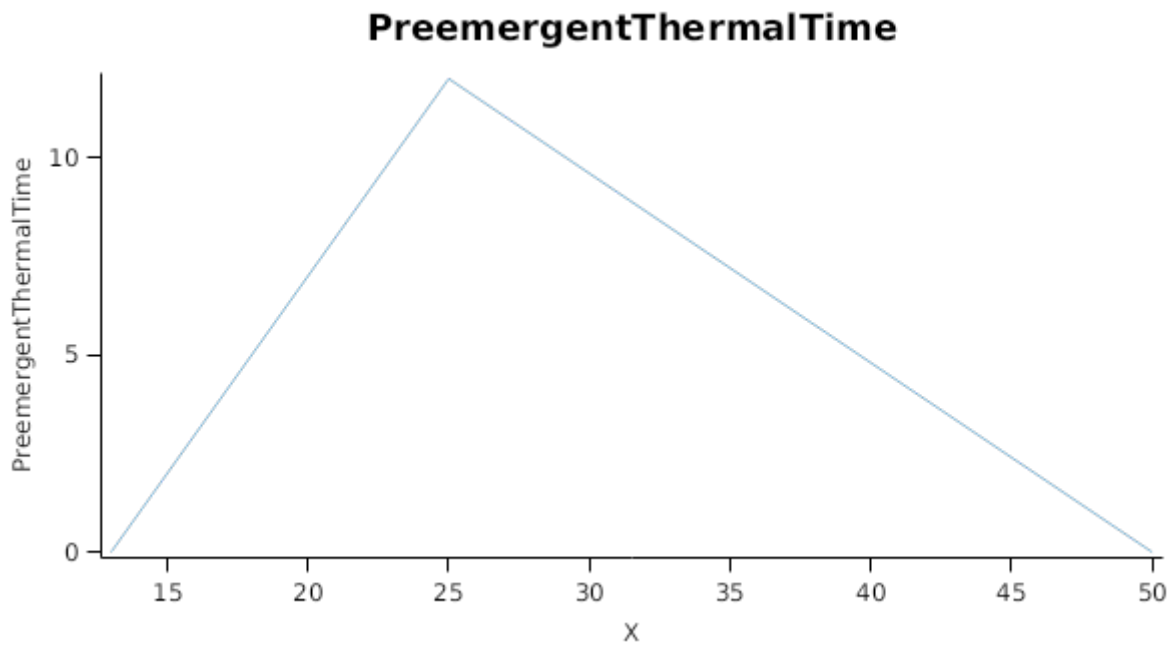
### 1.2.12 ReadyForHarvesting

It is the end phase in phenology and the crop will sit, unchanging, in this phase until it is harvested or removed by other method

### 1.2.13 PreemergentThermalTime

*PreemergentThermalTime* is calculated using linear interpolation

X	PreemergentThermalTime
13.0	0.0
25.0	12.0
50.0	0.0

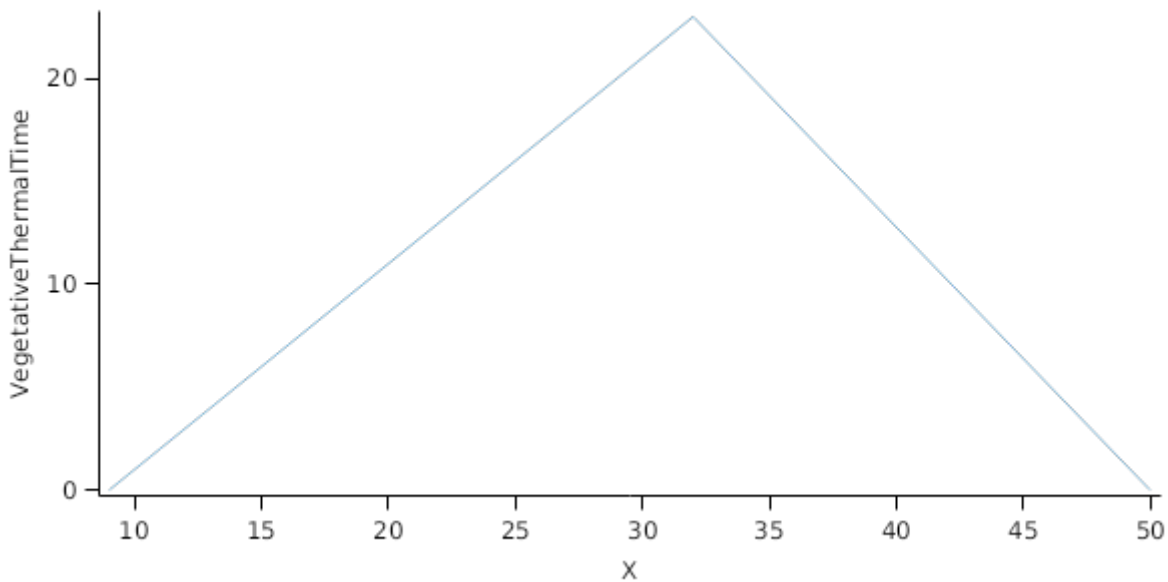


### 1.2.14 VegetativeThermalTime

*VegetativeThermalTime* is calculated using linear interpolation

X	VegetativeThermalTime
9.0	0.0
32.0	23.0
50.0	0.0

## VegetativeThermalTime

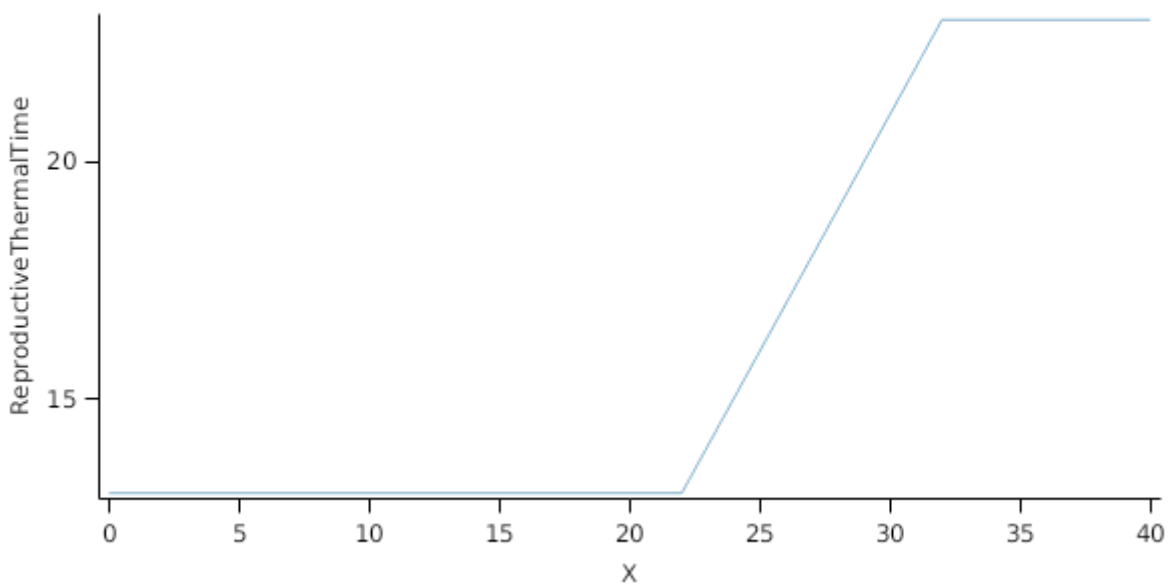


### 1.2.15 ReproductiveThermalTime

*ReproductiveThermalTime* is calculated using linear interpolation

X	ReproductiveThermalTime
0.0	13.0
22.0	13.0
32.0	23.0
40.0	23.0

## ReproductiveThermalTime



### 1.2.16 Photoperiod

Returns the duration of the day, or photoperiod, in hours. This is calculated using the specified latitude (given in the weather file) and twilight sun angle threshold. If a variable called `ClimateControl.PhotoPeriod` is found in the simulation, it will be used instead.

The day length is calculated with `\ref MathUtilities.DayLength`.



*Twilight = -6 (degrees)*

### **1.2.17 EmergenceDAS**

Before Emergence

PreEventValue = 0

On Emergence the value is set to:

*PostEventValue = [Plant].DaysAfterSowing*

### **1.2.18 FloweringDAS**

Before StartFlowering

PreEventValue = 0

On StartFlowering the value is set to:

*PostEventValue = [Plant].DaysAfterSowing*

### **1.2.19 MaturityDAS**

Before Maturity

PreEventValue = 0

On Maturity the value is set to:

*PostEventValue = [Plant].DaysAfterSowing*

### **1.2.20 DaysAfterEmergence**

*DaysAfterEmergence = Accumulated OneDay between emergence and harvestripe*

OneDay = 1

## **1.3 Leaf**

This organ is simulated using a SimpleLeaf organ type. It provides the core functions of intercepting radiation, producing biomass through photosynthesis, and determining the plant's transpiration demand. The model also calculates the growth, senescence, and detachment of leaves. SimpleLeaf does not distinguish leaf cohorts by age or position in the canopy.

Radiation interception and transpiration demand are computed by the MicroClimate model. This model takes into account competition between different plants when more than one is present in the simulation. The values of canopy Cover, LAI, and plant Height (as defined below) are passed daily by SimpleLeaf to the MicroClimate model. MicroClimate uses an implementation of the Beer-Lambert equation to compute light interception and the Penman-Monteith equation to calculate potential evapotranspiration.

These values are then given back to SimpleLeaf which uses them to calculate photosynthesis and soil water demand.

NOTE: the summary above is used in the Apsim's autdoc.

SimpleLeaf has two options to define the canopy: the user can either supply a function describing LAI or a function describing canopy cover directly. From either of these functions SimpleLeaf can obtain the other property using the Beer-Lambert equation with the specified value of extinction coefficient. The effect of growth rate on transpiration is captured by the Fractional Growth Rate (FRGR) function, which is passed to the MicroClimate model.

### **1.3.1 Initial Dry Matter**

*InitialWt = MassPerPlant x [Peanut].SowingData.Population*

MassPerPlant = 0.06

### **1.3.2 Dry Matter Demand**

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

#### **1.3.2.1 DMDemands**

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

Growth in leaf area is used to calculate the leaf dry matter demand. Note that leaf biomass is assumed to consist of entirely structural dry matter.

$$\text{Structural} = [\text{Leaf}].\text{DeltaLAI} / [\text{Leaf}].\text{SpecificArea}$$

$$\text{Storage} = 0$$

$$\text{Metabolic} = 0$$

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

### 1.3.3 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

#### 1.3.3.1 NDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$$\text{Structural} = [\text{Leaf}].\text{minimumNconc} \times [\text{Leaf}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{Metabolic} = \text{MetabolicNconc} \times [\text{Leaf}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{MetabolicNconc} = [\text{Leaf}].\text{criticalNConc} - [\text{Leaf}].\text{minimumNconc}$$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$$\text{Storage} = [\text{Leaf}].\text{maximumNconc} \times ([\text{Leaf}].\text{Live.Wt} + \text{potentialAllocationWt}) - [\text{Leaf}].\text{Live.N}$$

The demand for storage N is further reduced by a factor specified by the [Leaf].NitrogenDemandSwitch.

$$\text{NitrogenDemandSwitch} = 1$$

$$\text{MaxNconc} = [\text{Leaf}].\text{maximumNconc}$$

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

### 1.3.4 Nitrogen Concentration Thresholds

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

#### 1.3.4.1 NDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$$\text{Structural} = [\text{Leaf}].\text{minimumNconc} \times [\text{Leaf}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{Metabolic} = \text{MetabolicNconc} \times [\text{Leaf}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{MetabolicNconc} = [\text{Leaf}].\text{criticalNConc} - [\text{Leaf}].\text{minimumNconc}$$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$$\text{Storage} = [\text{Leaf}].\text{maximumNconc} \times ([\text{Leaf}].\text{Live.Wt} + \text{potentialAllocationWt}) - [\text{Leaf}].\text{Live.N}$$

The demand for storage N is further reduced by a factor specified by the [Leaf].NitrogenDemandSwitch.

$$\text{NitrogenDemandSwitch} = 1$$

$$\text{MaxNconc} = [\text{Leaf}].\text{maximumNconc}$$

$$\text{QStructuralPriority} = 1$$

QMetabolicPriority = 1

QStoragePriority = 1

### 1.3.5 Dry Matter Supply

DMReallocationFactor = 0

DMRetranslocationFactor = 0.5

### 1.3.6 Photosynthesis

Biomass fixation is modelled as the product of intercepted radiation and its conversion efficiency, the radiation use efficiency (RUE) (Monteith et al., 1977).

This approach simulates net photosynthesis rather than providing separate estimates of growth and respiration.

The potential photosynthesis calculated using RUE is then adjusted according to stress factors, these account for plant nutrition (FN), air temperature (FT), vapour pressure deficit (FVPD), water supply (FW) and atmospheric CO<sub>2</sub> concentration (FCO<sub>2</sub>).

NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

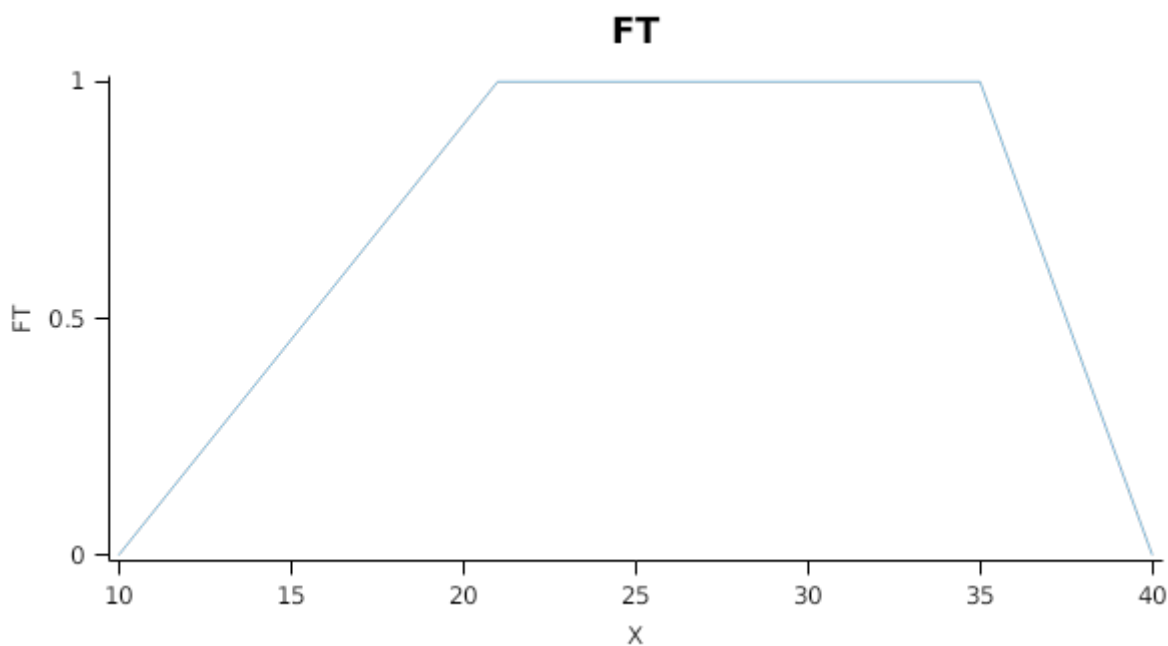
This approach is based upon the value used in Robertson et al., 2002 increased to account for partitioning to roots.

RUE = 1.5

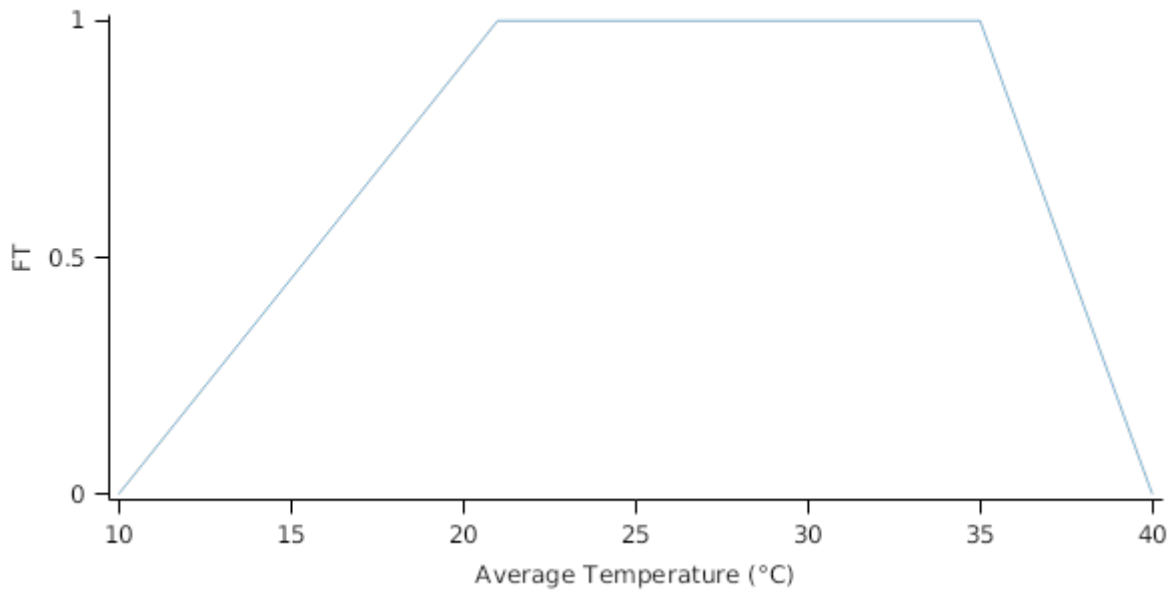
FT is calculated as a function of daily min and max temperatures, these are weighted toward max temperature according to the specified MaximumTemperatureWeighting factor. A value equal to 1.0 means it will use max temperature, a value of 0.5 means average temperature.

MaximumTemperatureWeighting = 0.75

X	FT
10.0	0.0
21.0	1.0
35.0	1.0
40.0	0.0



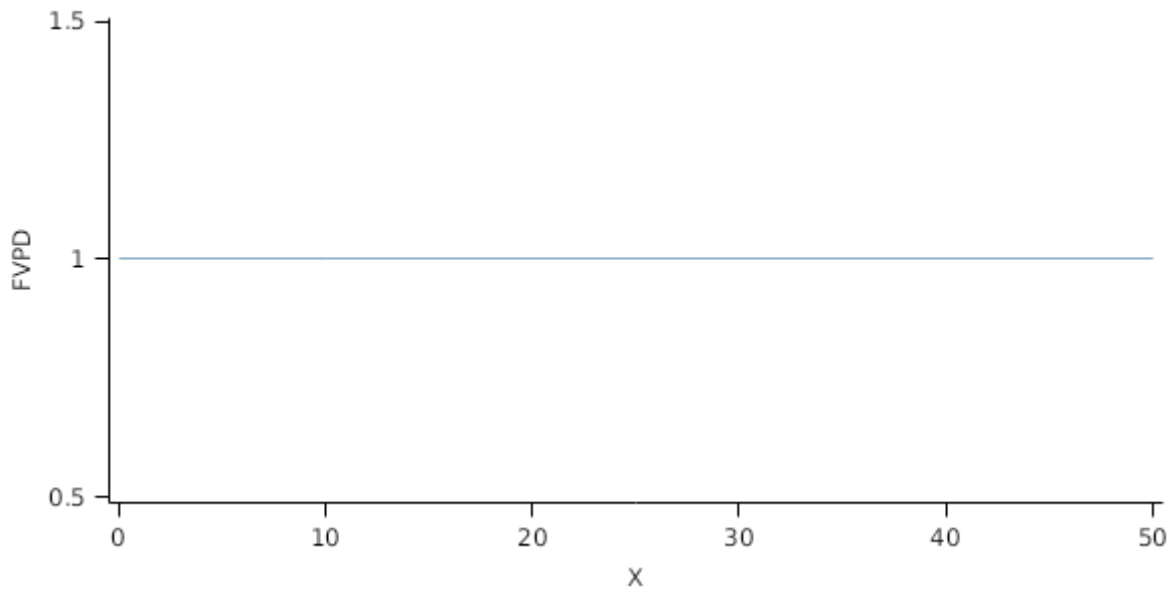
### FT



FVPD is calculated using linear interpolation

X	FVPD
0.0	1.0
10.0	1.0
50.0	1.0

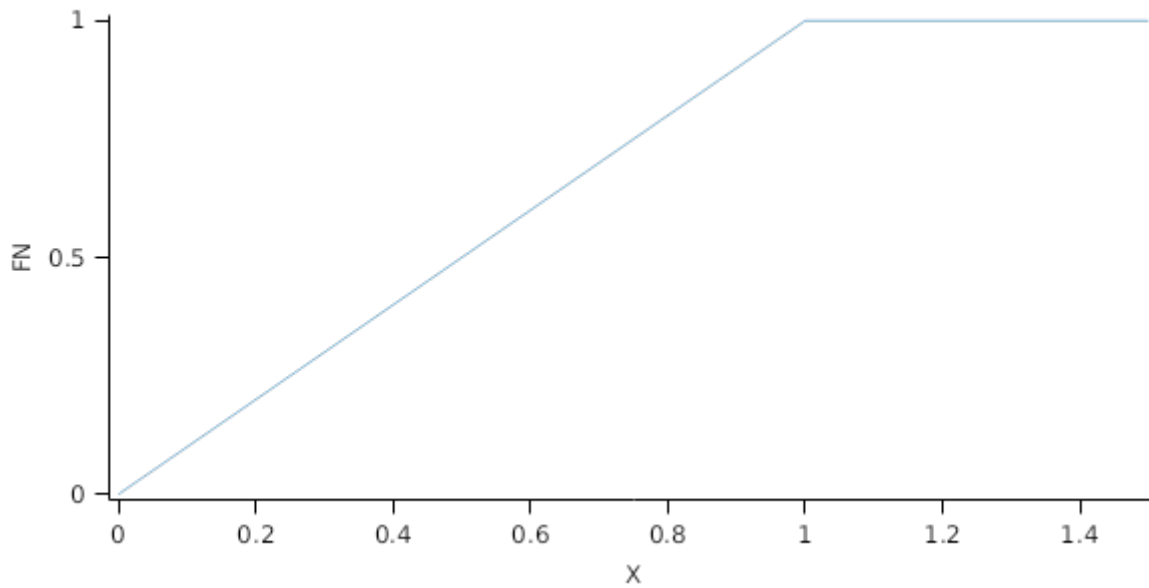
### FVPD



FN is calculated using linear interpolation

X	FN
0.0	0.0
1.0	1.0
1.5	1.0

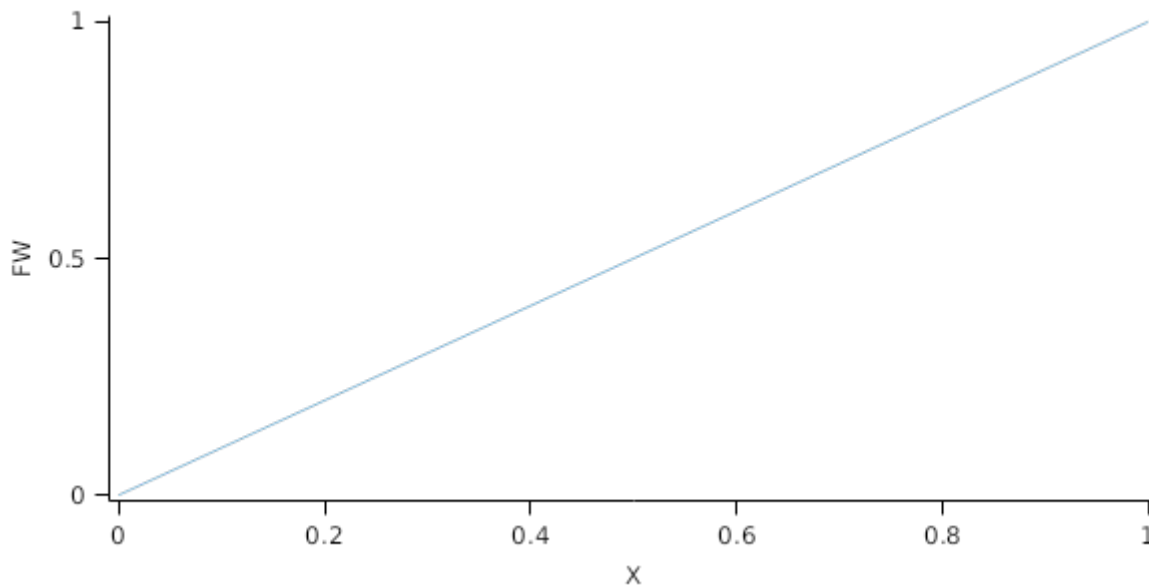
## FN



FW is calculated using linear interpolation

X	FW
0.0	0.0
1.0	1.0

## FW



This model calculates the CO<sub>2</sub> impact on RUE using the approach of [Reyenga et al., 1999](#).

For C3 plants,

$$F_{CO_2} = (CO_2 - CP) \times (350 + 2 \times CP) / (CO_2 + 2 \times CP) \times (350 - CP)$$

where CP, is the compensation point calculated from daily average temperature (T) as

$$CP = (163.0 - T) / (5.0 - 0.1 \times T)$$

For C4 plants,

$$F_{CO_2} = 0.000143 \times CO_2 + 0.95$$

$$RadInt = [Leaf].RadiationIntercepted$$

### 1.3.7 Nitrogen Supply

This model assumes that all metabolic and storage N is available for reallocation from senescing leaves.

$N_{\text{ReallocationFactor}} = 1$

$N_{\text{RetranslocationFactor}} = 0.5$

### 1.3.8 Canopy Properties

Leaf has been defined with a LAIFunction, cover is calculated using the Beer-Lambert equation.

$\text{Area} = \text{Initial} + \text{Change}$

$\text{Initial} = \text{AreaPerPlant} \times [\text{Peanut}].\text{SowingData}.\text{Population}$

$\text{AreaPerPlant} = 0.005$

$\text{Change} = \text{AreaGrowth} - \text{AreaLoss}$

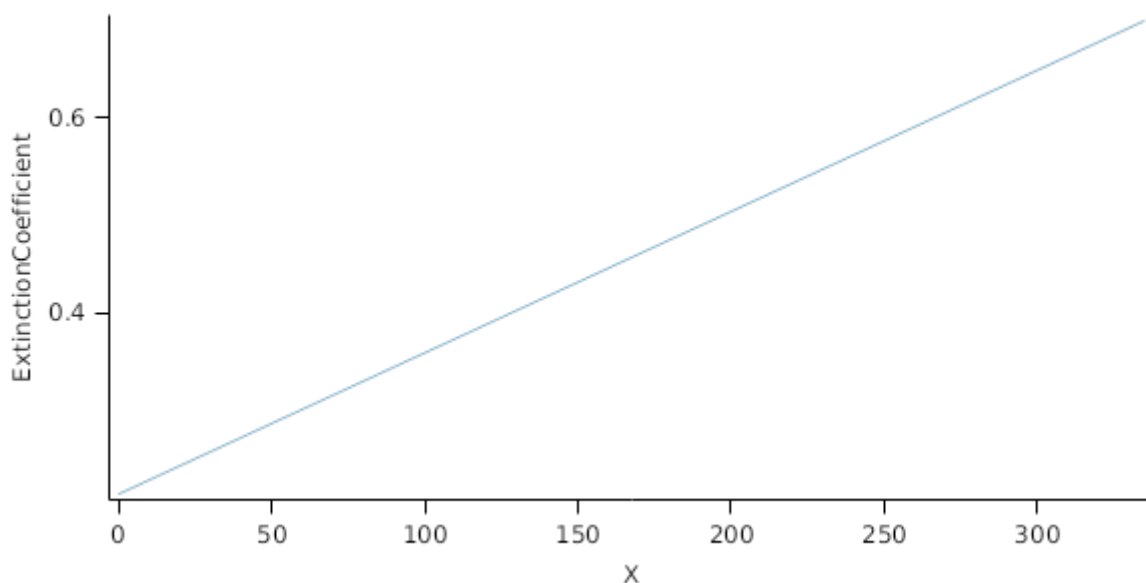
**AreaGrowth** is a daily accumulation of the values of functions listed below between the Emergence and EndCanopyDevelopment stages. Function values added to the accumulate total each day are:

**AreaLoss** is a daily accumulation of the values of functions listed below between the EndCanopyDevelopment and HarvestRipe stages. Function values added to the accumulate total each day are:

*ExtinctionCoefficient* is calculated using linear interpolation

X	ExtinctionCoefficient
0.0	0.2
335.0	0.7

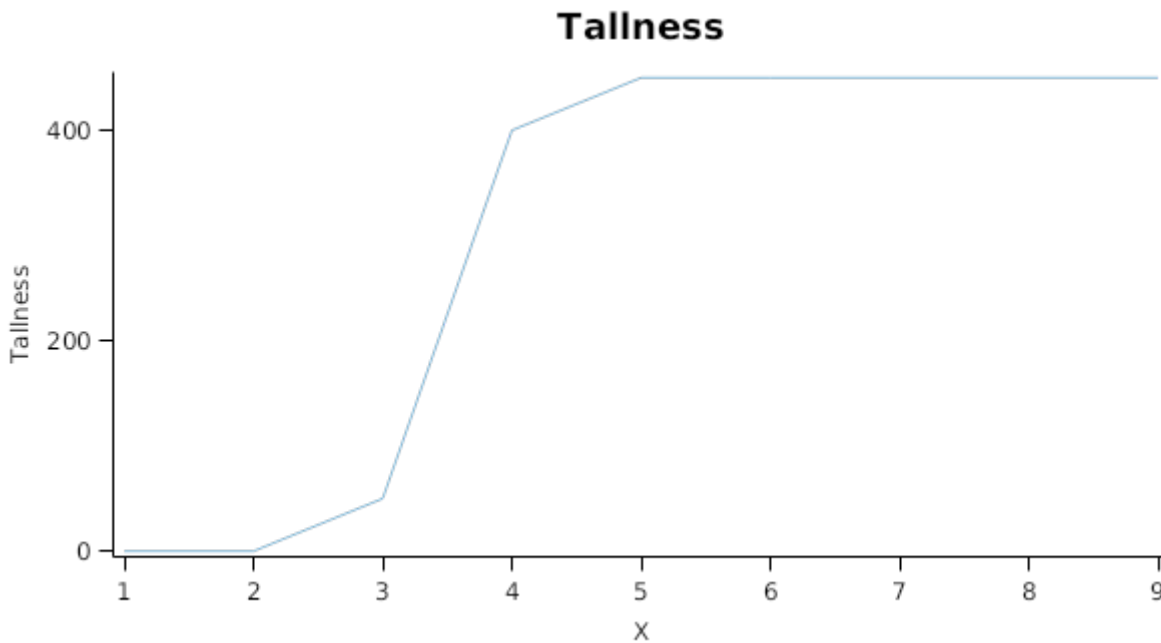
**ExtinctionCoefficient**



*Tallness* is calculated using linear interpolation

X	Tallness
1.0	0.0
2.0	0.0
3.0	50.0
4.0	400.0

X	Tallness
5.0	450.0
6.0	450.0
9.0	450.0



### 1.3.9 StomatalConductance

Stomatal Conductance (gs) is calculated for use within the micromet model by adjusting a value provided for an atmospheric CO<sub>2</sub> concentration of 350 ppm. The impact of other stresses (e.g. Temperature, N) are captured through the modifier, Frgr.

$$gs = G_{max350} \times FRGR \times stomatalConductanceCO_2Modifier$$

This model calculates the CO<sub>2</sub> impact on stomatal conductance using the approach of [Elli et al., 2020](#).

$$StomatalConductanceCO_2Modifier = PhotosynthesisCO_2Modifier \times (350 - CP) / (CO_2 - CP)$$

where CP, is the compensation point calculated from daily average temperature (T) as

$$CP = (163.0 - T) / (5.0 - 0.1 * T)$$

$$PhotosynthesisCO_2Modifier = [Leaf].Photosynthesis.FCO_2$$

### 1.3.10 Senescence and Detachment

The proportion of live biomass that senesces and moves into the dead pool each day is quantified by the SenescenceRate.

SenescenceRate is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Vegetative has a value between Emergence and EndPodDevelopment calculated as:

$$Rate = 0$$

Leaf senescence rate is calculated to ensure that all leaves are senesced by crop maturity.

Reproductive has a value between EndPodDevelopment and Maturity calculated as:

Leaf senescence rate is calculated to ensure that all leaves are senesced by crop maturity.

$$Rate = [Phenology].ThermalTime / TTRemaining / Modifier$$

$$TTRemaining = TTRemaining - TTComplete$$

$TT_{Required} = [Phenology].LateGrainFilling.Target + [Phenology].Maturing.Target + 1000$

$TT_{Complete} = [Phenology].LateGrainFilling.ProgressThroughPhase + [Phenology].Maturing.ProgressThroughPhase$

Modifier = 1

Leaf senescence rate is calculated to ensure that all leaves are senesced by crop maturity.

Senescent has a value between Maturity and HarvestRipe calculated as:

Leaf senescence rate is calculated to ensure that all leaves are senesced by crop maturity.

Rate = 0.1

Leaf detaches 5% of its dead biomass each day, passing it to the surface organic matter model for decomposition.

### 1.3.11 Biomass removal

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	100	100
Cut	0	0	100	100
Prune	0	0	0	0
Graze	0	0	0	0

## 1.4 Grain

This organ uses a generic model for plant reproductive components. Yield is calculated from its components in terms of organ number and size (for example, grain number and grain size).

### 1.4.1 Constants

MinimumNConc = 0.05

MaximumNConc = 0.058

WaterContent = 0.13

DMConversionEfficiency = 0.89

RemobilisationCost = 0

CarbonConcentration = 0.4

This parameter is not used. Grain growth is calculated from harvest index increase.

MaximumPotentialGrainSize = 0

Grain number is not estimated by this model.

NumberFunction = 0

### 1.4.2 FillingDuration

$FillingDuration = [Phenology].EarlyGrainFilling.Target + [Phenology].MidGrainFilling.Target + [Phenology].LateGrainFilling.Target$

### 1.4.3 DMDemandFunction

DMDemandFunction is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.



GrainFilling has a value between StartGrainFilling and EndGrainFill calculated as:

$$HI_{GrainDemand} = [AboveGround].Wt \times HarvestIndexIncrease \times [Phenology].ThermalTime$$

$$HarvestIndexIncrease = OutstandingHarvestIndexIncrease / TTRemaining$$

$$OutstandingHarvestIndexIncrease = \text{Max}(Increase, Zero)$$

Where:

$$Increase = [Grain].PotentialHarvestIndex - [Grain].HarvestIndex$$

$$Zero = 0$$

$$TTRemaining = ReproductiveTT - ReproductiveTTComplete$$

$$ReproductiveTT = [Phenology].EarlyGrainFilling.Target + [Phenology].MidGrainFilling.Target + [Phenology].LateGrainFilling.Target$$

$$ReproductiveTTComplete = [Phenology].EarlyGrainFilling.ProgressThroughPhase + [Phenology].MidGrainFilling.ProgressThroughPhase + [Phenology].LateGrainFilling.ProgressThroughPhase$$

#### 1.4.4 NFillingRate

$$NFillingRate = [Grain].MaximumNConc \times [Grain].DMDemandFunction \times [Grain].DMConversionEfficiency$$

#### 1.4.5 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	100	100	0	0
Cut	0	0	100	100
Prune	0	0	0	0
Graze	0	0	0	0

#### 1.4.6 HarvestIndex

$$HarvestIndex = [Grain].Wt / [AboveGround].Wt$$

#### 1.4.7 PotentialHarvestIndex

$$PotentialHarvestIndex = [Pod].PotentialHarvestIndex \times GrainFraction$$

$$GrainFraction = 0.72 \text{ (0-1)}$$

#### 1.4.8 DMDemandPriorityFactors

This class holds the functions for calculating the absolute demands for each biomass fraction.

$$\text{Structural} = 1$$

$$\text{Metabolic} = 1$$

$$\text{Storage} = 1$$

#### 1.4.9 NDemandPriorityFactors

This class holds the functions for calculating the absolute demands for each biomass fraction.

$$\text{Structural} = 1$$

Metabolic = 1

Storage = 1

## 1.5 Root

The root model calculates root growth in terms of rooting depth, biomass accumulation and subsequent root length density in each soil layer.

### 1.5.1 Growth

Roots grow downwards through the soil profile, with initial depth determined by sowing depth and the growth rate determined by RootFrontVelocity. The RootFrontVelocity is modified by multiplying it by the soil's XF value, which represents any resistance posed by the soil to root extension.

$$\text{Root Depth Increase} = \text{RootFrontVelocity} \times \text{XF}_i \times \text{RootDepthStressFactor}$$

where  $i$  is the index of the soil layer at the rooting front.

Root depth is also constrained by a maximum root depth.

Root length growth is calculated using the daily DM partitioned to roots and a specific root length. Root proliferation in layers is calculated using an approach similar to the generalised equimarginal criterion used in economics. The uptake of water and N per unit root length is used to partition new root material into layers of higher 'return on investment'. For example, the Root Activity for water is calculated as

$$\text{RAw}_i = -\text{WaterUptake}_i / \text{LiveRootWt}_i \times \text{LayerThickness}_i \times \text{ProportionThroughLayer}$$

The amount of root mass partitioned to a layer is then proportional to root activity

$$\text{DMAllocated}_i = \text{TotalDMAllocated} \times \text{RAw}_i / \text{TotalRAw}$$

### 1.5.2 Dry Matter Demands

A daily DM demand is provided to the organ arbitrator and a DM supply returned. By default, 100% of the dry matter (DM) demanded from the root is structural. The daily loss of roots is calculated using a SenescenceRate function. All senesced material is automatically detached and added to the soil FOM.

### 1.5.3 Nitrogen Demands

The daily structural N demand from root is the product of total DM demand and the minimum N concentration. Any N above this is considered Storage and can be used for retranslocation and/or reallocation as the respective factors are set to values other than zero.

### 1.5.4 Nitrogen Uptake

Potential N uptake by the root system is calculated for each soil layer ( $i$ ) that the roots have extended into. In each layer potential uptake is calculated as the product of the mineral nitrogen in the layer, a factor controlling the rate of extraction ( $k\text{NO}_3$  or  $k\text{NH}_4$ ), the concentration of N form (ppm), and a soil moisture factor (NUptakeSWFactor) which typically decreases as the soil dries.  $\text{NO}_3 \text{ uptake} = \text{NO}_3 \times k\text{NO}_3 \times \text{NO}_3_{\text{ppm}, i} \times \text{NUptakeSWFactor}$   $\text{NH}_4 \text{ uptake} = \text{NH}_4 \times k\text{NH}_4 \times \text{NH}_4_{\text{ppm}, i} \times \text{NUptakeSWFactor}$  As can be seen from the above equations, the values of  $k\text{NO}_3$  and  $k\text{NH}_4$  equate to the potential fraction of each mineral N pool which can be taken up per day for wet soil when that pool has a concentration of 1 ppm. Nitrogen uptake demand is limited to the maximum daily potential uptake (MaxDailyNUptake) and the plant's N demand. The former provides a means to constrain N uptake to a maximum value observed in the field for the crop as a whole. The demand for soil N is then passed to the soil arbitrator which determines how much of the N uptake demand each plant instance will be allowed to take up.

### 1.5.5 Water Uptake

Potential water uptake by the root system is calculated for each soil layer that the roots have extended into. In each layer potential uptake is calculated as the product of the available water in the layer (water above LL limit) and a factor controlling the rate of extraction (KL). The values of both LL and KL are set in the soil interface and KL may be further modified by the crop via the KLModifier function.  $\text{SW uptake} = (\text{SW}_i - \text{LL}_i) \times \text{KL}_i \times \text{KLModifier}$

### 1.5.6 Constants

SenescenceRate = 0.005

MaximumRootDepth = 1000000

MaximumNConc = 0.01

MinimumNConc = 0.005

KNO3 = 0.02

The current model version considers only NO3 uptake but NH4 option can be activated by the user . We do not expect activation of NH4 to change total plant N uptake because the nitrification rate (conversion of NH4 to NO3) is very fast in the model. In the future we will re-evaluate this.

KNH4 = 0

This approach is based upon that used in [Robertson et al., 2002](#).

SpecificRootLength = 60 (m/g)

DMConversionEfficiency = 1

MaintenanceRespirationFunction = 1

RemobilisationCost = 0

CarbonConcentration = 0.4

RootDepthStressFactor = 1

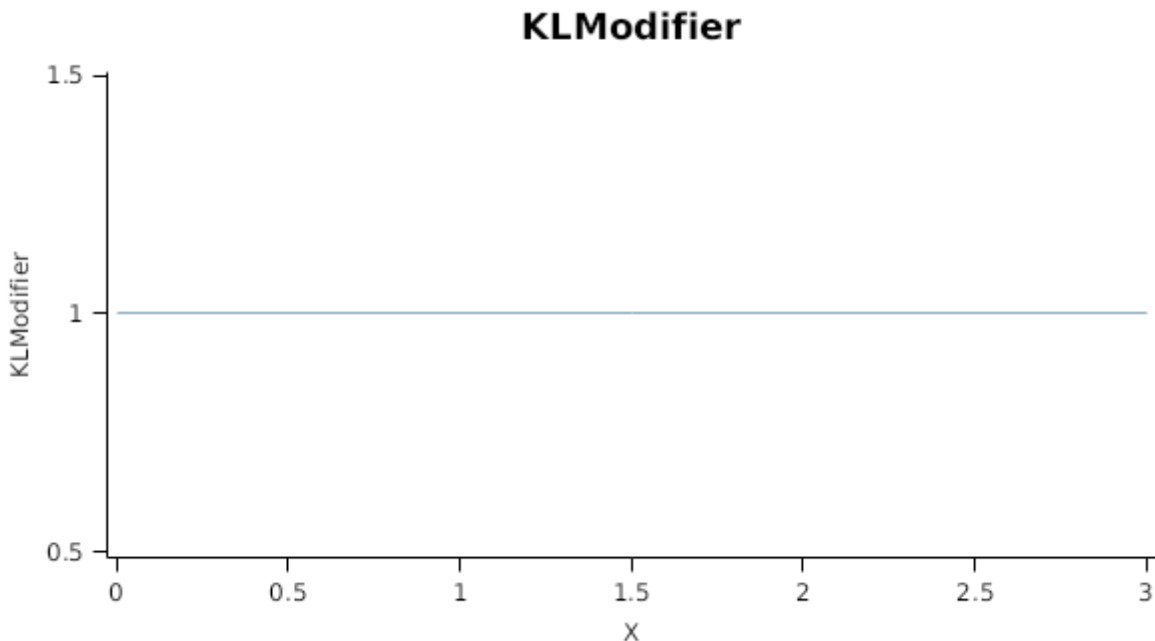
### 1.5.7 RootShape

This model calculates the proportion of each soil layer occupied by roots.

### 1.5.8 KLModifier

*KLModifier* is calculated using linear interpolation

X	KLModifier
0.0	1.0
3.0	1.0



### 1.5.9 TemperatureEffect

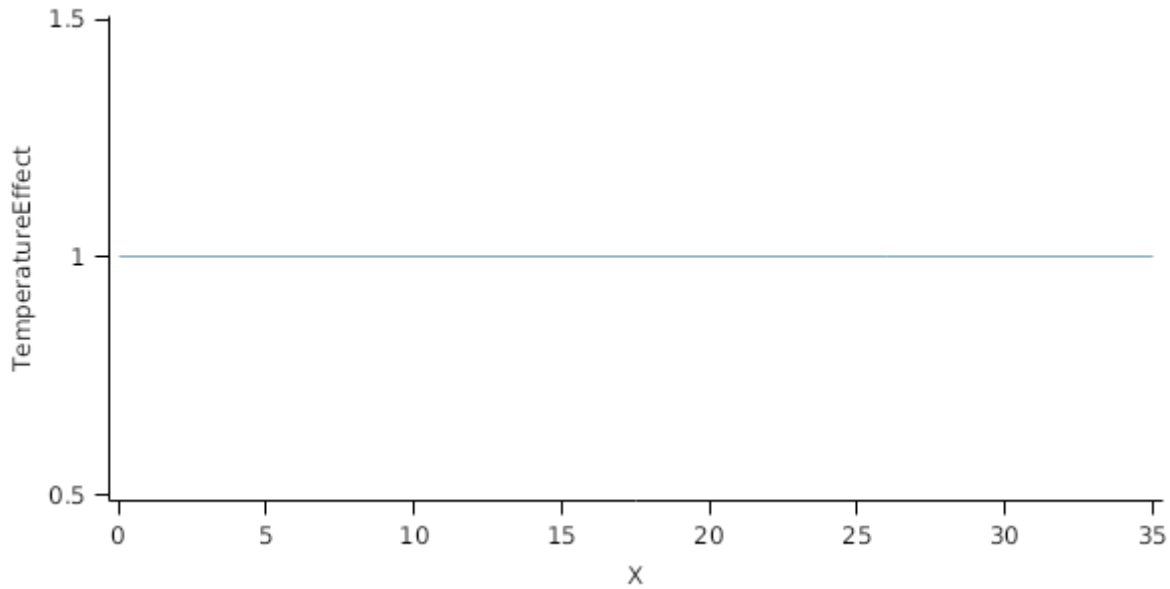
TemperatureEffect is the average of sub-daily values from a XYPairs.

Firstly 3-hourly estimates of air temperature ( $T_a$ ) are interpolated using the method of [Jones et al., 1986](#) which assumes a sinusoidal temperature. pattern between  $T_{max}$  and  $T_{min}$ .

Each of the interpolated air temperatures are then passed into the following Response and the Average taken to give daily TemperatureEffect

X	TemperatureEffect
0.0	1.0
26.0	1.0
35.0	1.0

### TemperatureEffect

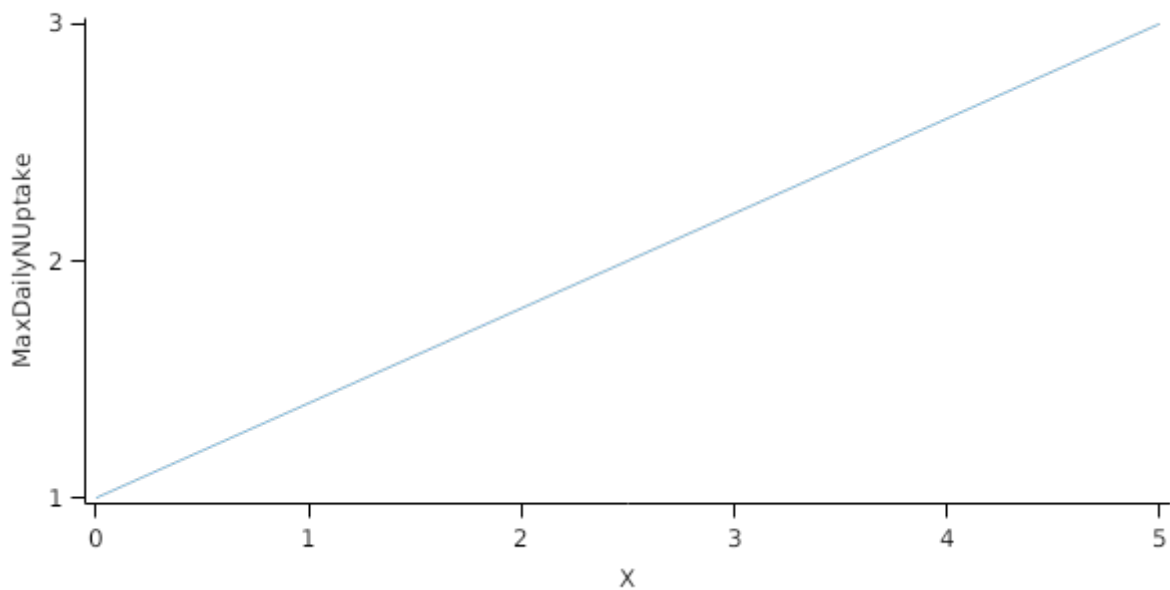


### 1.5.10 MaxDailyNUptake

*MaxDailyNUptake* is calculated using linear interpolation

X	MaxDailyNUptake
0.0	1.0
5.0	3.0

### MaxDailyNUptake



### 1.5.11 RootFrontVelocity

RootFrontVelocity is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

This approach is based upon that used in [Robertson et al., 2002](#).

PreEmergence has a value between Germination and Emergence calculated as:

Function = 15

early has a value between Emergence and StartGrainFilling calculated as:

Function = 35

late has a value between StartGrainFilling and Maturity calculated as:

Function = 0

### 1.5.12 NitrogenDemandSwitch

NitrogenDemandSwitch has a value between Emergence and EndGrainFill calculated as:

Constant = 1

### 1.5.13 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

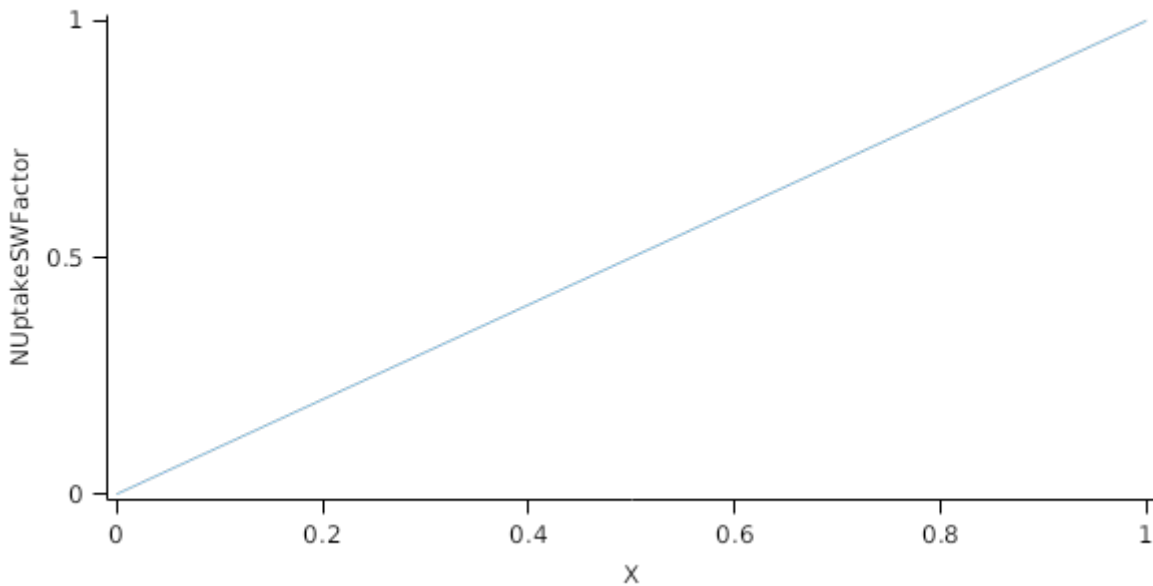
Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	20	0
Cut	0	0	20	0
Prune	0	0	0	0
Graze	0	0	0	0

### 1.5.14 NUptakeSWFactor

*NUptakeSWFactor* is calculated using linear interpolation

X	NUptakeSWFactor
0.0	0.0
1.0	1.0

## NUptakeSWFactor



### 1.5.15 DMDemands

#### 1.5.15.1 DMDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$Structural = DMDemandFunction \times StructuralFraction$

Returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs.

$DMDemandFunction = PartitionFraction \times [Arbitrator].DM.TotalFixationSupply$

PartitionFraction is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Early has a value between Emergence and StartFlowering calculated as:

Function = 0.2

Middle has a value between StartFlowering and StartGrainFilling calculated as:

Function = 0.2

Late has a value between StartGrainFilling and Maturity calculated as:

Function = 0.05

StructuralFraction = 1

Metabolic = 0

The partitioning of daily growth to storage biomass is based on a storage fraction.

$StorageFraction = 1 - [Root].DMDemands.Structural.StructuralFraction$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

### 1.5.16 NDemands

#### 1.5.16.1 NDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$Structural = [Root].minimumNconc \times [Root].potentialDMAAllocation.Structural$

$Metabolic = MetabolicNconc \times [Root].potentialDMAAllocation.Structural$

$MetabolicNconc = [Root].criticalNConc - [Root].minimumNconc$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$Storage = [Root].maximumNconc \times ([Root].Live.Wt + potentialAllocationWt) - [Root].Live.N$

The demand for storage N is further reduced by a factor specified by the [Root].NitrogenDemandSwitch.

$NitrogenDemandSwitch = [Root].nitrogenDemandSwitch$

$MaxNconc = [Root].maximumNconc$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

### 1.5.17 CriticalNConc

$CriticalNConc = [Root].MinimumNConc$

### 1.5.18 InitialWt

This class holds the functions for calculating the absolute demands for each biomass fraction.

Structural = 0.1 (g/plant)

Metabolic = 0

Storage = 0

## 1.6 Nodule

This organ simulates the root structure associate with symbiotic N-fixing bacteria. It provides the core functions of determining N fixation supply and related costs. It also calculates the growth, senescence and detachment of nodules.

### 1.6.1 Dry Matter Demand

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

This class holds the functions for calculating the absolute demands for each biomass fraction.

$Structural = DMDemandFunction \times StructuralFraction$

Returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs.

$DMDemandFunction = PartitionFraction \times [Arbitrator].DM.TotalFixationSupply$

PartitionFraction is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Early has a value between Emergence and StartFlowering calculated as:

Function = 0.03

Middle has a value between StartFlowering and StartGrainFilling calculated as:

Function = 0.02

Late has a value between StartGrainFilling and Maturity calculated as:

Function = 0

StructuralFraction = 1 (g/g)

Metabolic = 0

The partitioning of daily growth to storage biomass is based on a storage fraction.

$$\text{StorageFraction} = 1 - [\text{Nodule}].\text{DMDemands}.\text{Structural}.\text{StructuralFraction}$$

### 1.6.2 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

This class holds the functions for calculating the absolute demands for each biomass fraction.

$$\text{Structural} = [\text{Nodule}].\text{minimumNconc} \times [\text{Nodule}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{Metabolic} = \text{MetabolicNconc} \times [\text{Nodule}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{MetabolicNconc} = [\text{Nodule}].\text{criticalNConc} - [\text{Nodule}].\text{minimumNconc}$$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$$\text{Storage} = [\text{Nodule}].\text{maximumNconc} \times ([\text{Nodule}].\text{Live}.\text{Wt} + \text{potentialAllocationWt}) - [\text{Nodule}].\text{Live}.\text{N}$$

The demand for storage N is further reduced by a factor specified by the [Nodule].NitrogenDemandSwitch.

$$\text{NitrogenDemandSwitch} = [\text{Nodule}].\text{nitrogenDemandSwitch}$$

$$\text{MaxNconc} = [\text{Nodule}].\text{maximumNconc}$$

$$\text{MinimumNConc} = 0.01$$

$$\text{CriticalNConc} = [\text{Nodule}].\text{MinimumNConc}$$

$$\text{MaximumNConc} = 0.02$$

The demand for N is reduced by a factor specified by the NitrogenDemandSwitch.

NitrogenDemandSwitch has a value between Emergence and EndGrainFill calculated as:

$$\text{Constant} = 1$$

### 1.6.3 Dry Matter Supply

Nodule does not reallocate DM when senescence of the organ occurs.

Nodule does not retranslocate non-structural DM.

### 1.6.4 Nitrogen Supply

Nodule does not reallocate N when senescence of the organ occurs.

Nodule does not retranslocate non-structural N.

*FixationRate* = *DailyPotentialFixationRate* until StartGrainFilling after which the value is fixed.

$$\text{DailyPotentialFixationRate} = \text{Min}(\text{PotentialFixationRate}, \text{MaximumFixationRate})$$

Where:

$$\text{PotentialFixationRate} = [\text{AboveGroundLive}].\text{Wt} \times \text{SpecificFixationRate}$$

SpecificFixationRate is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

VegetativeGrowth has a value between Emergence and StartFlowering calculated as:

$$\text{Rate} = 0.006$$

ReproductiveGrowth has a value between StartFlowering and EndGrainFill calculated as:

$$\text{Rate} = 0.002$$

$$\text{MaximumFixationRate} = 0.6$$

### 1.6.5 Senescence and Detachment



Nodule has senescence parameterised to zero so all biomass in this organ will remain alive.

Nodule has detachment parameterised to zero so all biomass in this organ will remain with the plant until a defoliation or harvest event occurs.

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	20	0
Cut	0	0	20	0
Prune	0	0	0	0
Graze	0	0	0	0

## 1.7 Shell

### 1.7.1 Shell

This organ is simulated using a GenericOrgan type. It is parameterised to calculate the growth, senescence, and detachment of any organ that does not have specific functions.

### 1.7.2 Dry Matter Demand

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

#### 1.7.2.1 DMDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$$\text{Structural} = \text{DMDemandFunction} \times \text{StructuralFraction}$$

DMDemandFunction is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

PodDevelopment has a value between StartPodDevelopment and EndPodDevelopment calculated as:

$$\text{HIPodDemand} = [\text{AboveGround}].\text{Wt} \times \text{HarvestIndexIncrease} \times [\text{Phenology}].\text{ThermalTime}$$

$$\text{HarvestIndexIncrease} = \text{OutstandingHarvestIndexIncrease} / \text{TTRemaining}$$

$$\text{OutstandingHarvestIndexIncrease} = \text{Max}(\text{Increase}, \text{Zero})$$

Where:

$$\text{Increase} = [\text{Shell}].\text{PotentialHarvestIndex} - [\text{Shell}].\text{HarvestIndex}$$

$$\text{Zero} = 0$$

$$\text{TTRemaining} = \text{ShellGrowthTT} - \text{ShellGrowthTTComplete}$$

$$\text{ShellGrowthTT} = [\text{Phenology}].\text{EarlyPodDevelopment}.\text{Target} + [\text{Phenology}].\text{EarlyGrainFilling}.\text{Target} + [\text{Phenology}].\text{MidGrainFilling}.\text{Target}$$

$$\text{ShellGrowthTTComplete} = [\text{Phenology}].\text{EarlyPodDevelopment}.\text{ProgressThroughPhase} + [\text{Phenology}].\text{EarlyGrainFilling}.\text{ProgressThroughPhase} + [\text{Phenology}].\text{MidGrainFilling}.\text{ProgressThroughPhase}$$

$$\text{StructuralFraction} = 1$$

$$\text{Metabolic} = 0$$

The partitioning of daily growth to storage biomass is based on a storage fraction.

$StorageFraction = 1 - [Shell].DMDemands.Structural.StructuralFraction$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

### 1.7.3 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

#### 1.7.3.1 NDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$Structural = [Shell].minimumNconc \times [Shell].potentialDMAAllocation.Structural$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$Storage = [Shell].maximumNconc \times ([Shell].Live.Wt + potentialAllocationWt) - [Shell].Live.N$

The demand for storage N is further reduced by a factor specified by the [Shell].NitrogenDemandSwitch.

$NitrogenDemandSwitch = [Shell].nitrogenDemandSwitch$

$MaxNconc = [Shell].maximumNconc$

Metabolic = 0

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

#### 1.7.4 N Concentration Thresholds

MinimumNConc = 0.01

$CriticalNConc = [Shell].MinimumNConc$

MaximumNConc = 0.045

The demand for N is reduced by a factor specified by the NitrogenDemandSwitch.

NitrogenDemandSwitch has a value between StartPodDevelopment and EndPodDevelopment calculated as:

Constant = 1

#### 1.7.5 Dry Matter Supply

Shell does not reallocate DM when senescence of the organ occurs.

Shell will retranslocate 10% of non-structural DM each day.

#### 1.7.6 Nitrogen Supply

Shell can reallocate up to 100% of N that senesces each day if required by the plant arbitrator to meet N demands.

Shell can retranslocate up to 50% of non-structural N each day if required by the plant arbitrator to meet N demands.

#### 1.7.7 Senescence and Detachment

The proportion of live biomass that senesces and moves into the dead pool each day is quantified by the SenescenceRate.

SenescenceRate is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

ReproductivePhase has a value between StartPodDevelopment and Maturity calculated as:

$Rate = \text{Min}(Fraction, One)$

Where:

$Fraction = [Phenology].ThermalTime / TTRemaining$

$TTRemaining = ReproductiveTT - ReproductiveTT1Complete$

$ReproductiveTT = [Phenology].EarlyPodDevelopment.Target + [Phenology].EarlyGrainFilling.Target + [Phenology].MidGrainFilling.Target + [Phenology].LateGrainFilling.Target$

$ReproductiveTT1Complete = [Phenology].EarlyPodDevelopment.ProgressThroughPhase + [Phenology].EarlyGrainFilling.ProgressThroughPhase + [Phenology].MidGrainFilling.ProgressThroughPhase + [Phenology].LateGrainFilling.ProgressThroughPhase$

One = 1

Shell has detachment parameterised to zero so all biomass in this organ will remain with the plant until a defoliation or harvest event occurs.

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	100	100
Cut	0	0	100	100
Prune	0	0	0	0
Graze	0	0	0	0

## 1.8 Stem

### 1.8.1 Stem

This organ is simulated using a GenericOrgan type. It is parameterised to calculate the growth, senescence, and detachment of any organ that does not have specific functions.

### 1.8.2 Dry Matter Demand

The dry matter demand for the organ is calculated as defined in DMDemands, based on the DMDemandFunction and partition fractions for each biomass pool.

#### 1.8.2.1 DMDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$Structural = Fraction \times [Stem].PotentialGrowth$

Fraction is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

This approach is based upon that used in [Robertson et al., 2002](#).

StemGrowthPhase has a value between Emergence and EndCanopyDevelopment calculated as:

This approach is based upon that used in [Robertson et al., 2002](#).

Fraction = 0.9

PodAndGrainGrowth has a value between EndCanopyDevelopment and EndGrainFill calculated as:

Fraction = 0

Metabolic = 0

$$\text{Storage} = [\text{Stem}].\text{PotentialGrowth} \times \text{StorageFraction}$$

$$\text{StorageFraction} = 1 - [\text{Stem}].\text{DMDemands}.\text{Structural}.\text{Fraction}$$

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

### 1.8.3 Nitrogen Demand

The N demand is calculated as defined in NDemands, based on DM demand the N concentration of each biomass pool.

#### 1.8.3.1 NDemands

This class holds the functions for calculating the absolute demands and priorities for each biomass fraction.

$$\text{Structural} = [\text{Stem}].\text{minimumNconc} \times [\text{Stem}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{Metabolic} = \text{MetabolicNconc} \times [\text{Stem}].\text{potentialDMAAllocation}.\text{Structural}$$

$$\text{MetabolicNconc} = [\text{Stem}].\text{criticalNConc} - [\text{Stem}].\text{minimumNconc}$$

The partitioning of daily N supply to storage N attempts to bring the organ's N content to the maximum concentration.

$$\text{Storage} = [\text{Stem}].\text{maximumNconc} \times ([\text{Stem}].\text{Live}.\text{Wt} + \text{potentialAllocationWt}) - [\text{Stem}].\text{Live}.\text{N}$$

The demand for storage N is further reduced by a factor specified by the [Stem].NitrogenDemandSwitch.

$$\text{NitrogenDemandSwitch} = [\text{Stem}].\text{nitrogenDemandSwitch}$$

$$\text{MaxNconc} = [\text{Stem}].\text{maximumNconc}$$

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

#### 1.8.4 N Concentration Thresholds

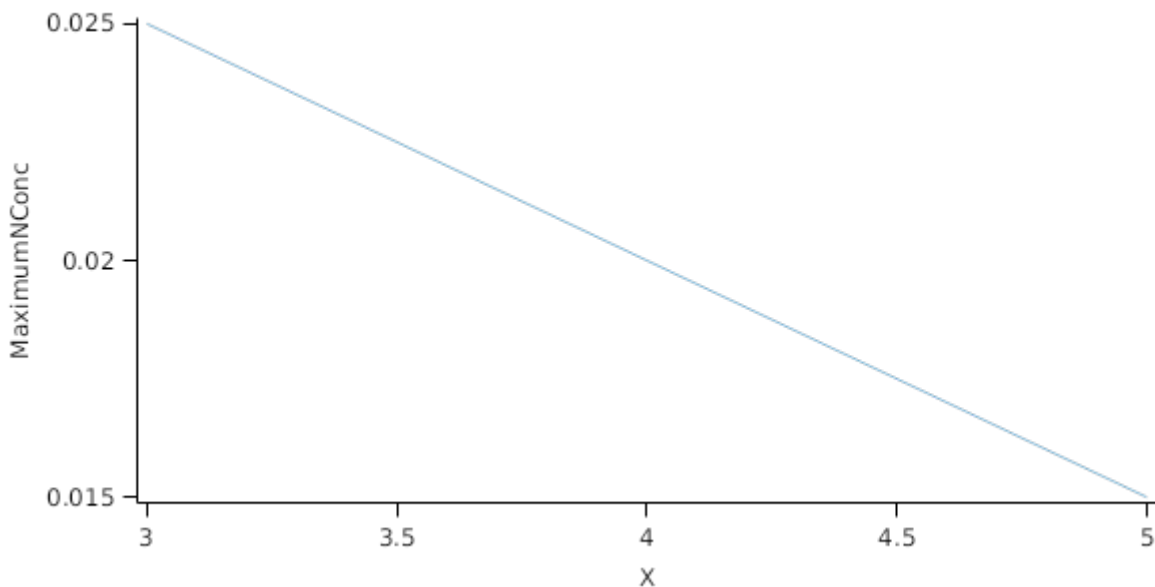
$$\text{MinimumNConc} = 0.006$$

$$\text{CriticalNConc} = [\text{Stem}].\text{MinimumNConc}$$

MaximumNConc is calculated using linear interpolation

X	MaximumNConc
3.0	0.0
4.0	0.0
5.0	0.0

## MaximumNConc



The demand for N is reduced by a factor specified by the NitrogenDemandSwitch.

NitrogenDemandSwitch has a value between Emergence and StartGrainFilling calculated as:

Constant = 1

### 1.8.5 Dry Matter Supply

Stem does not reallocate DM when senescence of the organ occurs.

The proportion of non-structural DM that is allocated each day is quantified by the DMReallocationFactor.

DMRetranslocationFactor is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Vegetative has a value between Emergence and EndCanopyDevelopment calculated as:

RetranslocationFactor = 0

GrainFilling has a value between EndCanopyDevelopment and EndGrainFill calculated as:

RetranslocationFactor = 0.2

### 1.8.6 Nitrogen Supply

Stem can reallocate up to 100% of N that senesces each day if required by the plant arbitrator to meet N demands.

Stem can retranslocate up to 50% of non-structural N each day if required by the plant arbitrator to meet N demands.

### 1.8.7 Senescence and Detachment

The proportion of live biomass that senesces and moves into the dead pool each day is quantified by the SenescenceRate.

SenescenceRate is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

PostCanopyPhase has a value between EndCanopyDevelopment and Maturity calculated as:

$Rate = \text{Min}(Fraction, One)$

Where:

$Fraction = [Phenology].ThermalTime / TTRemaining$

$TTRemaining = PostCanopyTT - PostCanopyTTComplete$

PostCanopyTT = [Phenology].EarlyPodDevelopment.Target+[Phenology].EarlyGrainFilling.Target+[Phenology].MidGrainFilling.Target+[Phenology].LateGrainFilling.Target

PostCanopyTTComplete = [Phenology].MidGrainFilling.ProgressThroughPhase+[Phenology].LateGrainFilling.ProgressThroughPhase

One = 1

Stem has detachment parameterised to zero so all biomass in this organ will remain with the plant until a defoliation or harvest event occurs.

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	100	100
Cut	0	0	100	100
Prune	0	0	0	0
Graze	0	0	0	0

## 1.9 AboveGround

### 1.9.1 AboveGround

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

AboveGround summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Grain
- \* Shell

## 1.10 BelowGround

### 1.10.1 BelowGround

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

BelowGround summarises the following biomass objects:

- \* Root
- \* Nodule

## 1.11 AboveGroundLive

### 1.11.1 AboveGroundLive

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

AboveGroundLive summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Grain
- \* Shell

## 1.12 Total

### 1.12.1 Total

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

Total summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Grain
- \* Shell
- \* Root
- \* Nodule

## 1.13 TotalLive

### 1.13.1 TotalLive

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

TotalLive summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Grain
- \* Shell
- \* Root
- \* Nodule

## 1.14 TotalDead

### 1.14.1 TotalDead

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

TotalDead summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Grain
- \* Shell
- \* Root
- \* Nodule

## 1.15 Pod

### 1.15.1 Pod

This is a composite biomass class, representing the sum of 1 or more biomass objects from one or more organs.

Pod summarises the following biomass objects:

- \* Shell
- \* Grain

## 1.16 AflotoxinRisk

AflotoxinRisk is the value of AccumulatedValue bound between a lower and upper bound where:

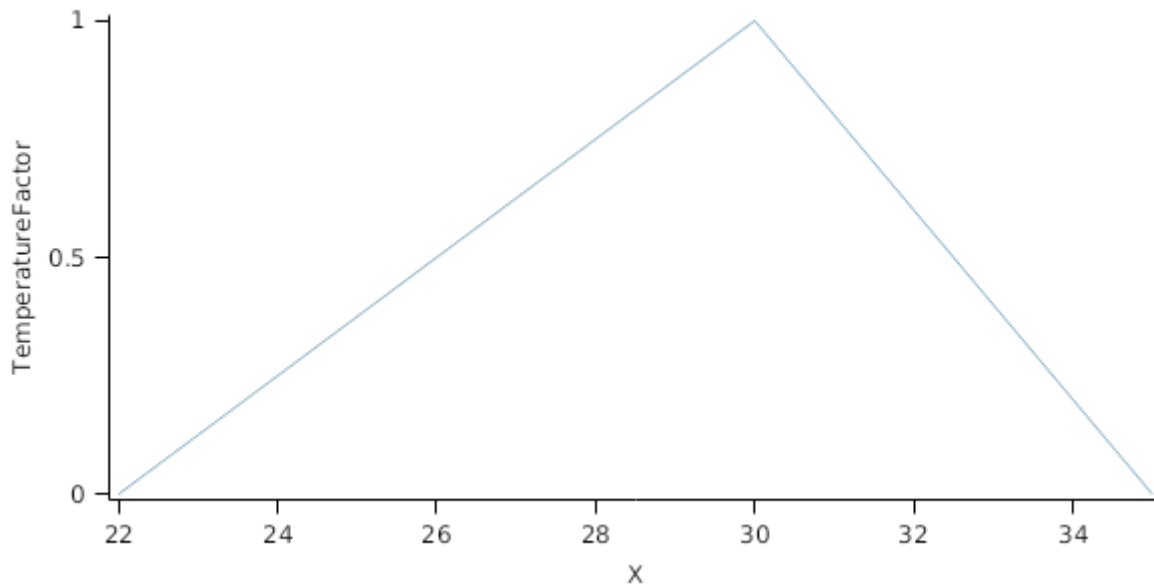
*AccumulatedValue* = Accumulated *DailyIncrease* between endpoddevelopment and endgrainfill

*DailyIncrease* = *TemperatureFactor* x *SoilWaterFactor* x 3

*TemperatureFactor* is calculated using linear interpolation

X	TemperatureFactor
22.0	0.0
30.0	1.0
35.0	0.0

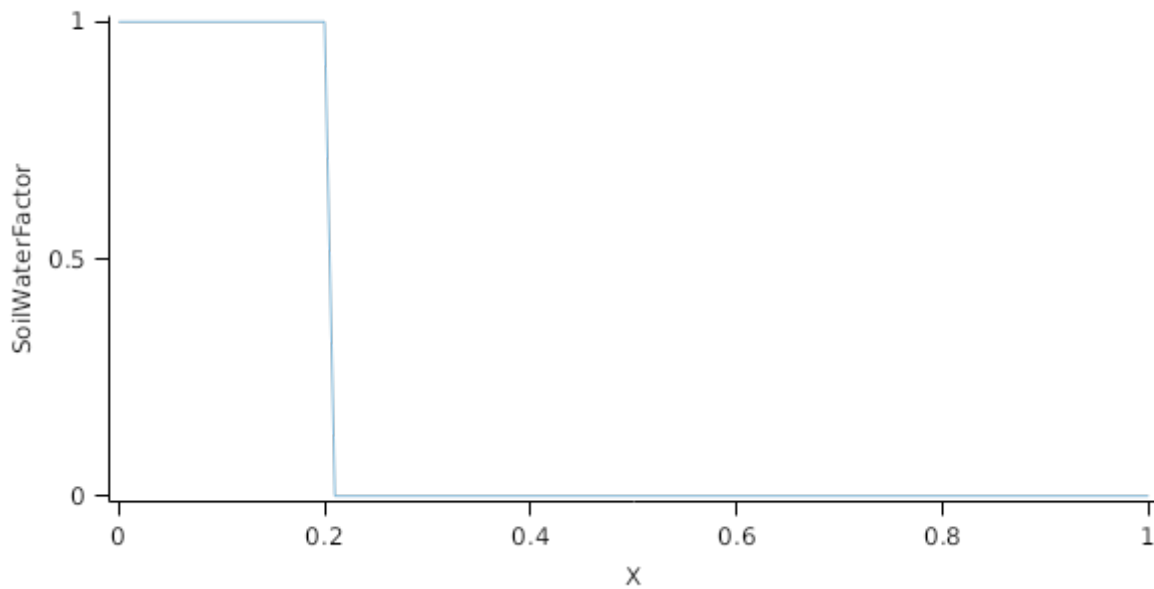
## TemperatureFactor



*SoilWaterFactor* is calculated using linear interpolation

X	SoilWaterFactor
0.0	1.0
0.2	1.0
0.2	0.0
1.0	0.0

## SoilWaterFactor



Lower = 0

Upper = 100

### 1.17 Cultivars

#### 1.17.1 EarlyBunch

EarlyBunch overrides the following properties:



[Phenology].Vegetative.Target.FixedValue = 420

[Phenology].EarlyFlowering.Target.FixedValue = 160

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1050

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.6

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.4

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,10,11

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2.0,2.0,0.4

[Leaf].ExtinctionCoefficient.XYPairs.X = 0, 335

[Leaf].ExtinctionCoefficient.XYPairs.Y = 0.5, 0.5

### **1.17.2 VirginiaBunch**

VirginiaBunch overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 450

[Phenology].EarlyFlowering.Target.FixedValue = 160

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1020

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.40

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.33

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,1.0,1.0,0

### **1.17.3 TMV2**

TMV2 overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 390

[Phenology].EarlyFlowering.Target.FixedValue = 160

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1020

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.40

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.33

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,1.0,1.0,0

### **1.17.4 Tapir**

Tapir overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 390

[Phenology].EarlyFlowering.Target.FixedValue = 160

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1020

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.40

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.33

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,1.0,1.0,0

### **1.17.5 VB97**

VB97 overrides the following properties:

[Phenology].Emerging.ShootLag=50

[Phenology].Vegetative.Target.FixedValue = 390

[Phenology].EarlyFlowering.Target.FixedValue = 100

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.05

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1020

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.33

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2,2,0

[Leaf].ExtinctionCoefficient.XYPairs.X = 5, 335

[Leaf].ExtinctionCoefficient.XYPairs.Y = 0.2, 0.55

### **1.17.6 Florunner**

Florunner overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 400

[Phenology].EarlyFlowering.Target.FixedValue = 160

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.7

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1020

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.20

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.33

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2,2,0

[Leaf].ExtinctionCoefficient.XYPairs.X = 0, 335

[Leaf].ExtinctionCoefficient.XYPairs.Y = 0.6, 0.6

[Leaf].SpecificArea.XYPairs.X = 0, 17, 21, 35

[Leaf].SpecificArea.XYPairs.Y = .02, .02, .02, .02

[Leaf].SenescenceRate.Reproductive.Rate.Modifier.FixedValue = 100

### **1.17.7 Redvale**

Redvale overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 420

[Phenology].EarlyFlowering.Target.FixedValue = 200

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 950

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.5

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.2

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2.0,1.0,0

### **1.17.8 Taabinga**

Taabinga overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 420

[Phenology].EarlyFlowering.Target.FixedValue = 200

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 950

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.5

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.20

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2.0,1.0,0

### **1.17.9 Holt**

Holt overrides the following properties:

[Phenology].Vegetative.Target.FixedValue = 500

[Phenology].EarlyFlowering.Target.FixedValue = 250

[Phenology].LateGrainFilling.Target.EntireGrainfillPeriod.FixedValue = 1100

[Phenology].EarlyGrainFilling.Target.FractionofGrainfilling.FixedValue = 0.5

[Phenology].MidGrainFilling.Target.FractionofMidToLateGrainfilling.FixedValue=0.2

[Grain].MaximumNConc.FixedValue = 0.067

[Stem].PotentialGrowth.PartitionFraction.StemGrowthPhase.StemFraction.FixedValue = 0.5

[Leaf].BranchingRate.XYPairs.X = 0,1,2,3,15,16

[Leaf].BranchingRate.XYPairs.Y = 0,1,2,2.0,1.0,0

### **1.18 MortalityRate**

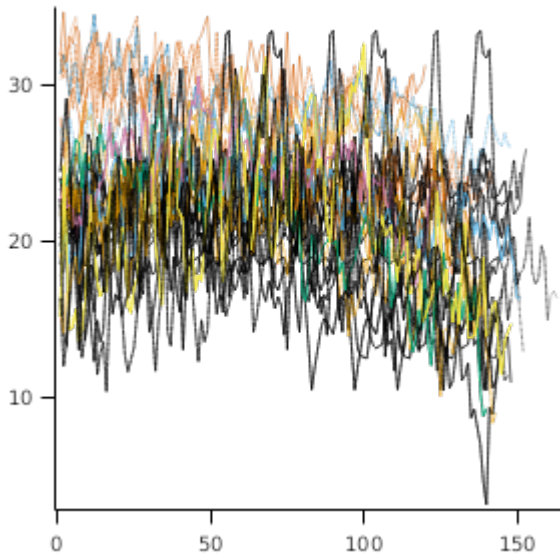
MortalityRate = 0

## 2 Validation

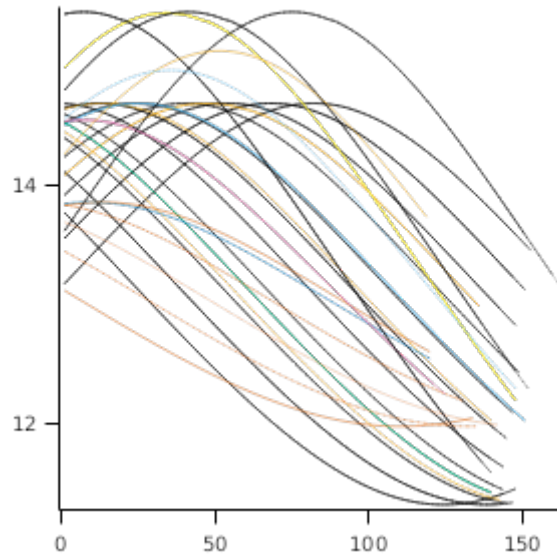
The Peanut model has been tested across a range of planting and agronomic conditions. These include datasets covering a wide range of weather conditions (See graphs that follow) and agronomic factors (irrigation, fertiliser, sowing dates).



**Growing Season Mean Air Temperature**



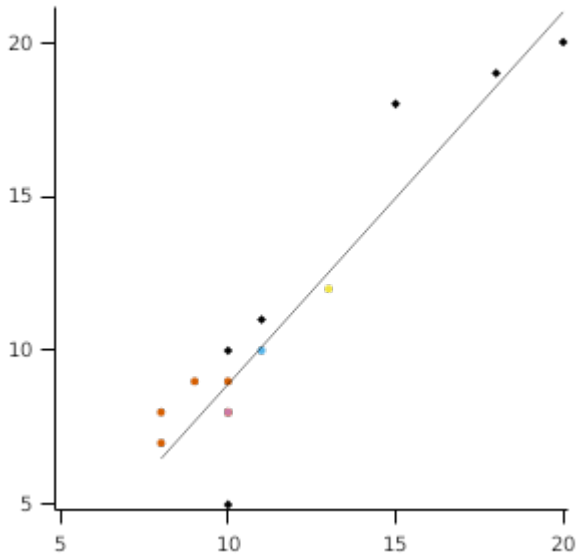
**Growing Season Photoperiod**



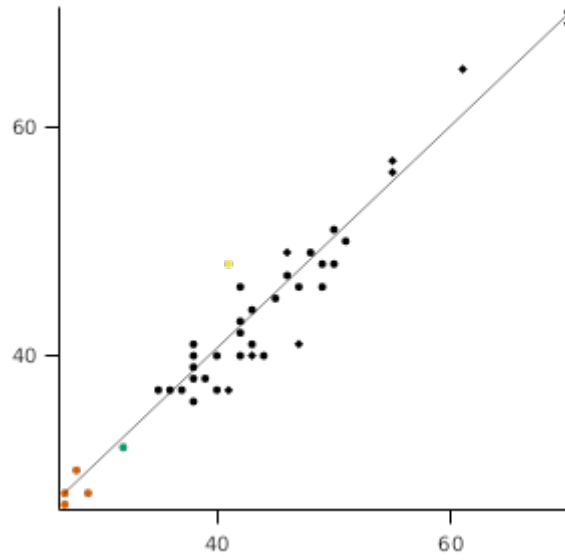
## 2.1 Combined Results

Simulation results for the combined datasets from the various countries are shown in the following graphs. The model is able to adequately capture the influence of growing conditions (soil, climate) and management (population, Nitrogen, irrigation, sowing date).

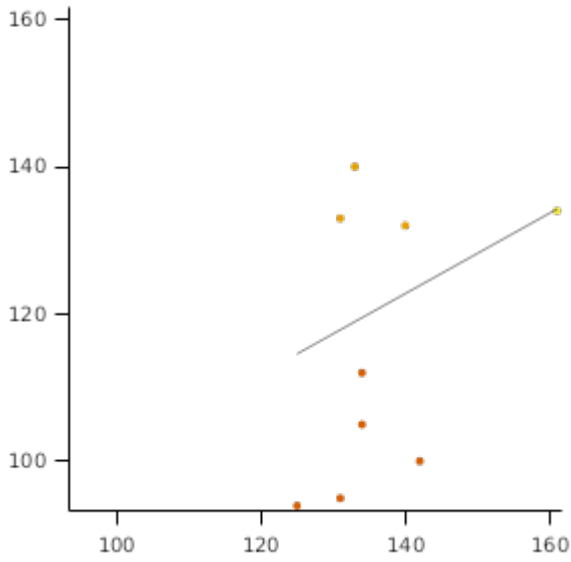
**EmergenceDAS**



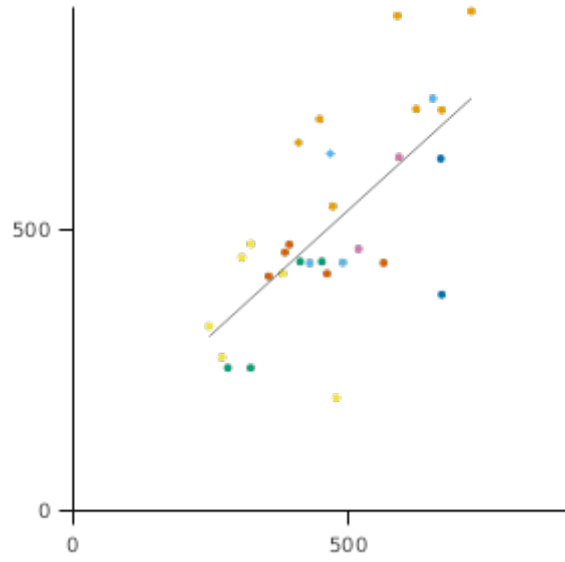
**FloweringDAS**



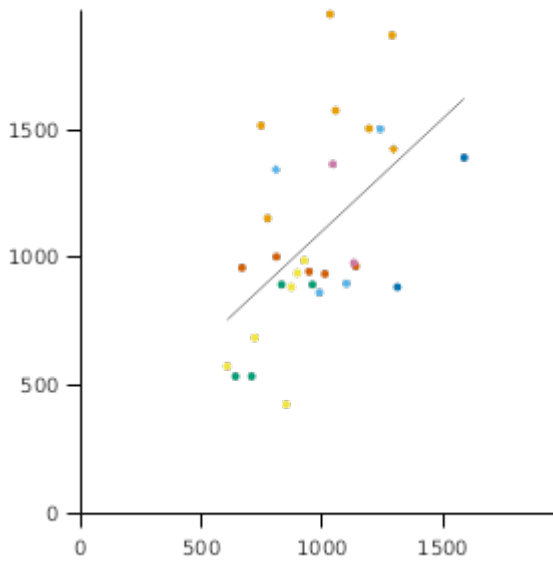
**MaturityDAS**



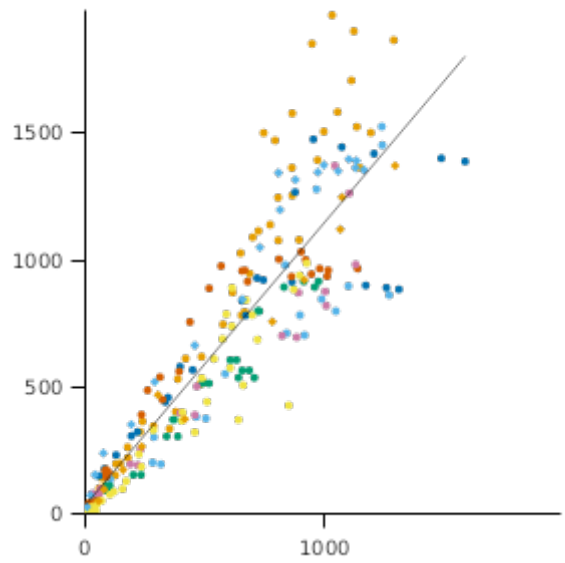
**Harvest Yield**

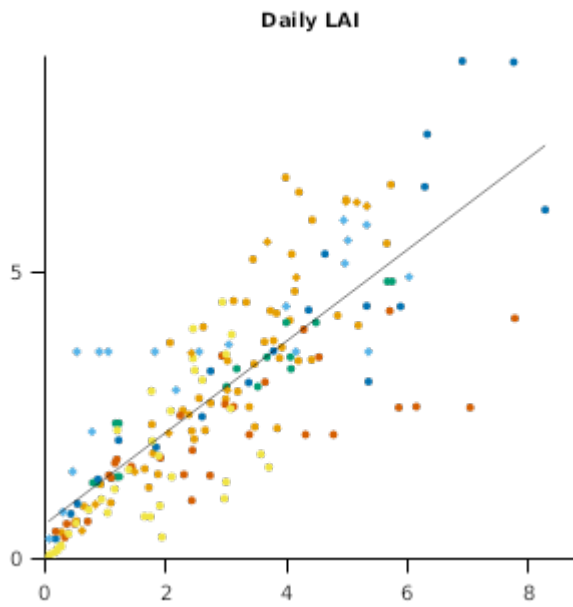


**HarvestBiomass**



**Daily Biomass**





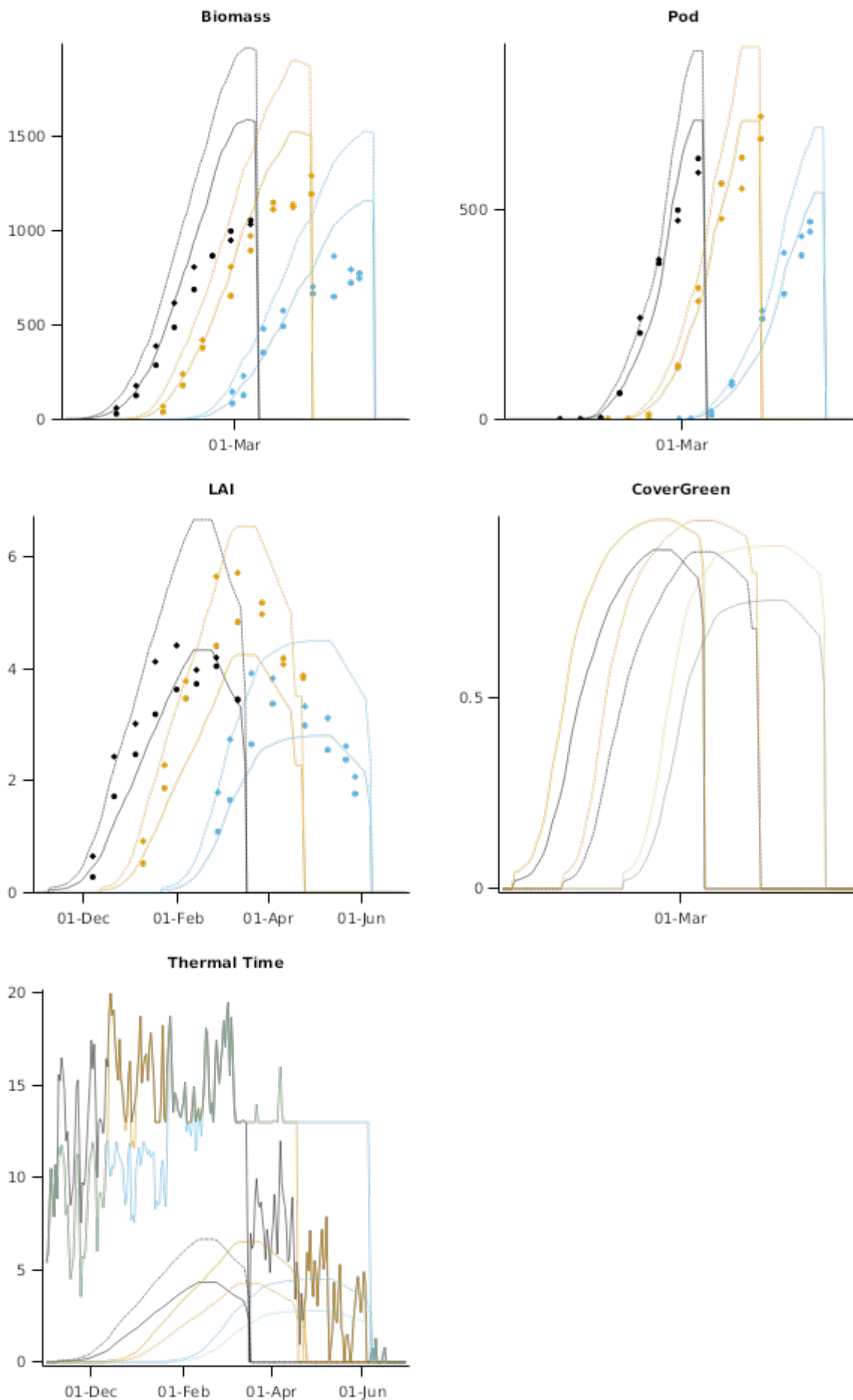
## 2.2 Australia

### List of experiments.

Experiment Name	Design (Number of Treatments)
Kingaroy8384	Sow x Cv (33)
Kingaroy8485	Sow x Pop (6)
Kingaroy8990	Cv (3)
Kingaroy1819	Cv x Irrigation (4)
Kunnunura8182	Pop (2)
Kunnunura8283	Sow (5)
Bundaberg8990	Cv (4)
Loxton9697	Pop (6)
Roseworthy96	Sow x Cv (9)

### 2.2.1 Kingaroy8485

#### 2.2.1.1 TimeSeries

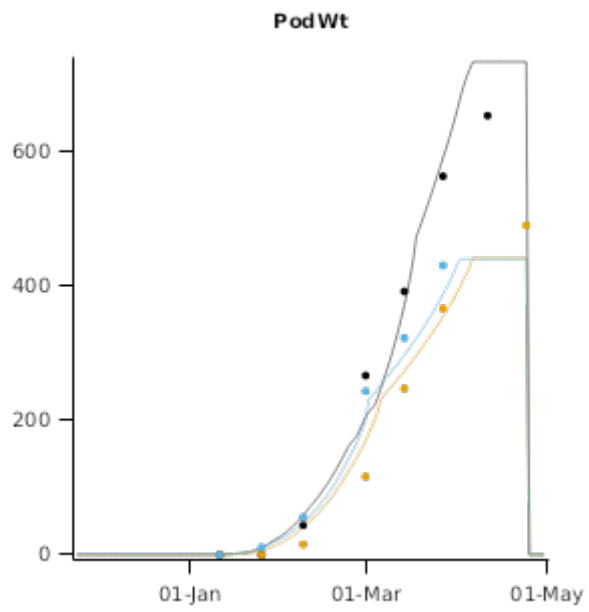
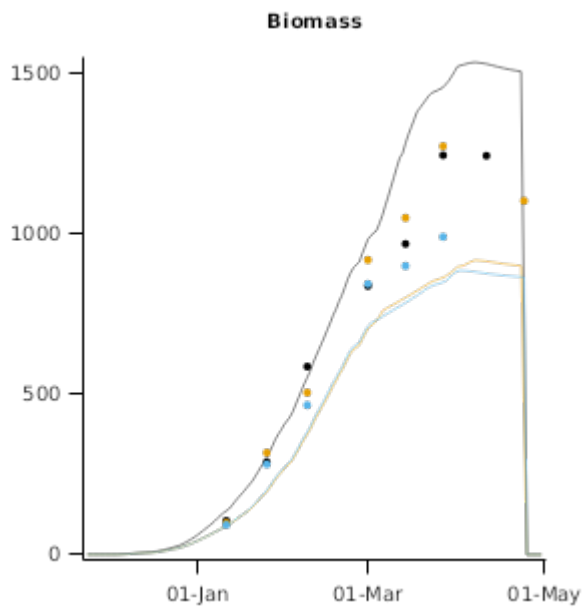


## 2.2.2 Kingaroy8990

This dataset (Bell, 1992) provides data on growth for three cultivars grown under irrigated conditions in at Kingaroy in Queensland. The soil type was a deep, red clay-loam with a friable structure and increasing clay content with depth. Supplementary irrigations were made to avoid moisture stress.

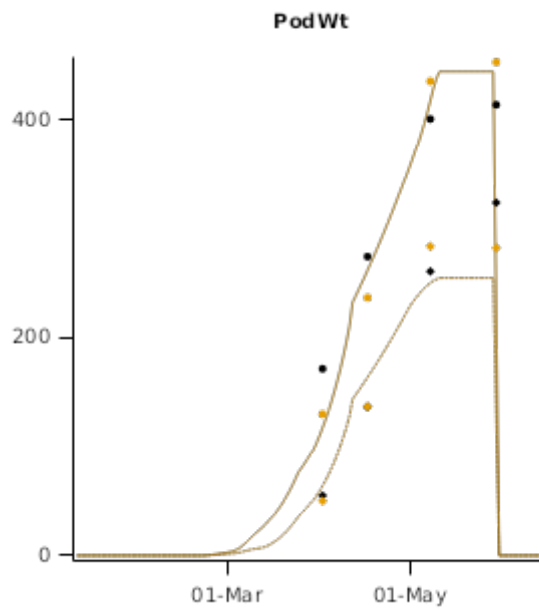
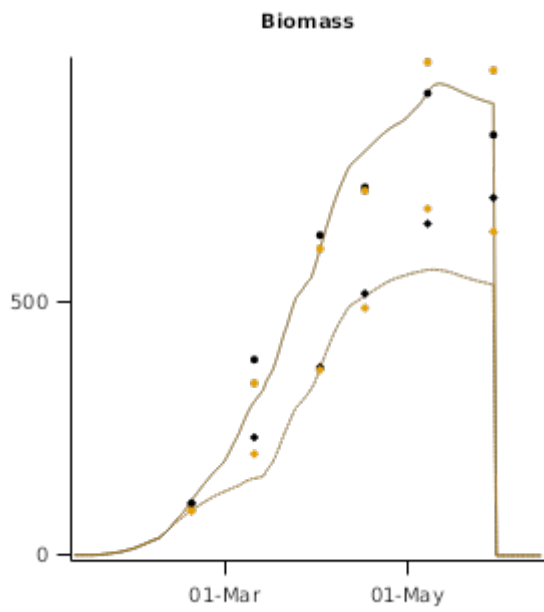


### 2.2.2.1 Biomass

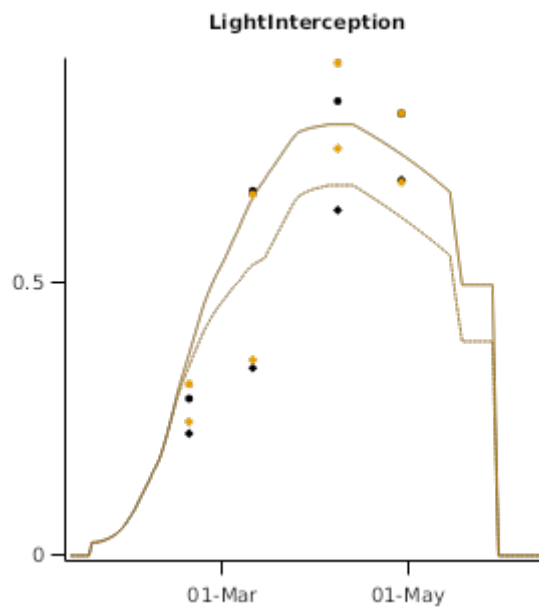
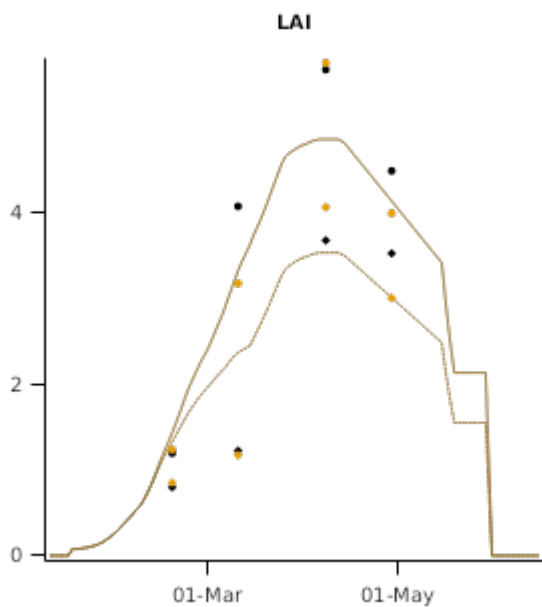


### 2.2.3 Kingaroy1819

#### 2.2.3.1 Biomass

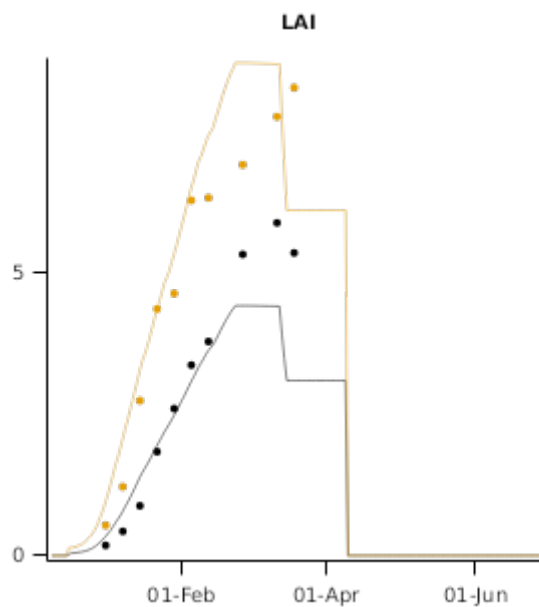
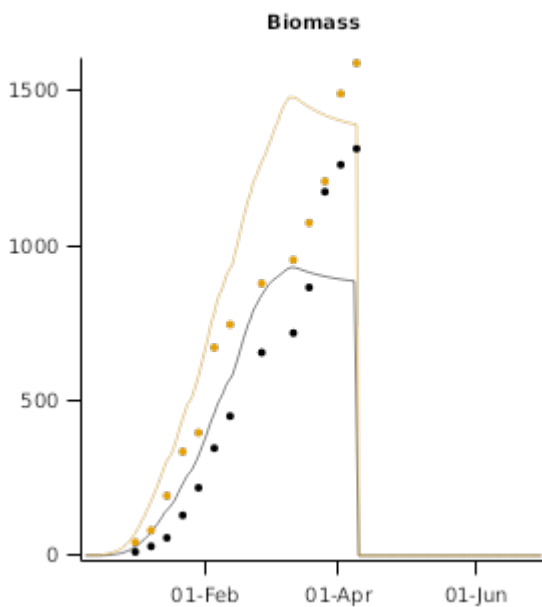


#### 2.2.3.2 Canopy



## 2.2.4 Kunnunura8182

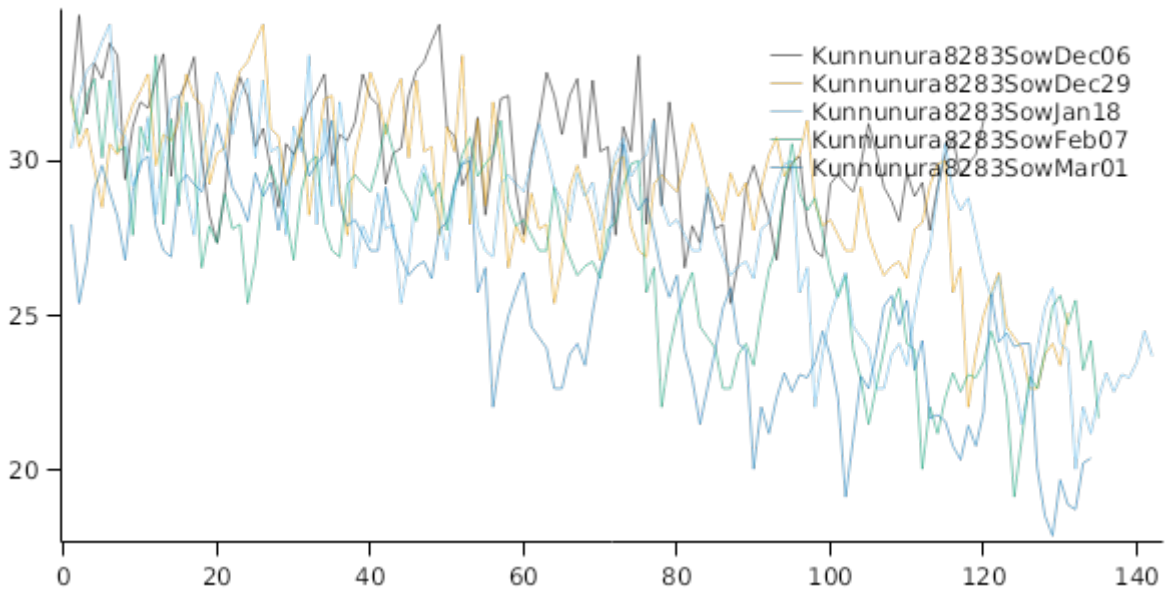
### 2.2.4.1 TimeSeries



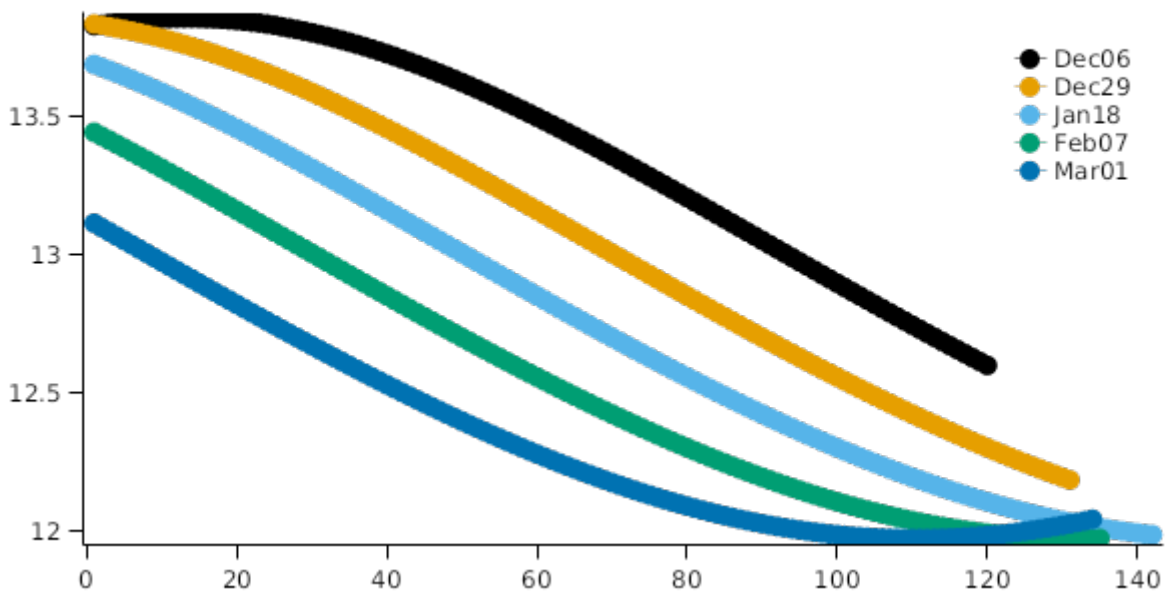
## 2.2.5 Kunnunura8283

This dataset (Bell, 1986) includes the cultivar "EarlyBunch" grown on 5 sowing dates under irrigated conditions in a monsoonal tropical environment at Kunnunura in Northern Australia. Plants were sampled regularly for biomass partitioning and canopy development. The soil is known locally as a normal phase Cockatoo Sand - an acid, coarse-textured and well-drained soil. Supplementary irrigations were made to avoid moisture stress.

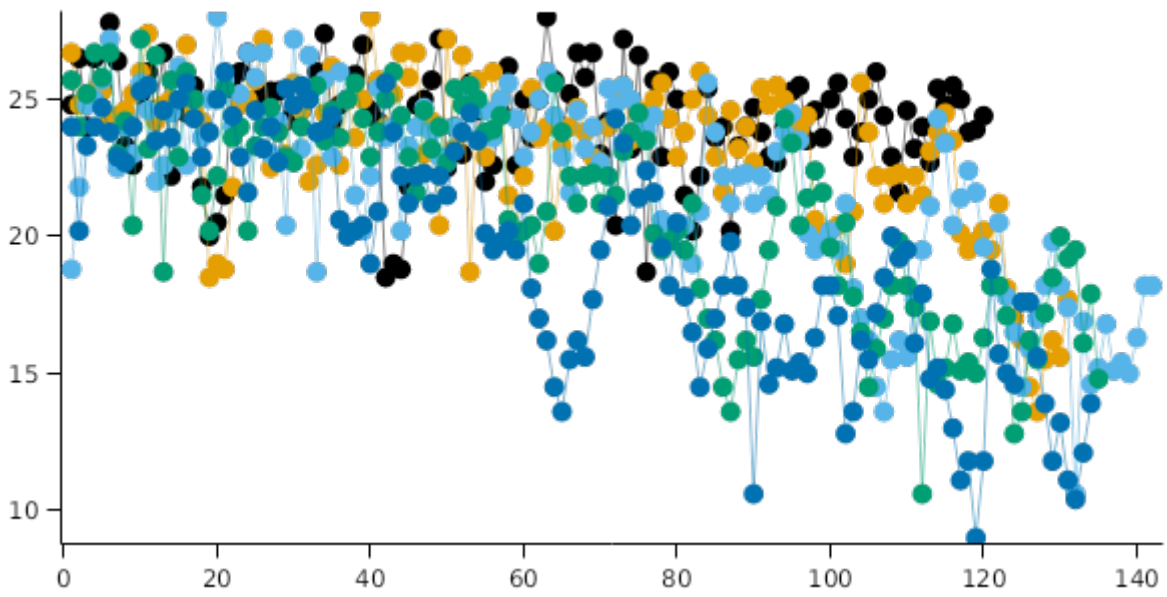
## Growing Season Mean Air Temperature



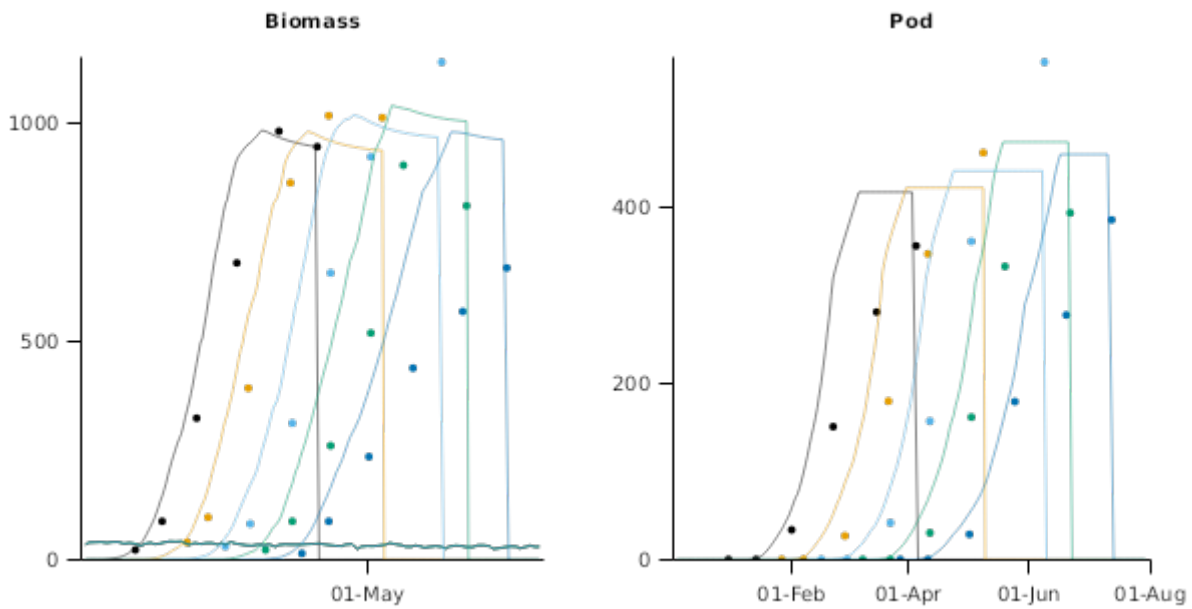
## Growing Season Photoperiod

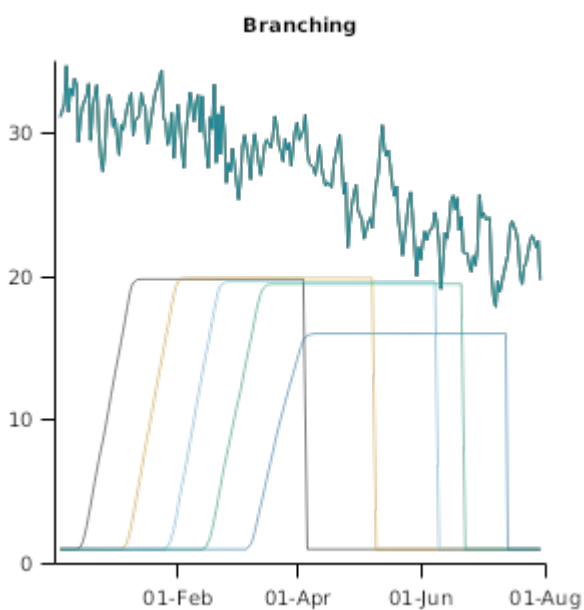
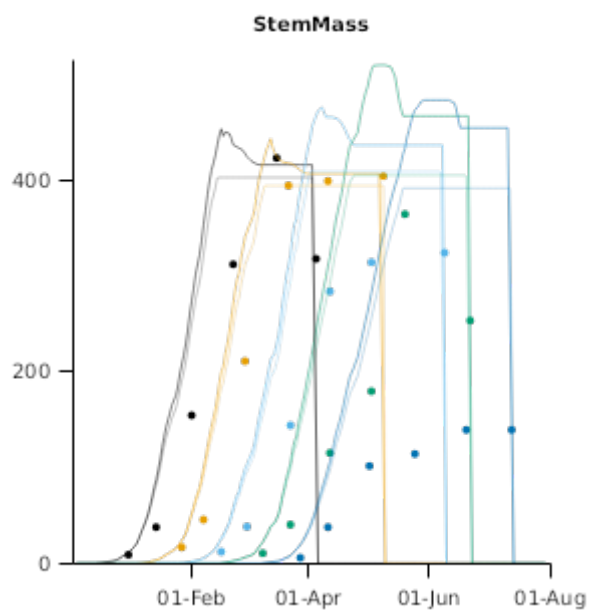
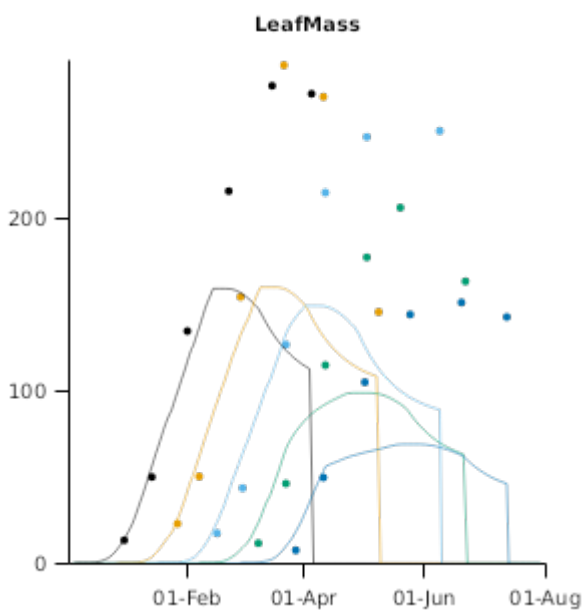
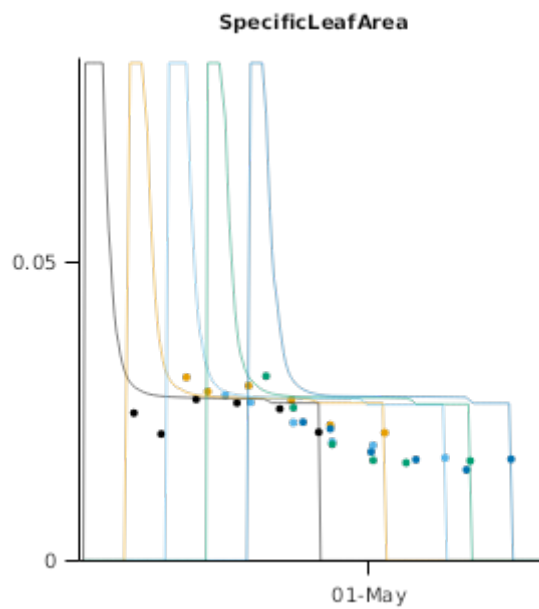
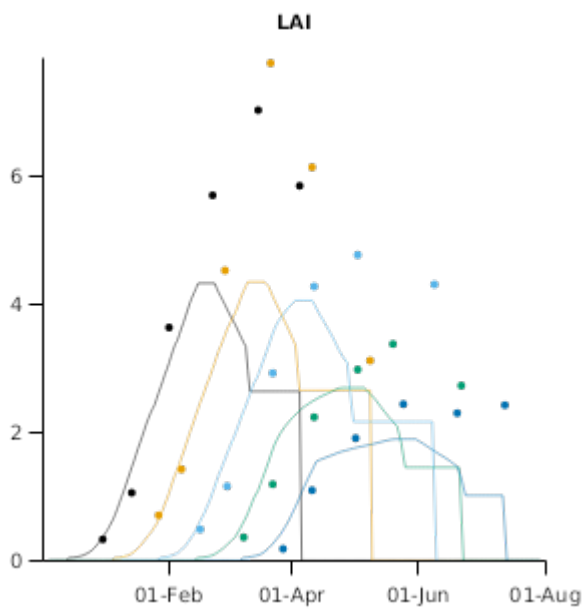


## Growing Season Min Air Temperature



### 2.2.5.1 TimeSeries



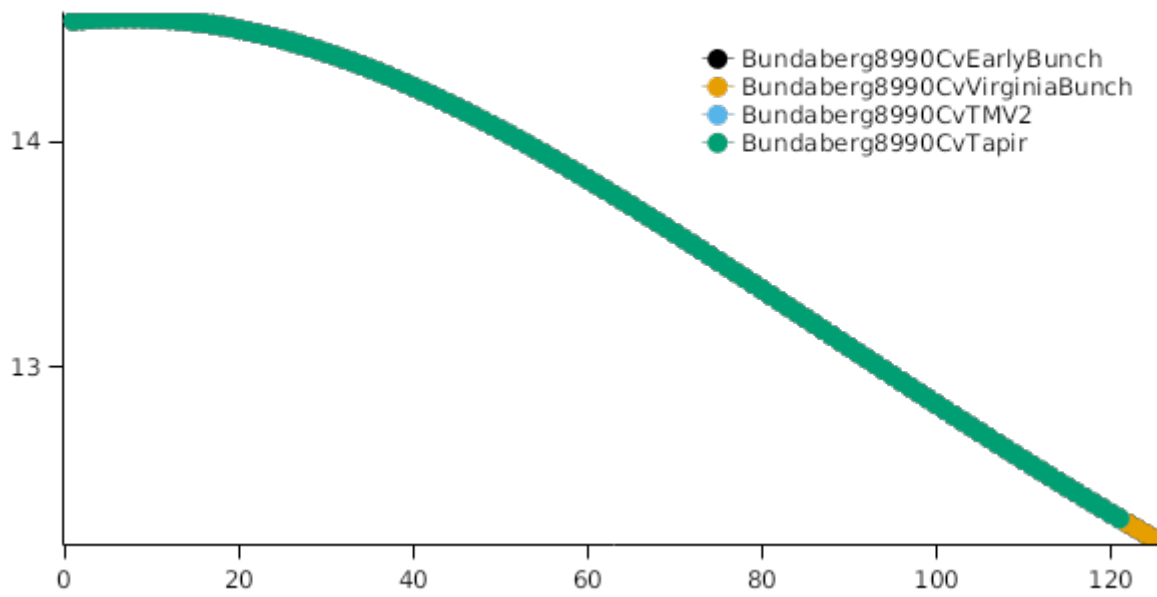


## 2.2.6 Bundaberg8990

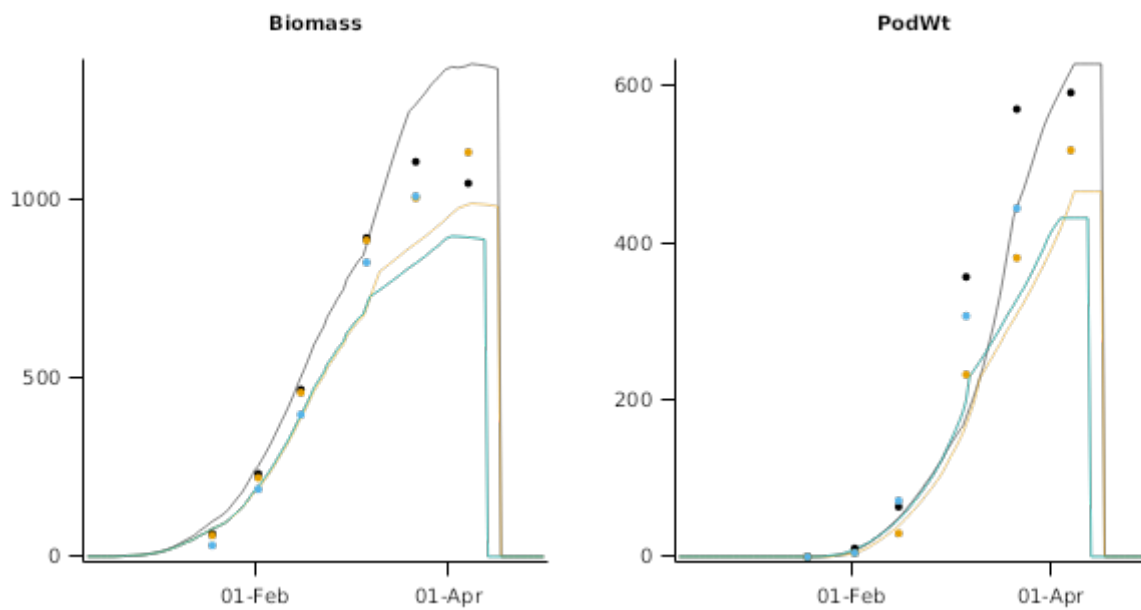
This dataset (Bell, 1994) studied N fixation of various Peanut cultivars at Bundaberg, Queensland, Australia. The soil type was a deep, red clay-loam with a friable structure and increasing clay content with depth. The experiment was grown

under irrigated conditions. Measurements were undertaken for biomass and nitrogen contents, and estimates of the contribution of N fixation to plant N contents.

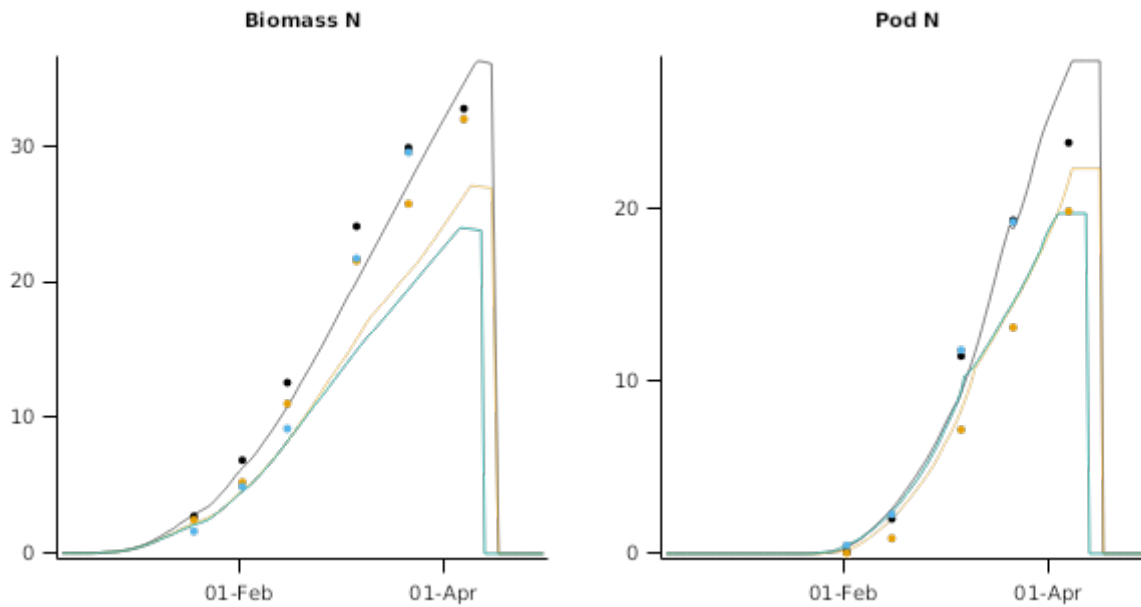
## Growing Season Photoperiod



### 2.2.6.1 Biomass



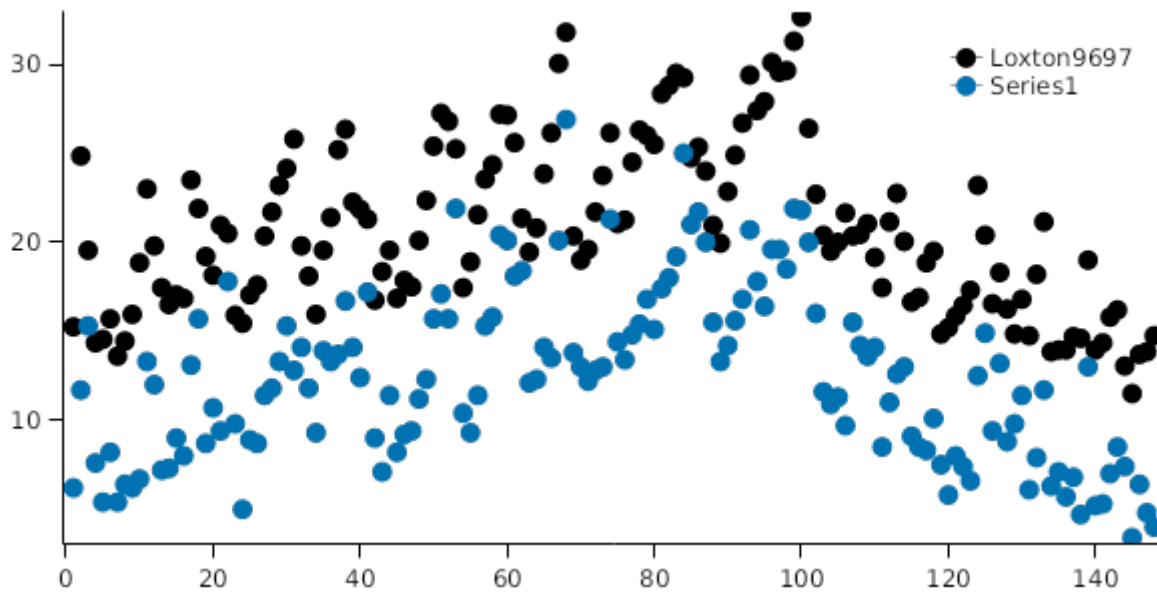
### 2.2.6.2 Nitrogen



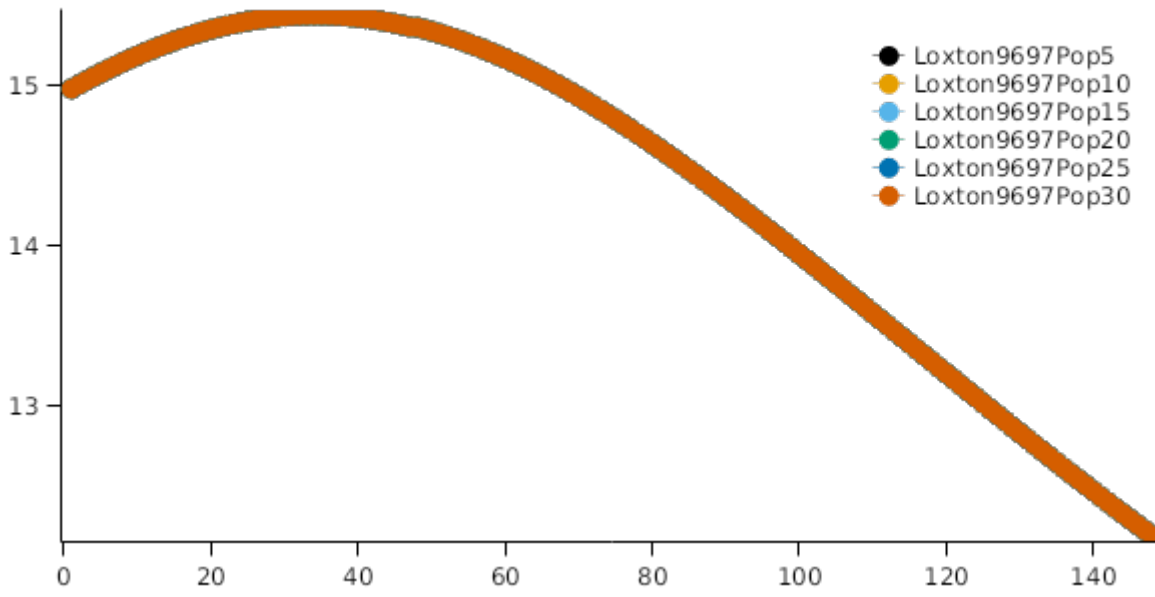
### 2.2.7 Loxton9697

This dataset (Porter, 2000) studied the impact of plant population on growth and productivity. Here we consider the cultivar EarlyBunch only (the comparison cultivar failed during the trial). The experiment was grown under irrigated conditions in at Loxton in South Australia. The soil type was sandy loam with good drainage but low water holding capacity. Supplementary irrigations were made to avoid moisture stress.

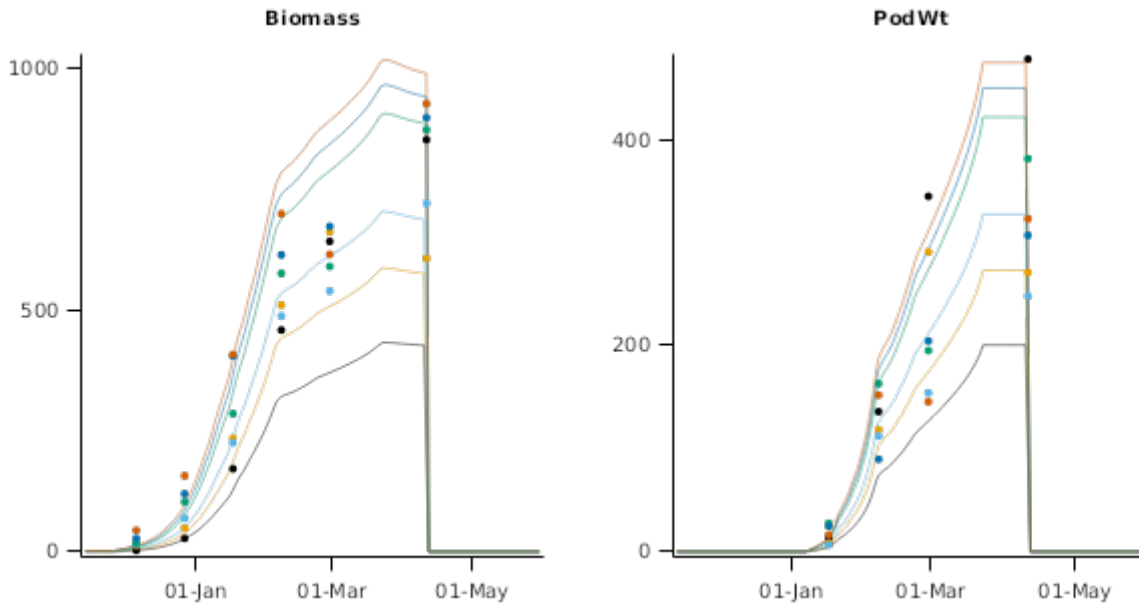
### Growing Season Mean Air Temperature



## Growing Season Photoperiod

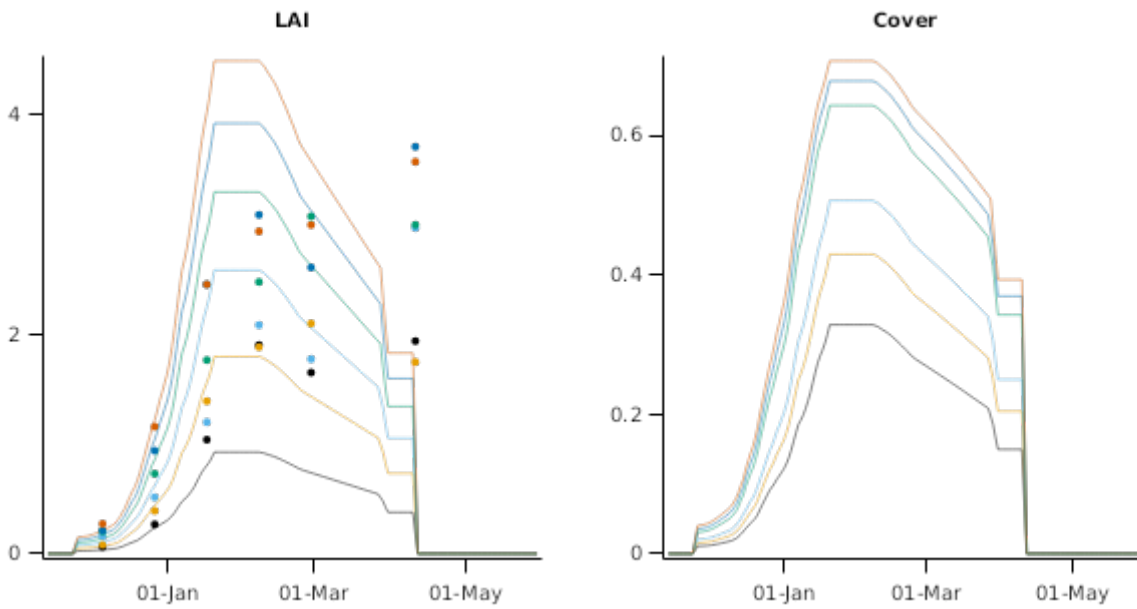


### 2.2.7.1 Biomass



### 2.2.7.2 Canopy

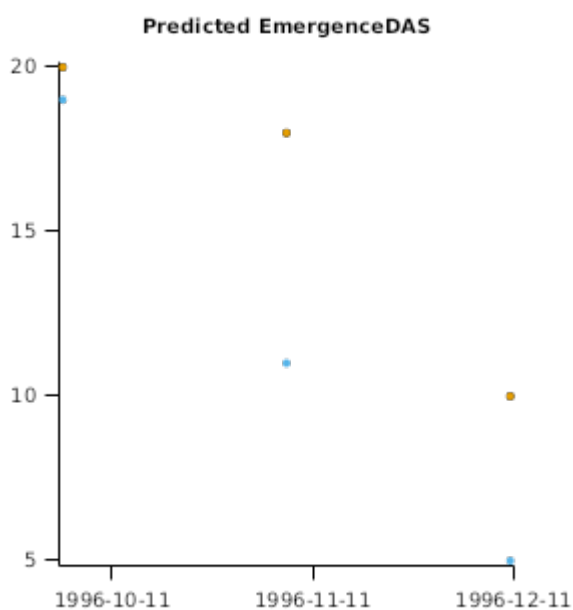
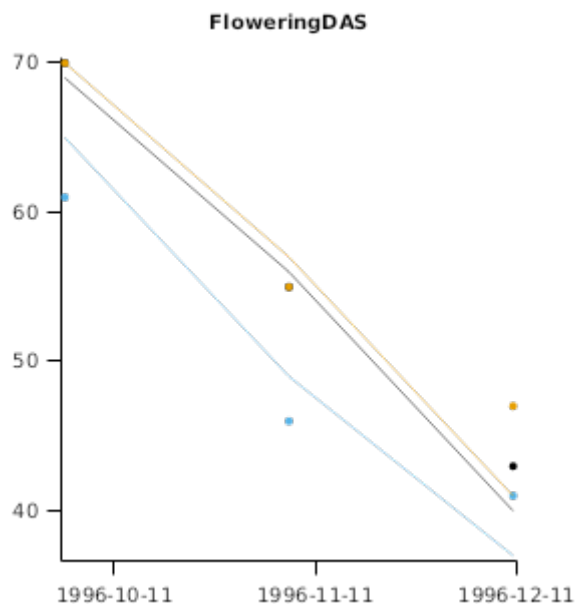
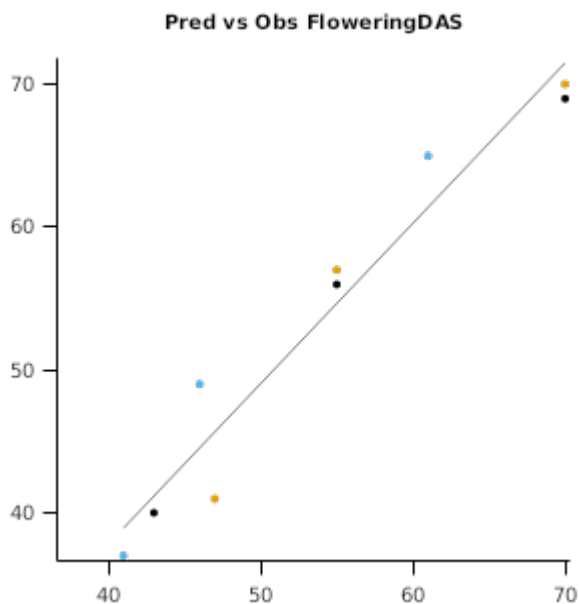




## 2.2.8 Roseworthy96

This dataset ([Porter, 2000](#)) studied the impact of sowing date on flowering and maturity for 24 germplasm lines. Here we consider the cultivars EarlyBunch, Florunner, and VB97. The experiment was grown under irrigated conditions in at Roseworthy in South Australia. The soil type was clay-loam with good water holding capacity. well-drained soil. Supplementary irrigations were made to avoid moisture stress.

### 2.2.8.1 Phenology



## 2.3 USA

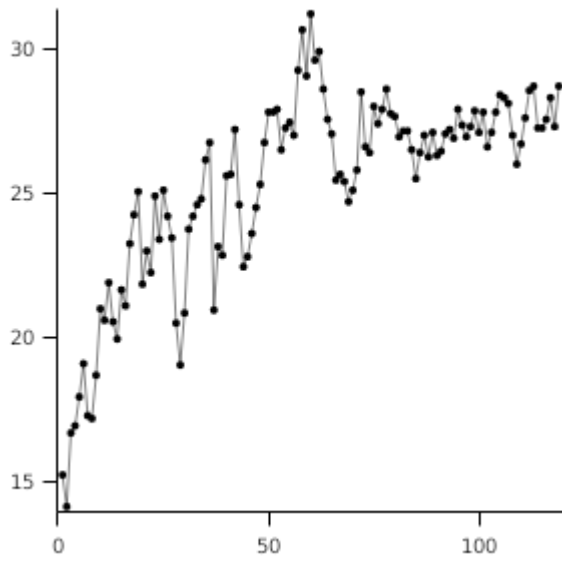
### 2.3.1 Headland Florunner

This dataset (Davis, 1991) studied the differences in canopy development for florunner cultivars. The experiment was grown under irrigated conditions at the Wiregrass Substation in Headland, Alabama. Supplementary irrigations were made to avoid moisture stress. Plants were sampled for plant vegetative stage, reproductive stage, height, number of leaves, leaf area, leaf dry weight, number of pods, pod dry weight, stem dry weight, and stand density.

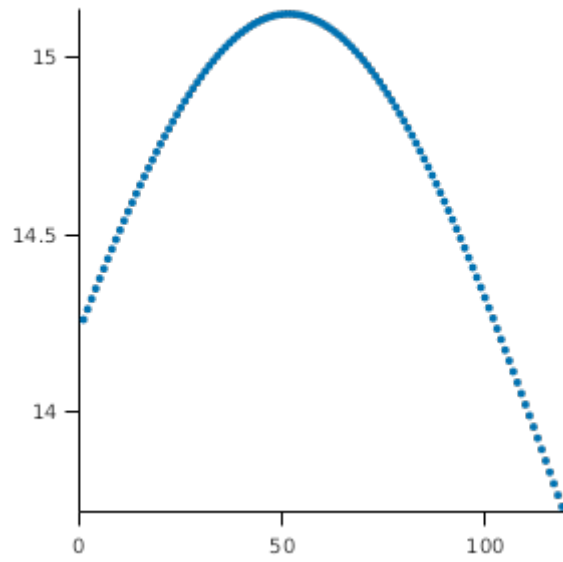
#### List of experiments.

Experiment Name	Design (Number of Treatments)
Headland88	Cv (1)

Growing Season Mean Air Temperature

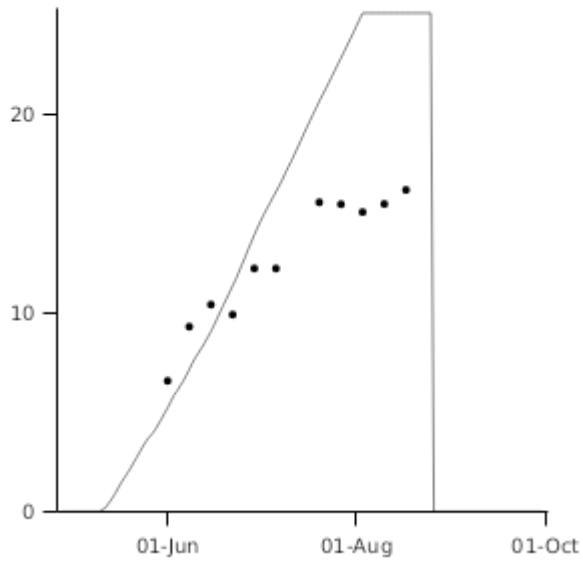


Growing Season Photoperiod

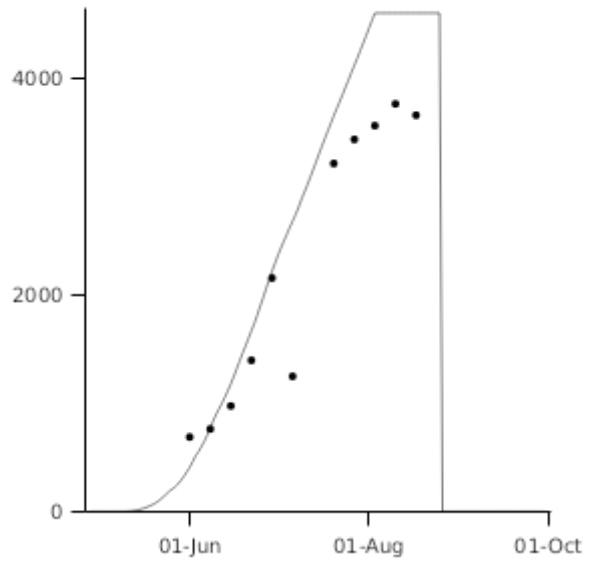


### 2.3.1.1 TimeSeries

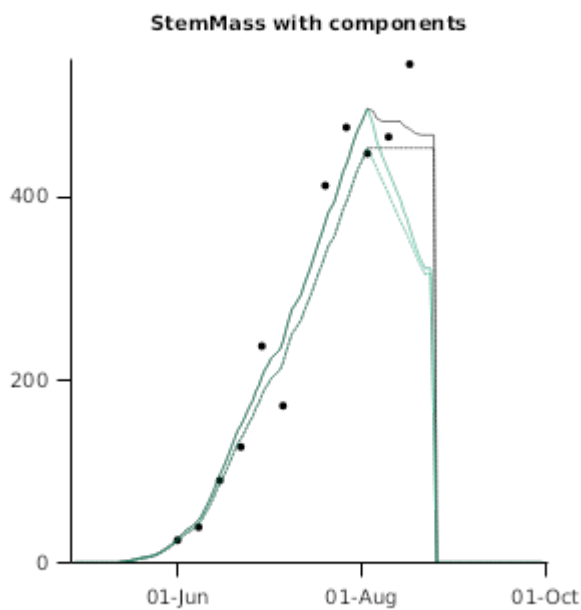
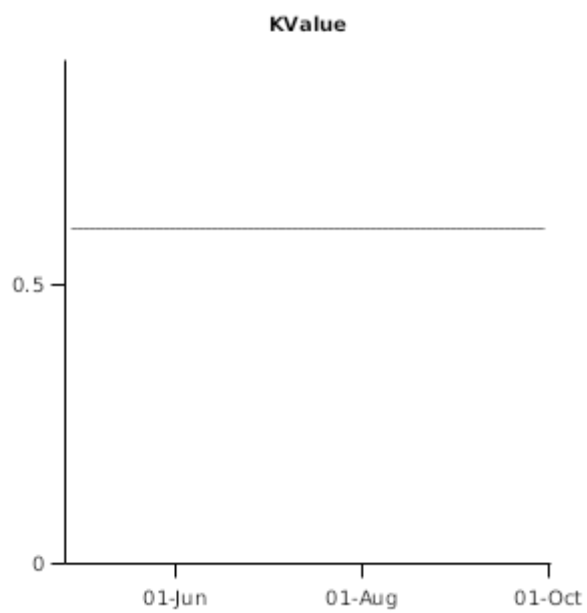
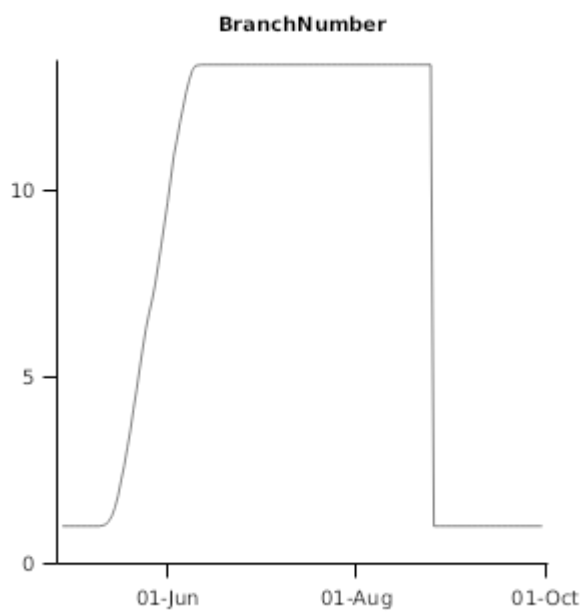
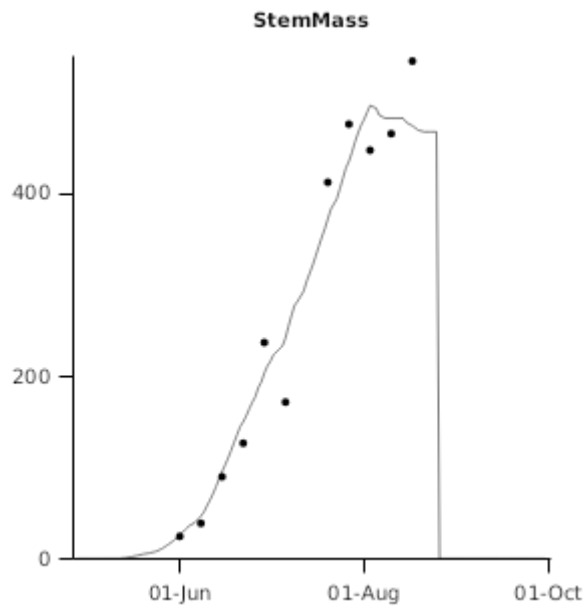
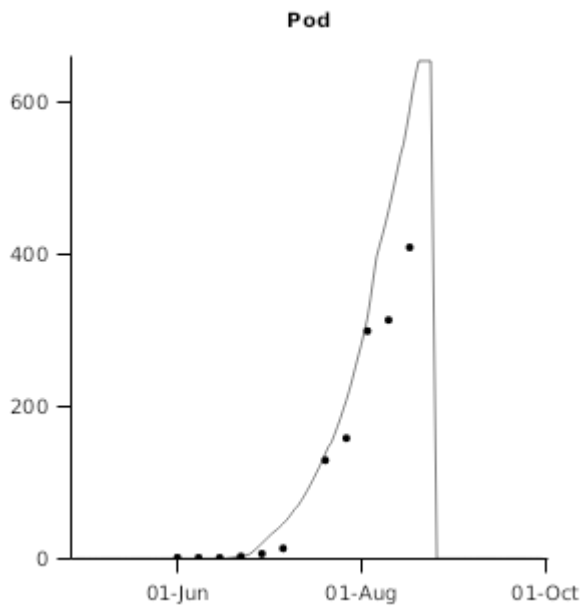
NodeNumber



LeafNumber





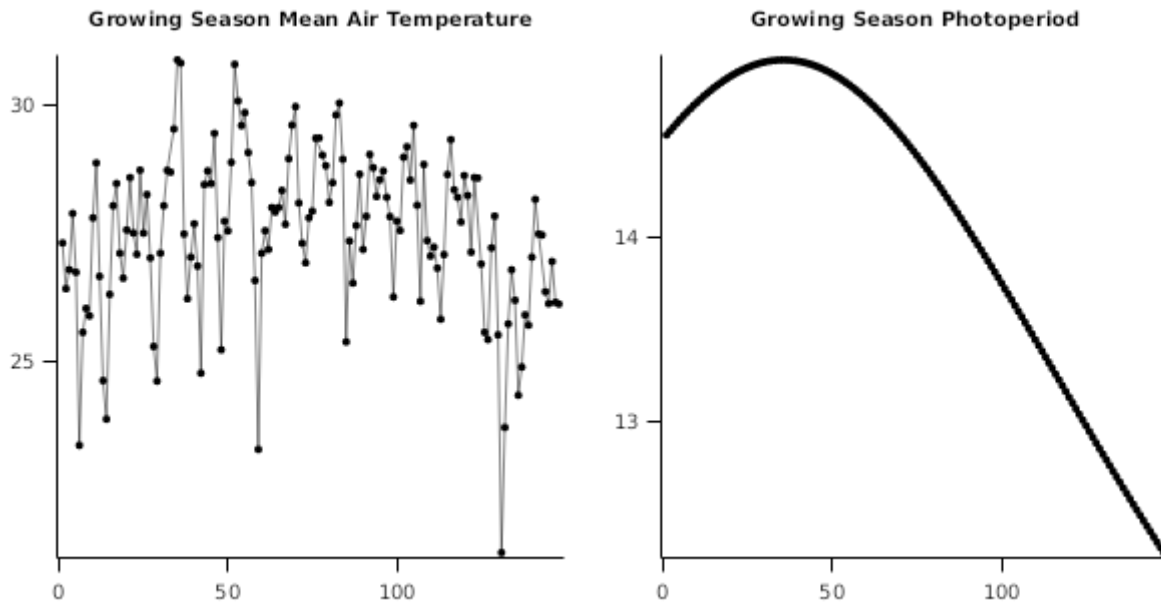


### 2.3.2 Gainesville EarlyBunch

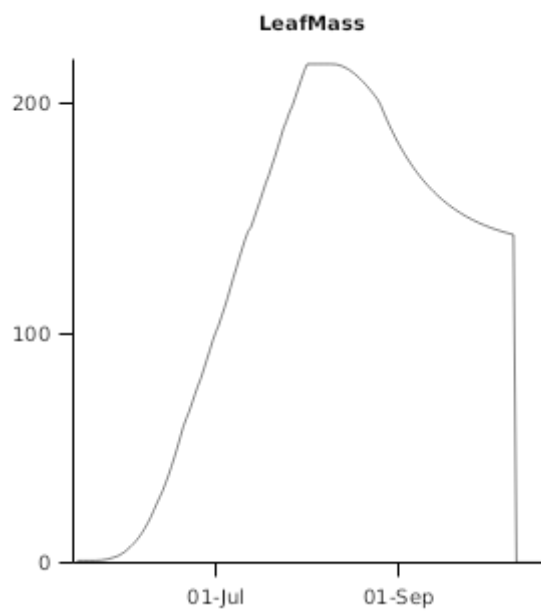
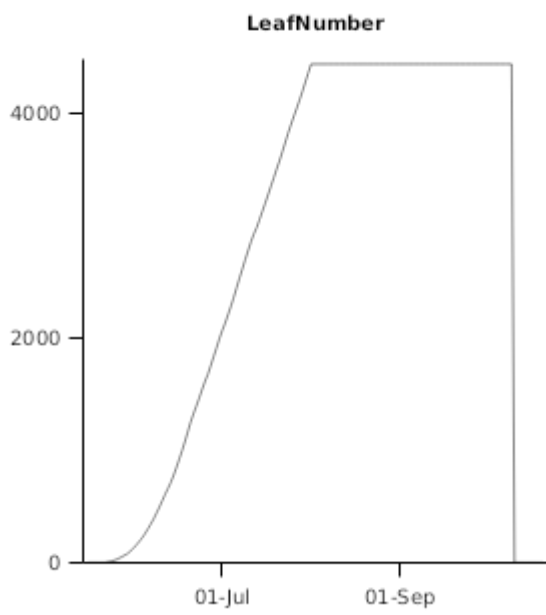
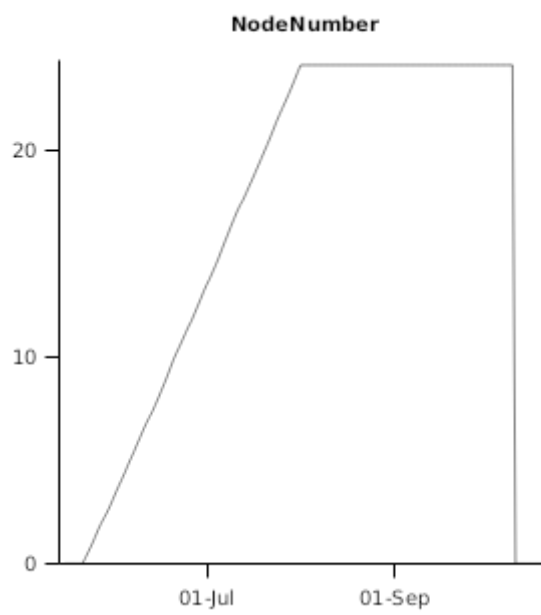
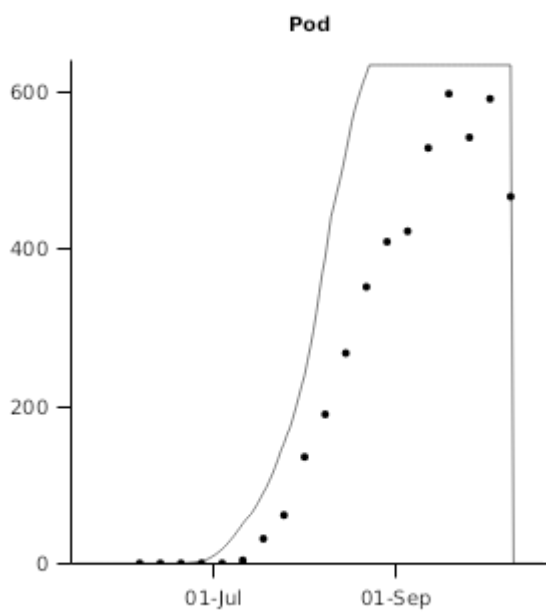
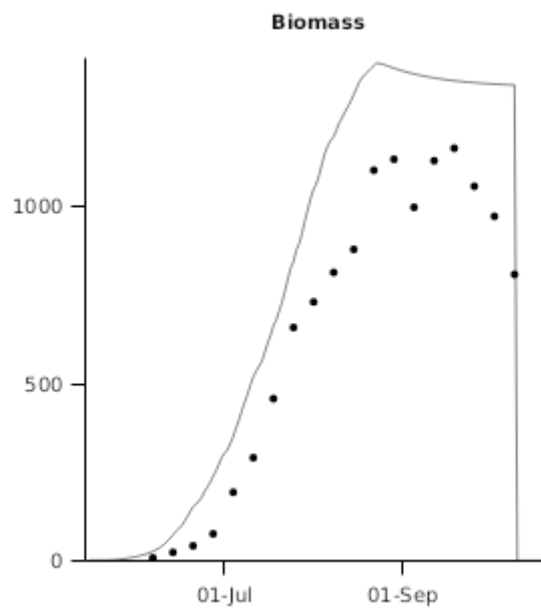
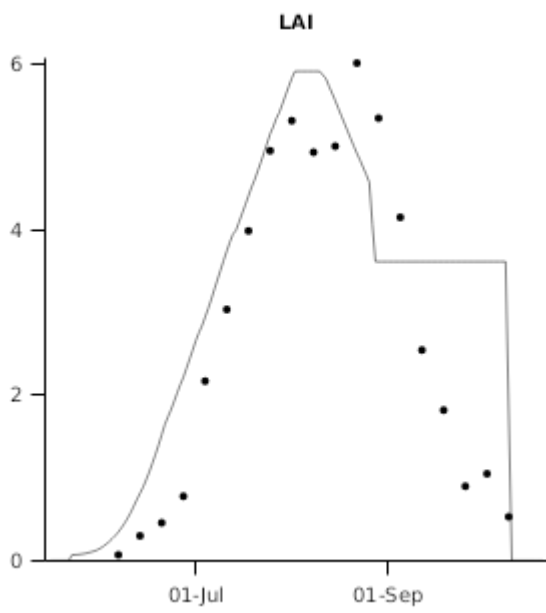
This dataset (from Hammer\_etal\_QNut) studied the differences in growth for two cultivars. We include only the cultivar "EarlyBunch". The experiment was grown under irrigated conditions at Gainesville, Florida. Supplementary irrigations were made to avoid moisture stress. Plants were sampled for total biomass, pod biomass and leaf area index.

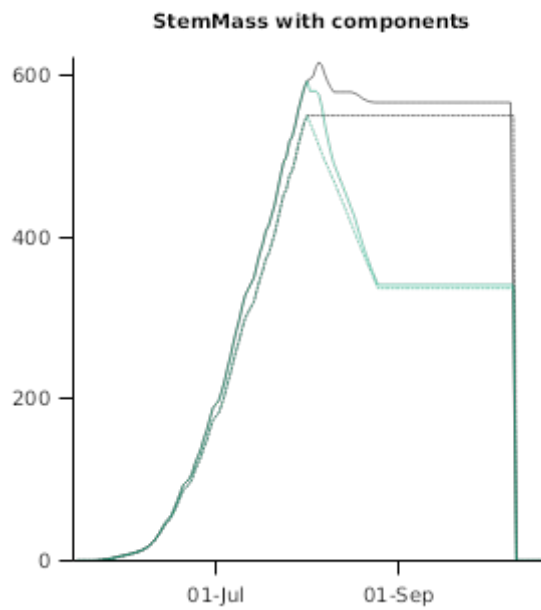
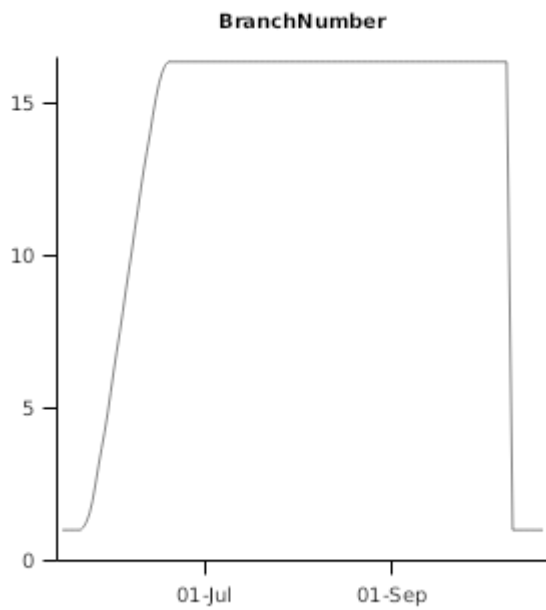
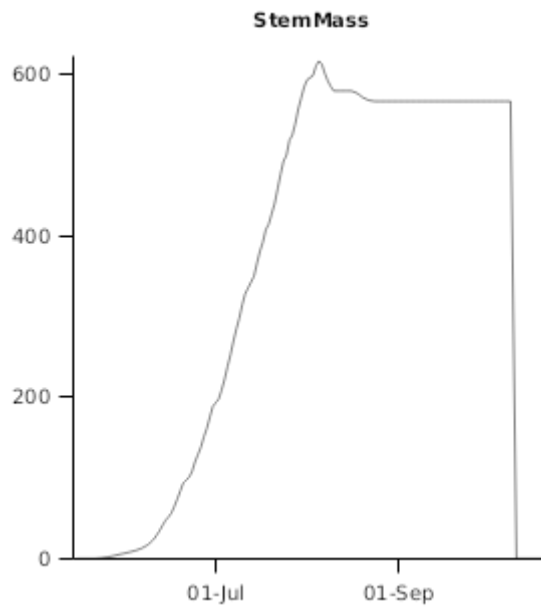
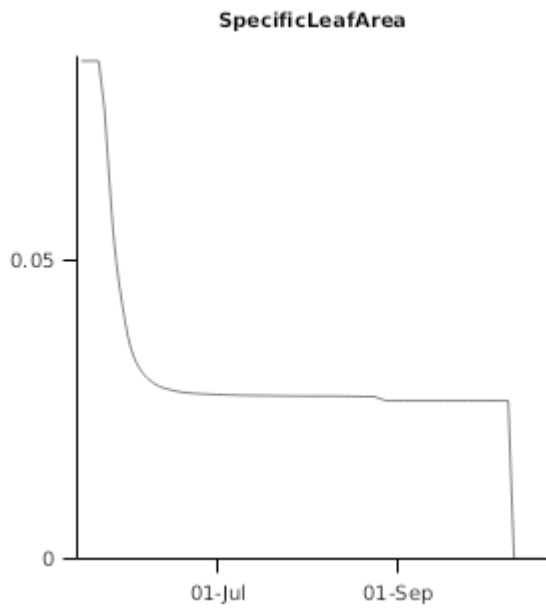
### List of experiments.

Experiment Name	Design (Number of Treatments)
Gainsville90	Cv (1)



### 2.3.2.1 TimeSeries





### 3 Sensibility

#### 3.1 DetailedDynamics

List of experiments.

Experiment Name	Design (Number of Treatments)
DetailedDynamics	Sow (1)

#### 3.2 Sugar Systems

List of experiments.

Experiment Name	Design (Number of Treatments)
SugarSystems	Rotation (0)

#### 3.3 Aflotoxin Risk

List of experiments.

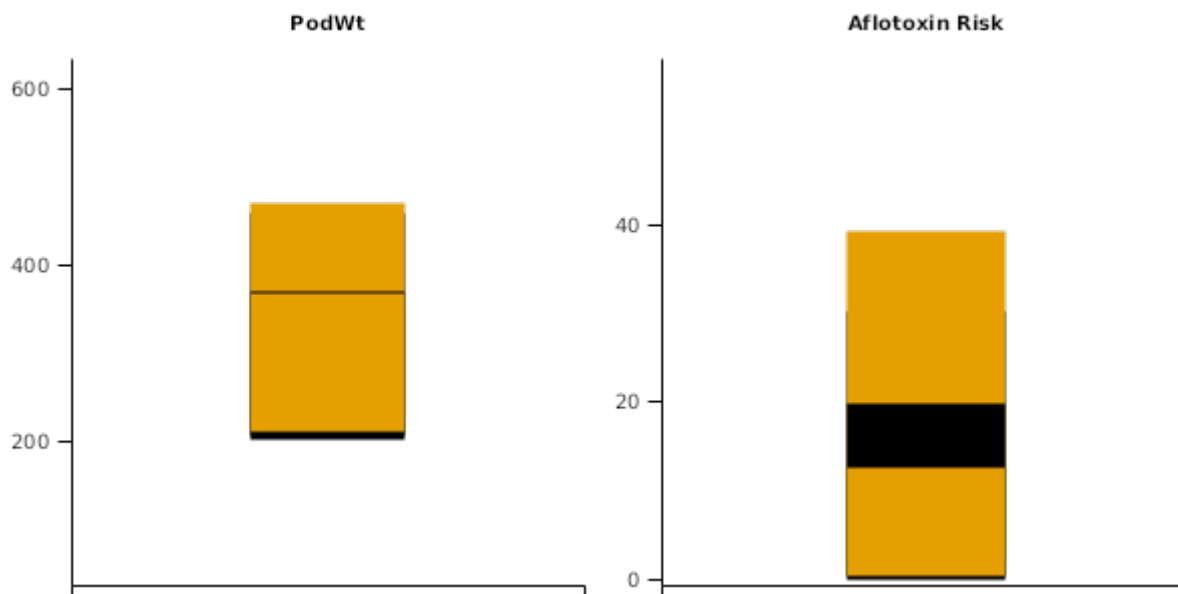
Experiment Name	Design (Number of Treatments)
Aflotoxin	Cv x Sow (4)



### 3.3.1 Aflotoxin

This sensibility test demonstrates predicted pod yields for rainfed crops for two sowing dates (mid November and mid December) for Kingary, Queensland, Australia. Pod yields should range from 2t/ha to 5 t/ha with decreased risk from Aflotoxin for later sowing and later maturing cultivars.

#### 3.3.1.1 Graphs



### 3.4 NewZealand

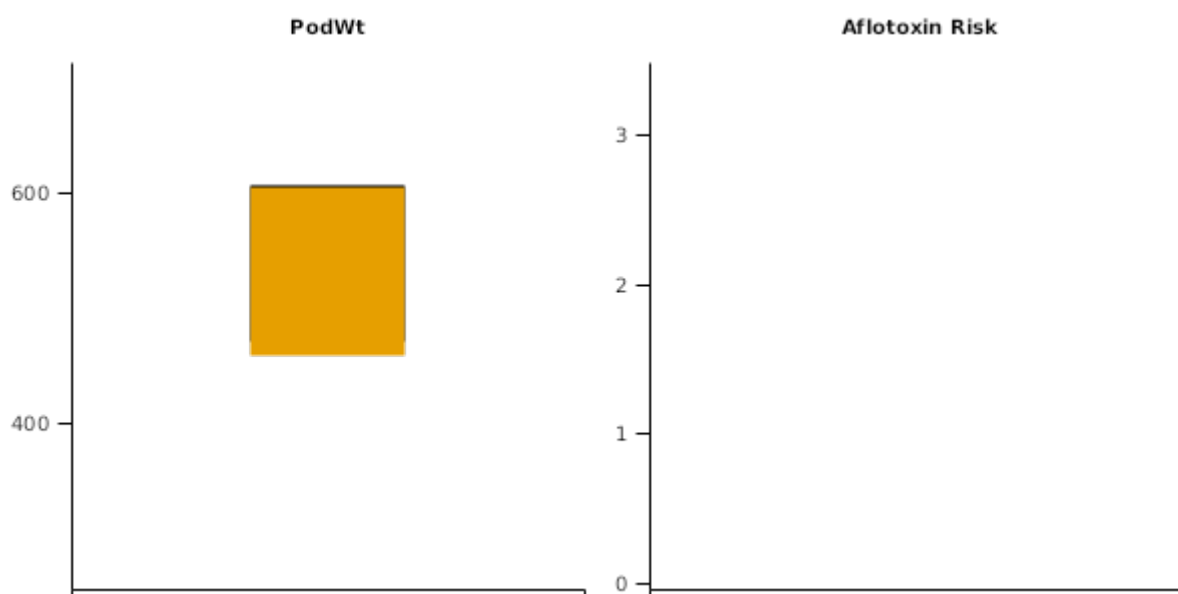
List of experiments.

Experiment Name	Design (Number of Treatments)
NewZealand	Water (2)

#### 3.4.1 NewZealand

Peanuts are currently being evaluated as a crop for the Northlands region of northern New Zealand. Test crops in the region are currently yielding up to 6-7 t/ha (Fresh Weight), which would correspond to a dry pod weight of 5-6 t/ha (16%-20% moisture).

#### 3.4.1.1 Graphs



## 4 UnderDevelopment

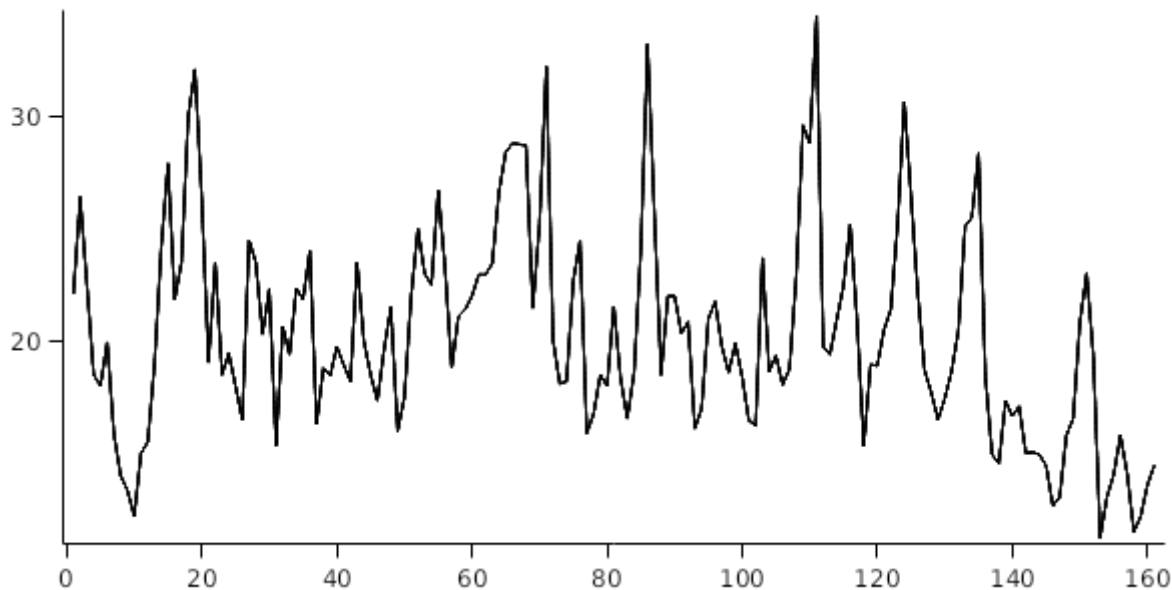
## List of experiments.

Experiment Name	Design (Number of Treatments)
Roseworthy9798	Pop x SR (16)

### 4.1 Roseworthy9798

This dataset (Porter, 2000) studied the impact of spatial arrangement on peanut crop microenvironment for the cultivar VB97. The experiment was grown under irrigated conditions in at Roseworthy in South Australia. The soil type was clay-loam with good water holding capacity. well-drained soil. Supplementary irrigations were made to avoid moisture stress. Treatments included a full factorial of population (8.8, 27.3, 39.4 and 58.8 plants/m<sup>2</sup>) and Spatial Ratio (1:1, 1:2.2, 1:4.6 and 1:7.2 Inter-row:Intra-row).

### Growing Season Mean Air Temperature



## 5 References

- Bell, M., 1986. Effect of sowing date on growth and development of irrigated peanuts, *Arachis hypogaea* L. cv. Early Bunch, in a monsoonal tropical environment.. *Australian Journal of Agricultural Research* 37, 361-373.
- Bell, MJ, Wright, GC, 1994. The N<sub>2</sub>-fixing capacity of peanut cultivars with differing assimilate partitioning characteristics. *Australian Journal of Agricultural Research* 45, 1455-1468.
- Bell, MJ, Wright, GC, Hammer, GL, 1992. Night Temperature Affects Radiation-Use Efficiency in Peanut. *Crop Science* 32, 1329-1335.
- [Brown, Hamish E., Huth, Neil I., Holzworth, Dean P., Teixeira, Edmar I., Zyskowski, Rob F., Hargreaves, John N. G., Moot, Derrick J., 2014. Plant Modelling Framework: Software for building and running crop models on the APSIM platform. \*Environmental Modelling and Software\* 62, 385-398.](#)
- Davis, David P., Mack, Timothy P., 1991. Relations Between Leaf Area Index and Growth Characteristics of Florunner, Southern Runner, and Sunrunner Peanut. *Peanut Science* 18, 30-37.
- Elli, Elvis, Huth, Neil, Sentelhas, Paulo, Carneiro, Rafaela, Alcarde Alvares, Clayton, 2020. Global sensitivity-based modelling approach to identify suitable Eucalyptus traits for adaptation to climate variability and change. in *silico plants* 2.
- [Jones, C.A., Kiniry, J.R., Dyke, P.T., 1986. CERES-Maize: a simulation model of maize growth and development..](#)
- Monteith, J. L., Moss, C. J., 1977. Climate and the Efficiency of Crop Production in Britain [and Discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 281 (980), 277-294.
- Porter, Wade, 2000. Potential for peanut production in Southern Australia..
- Reyenga, P.J., Howden, S. M., Meinke, H., McKeon, G.M., 1999. Modelling global change impacts on wheat cropping in south-east Queensland, Australia. *Environmental Modelling & Software* 14, 297-306.

Robertson, M. J., Carberry, P. S., Huth, N. I., Turpin, J. E., Probert, M. E., Poulton, P. L., Bell, M., Wright, G. C., Yeates, S. J., Brinsmead, R. B., 2002. Simulation of growth and development of diverse legume species in APSIM. *Australian Journal of Agricultural Research* 53 (4), 429-446.