

## 1 The APSIM Maize Model

*Brown, H.E., Teixeira, E.I., Huth, N.I. and Holzworth, D.P.*

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
Structure	Models.PMF.Struct.Structure
Grain	Models.PMF.Organs.ReproductiveOrgan
Root	Models.PMF.Organs.Root
Leaf	Models.PMF.Organs.Leaf
Husk	Models.PMF.Organs.GenericOrgan
Rachis	Models.PMF.Organs.GenericOrgan
Stem	Models.PMF.Organs.GenericOrgan
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).

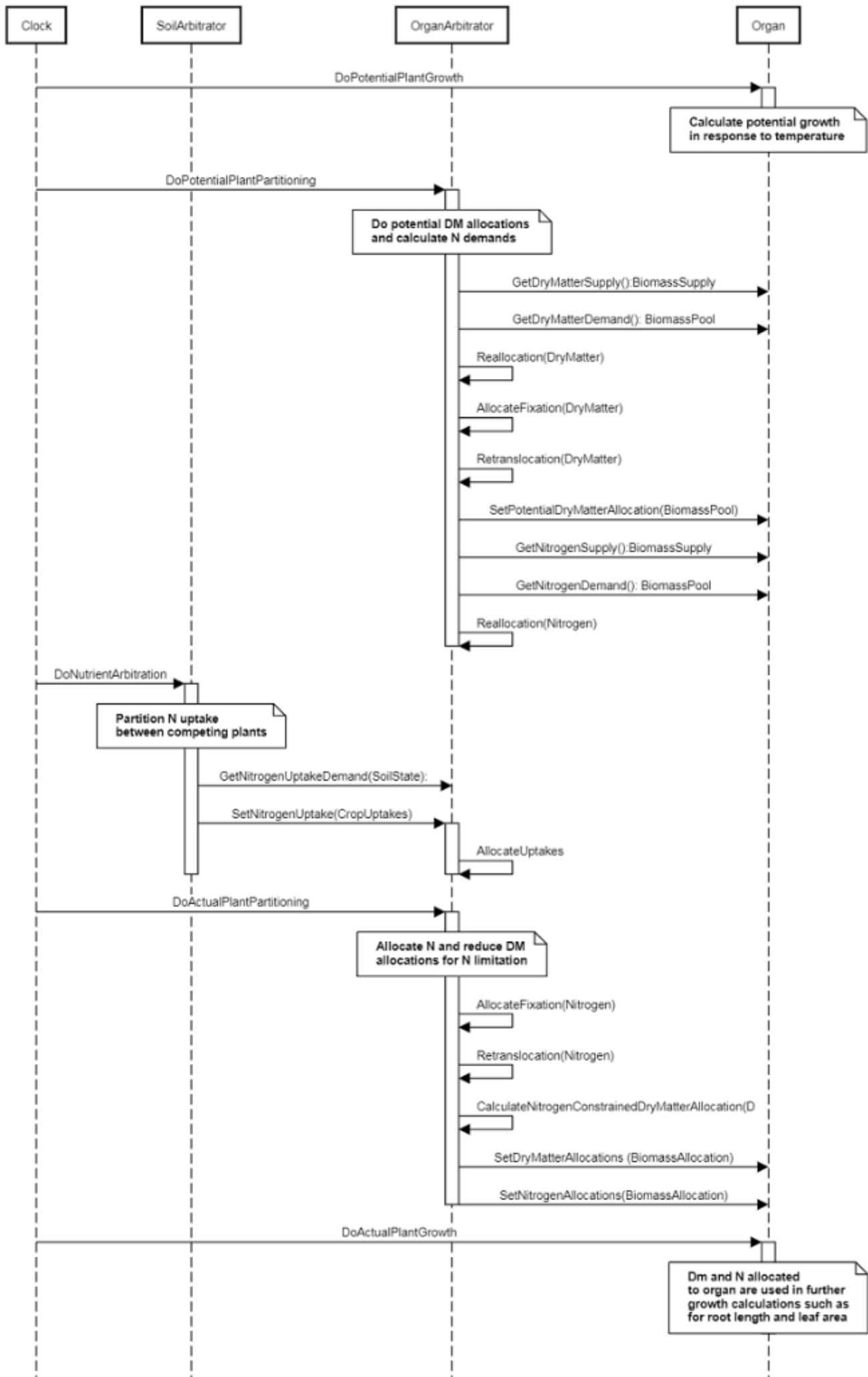
\* **Retranslocation supply.** Storage biomass that may be moved from organs to meet demands of other organs.

\* **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.

### 1.2.1 ThermalTime

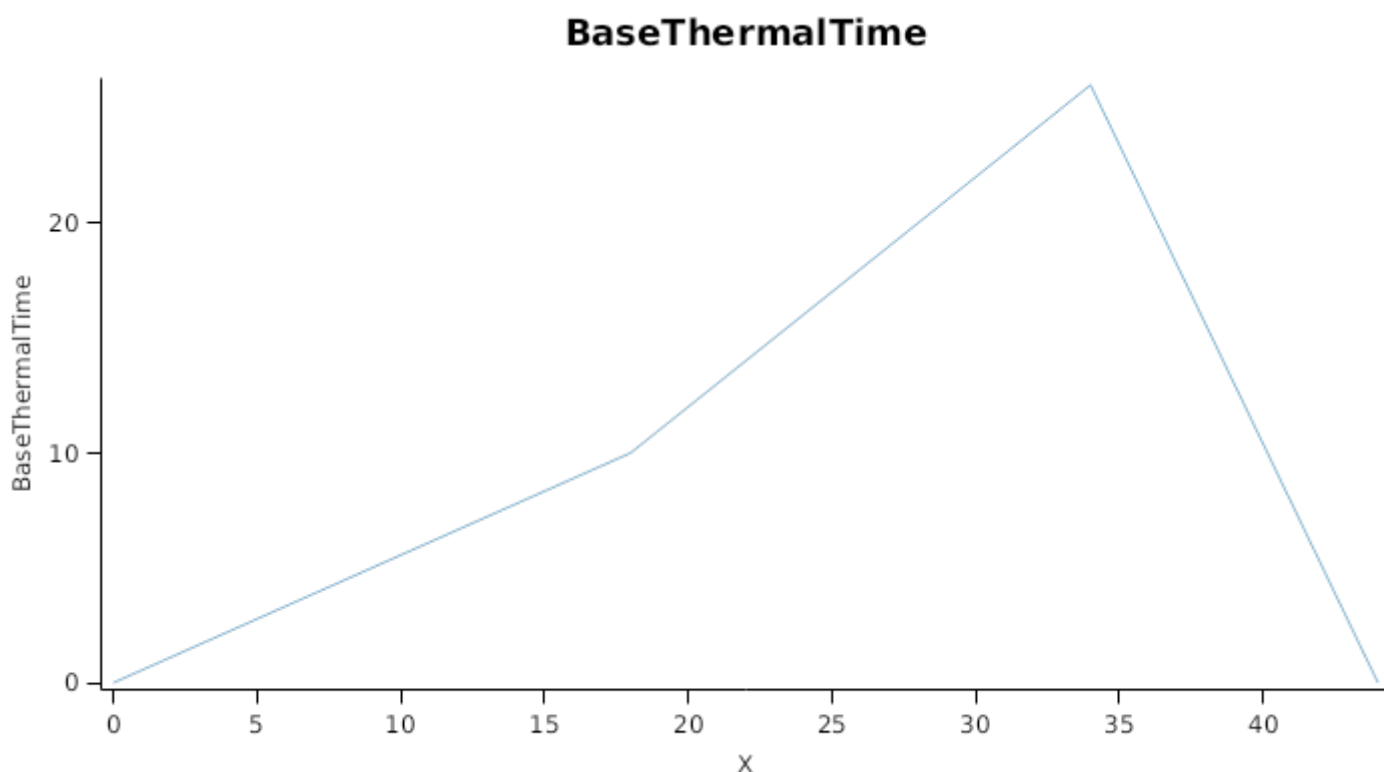
$$\text{ThermalTime} = \text{BaseThermalTime}$$

BaseThermalTime 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 BaseThermalTime

X	BaseThermalTime
0.0	0.0
18.0	10.0
26.0	18.0
34.0	26.0
44.0	0.0



### 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

Phase Number	Phase Name	Initial Stage	Final Stage
3	Juvenile	Emergence	EndJuvenile
4	Photosensitive	EndJuvenile	FloralInitiation
5	LeafAppearance	FloralInitiation	FlagLeaf
6	FlagLeafToFlowering	FlagLeaf	Flowering
7	FloweringToGrainFilling	Flowering	StartGrainFill
8	GrainFilling	StartGrainFill	EndGrainFill
9	Maturing	EndGrainFill	Maturity
10	MaturityToHarvestRipe	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:

$$\text{ShootRate} = 0.6 \text{ (deg day/mm),}$$

$$\text{ShootLag} = 55 \text{ (deg day),}$$

*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:

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

### 1.2.4 Juvenile

This phase goes from emergence to endjuvenile.

The *Target* for completion is calculated as:

$$\text{Target} = 200 \text{ (deg day)}$$

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

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

### 1.2.5 Photosensitive

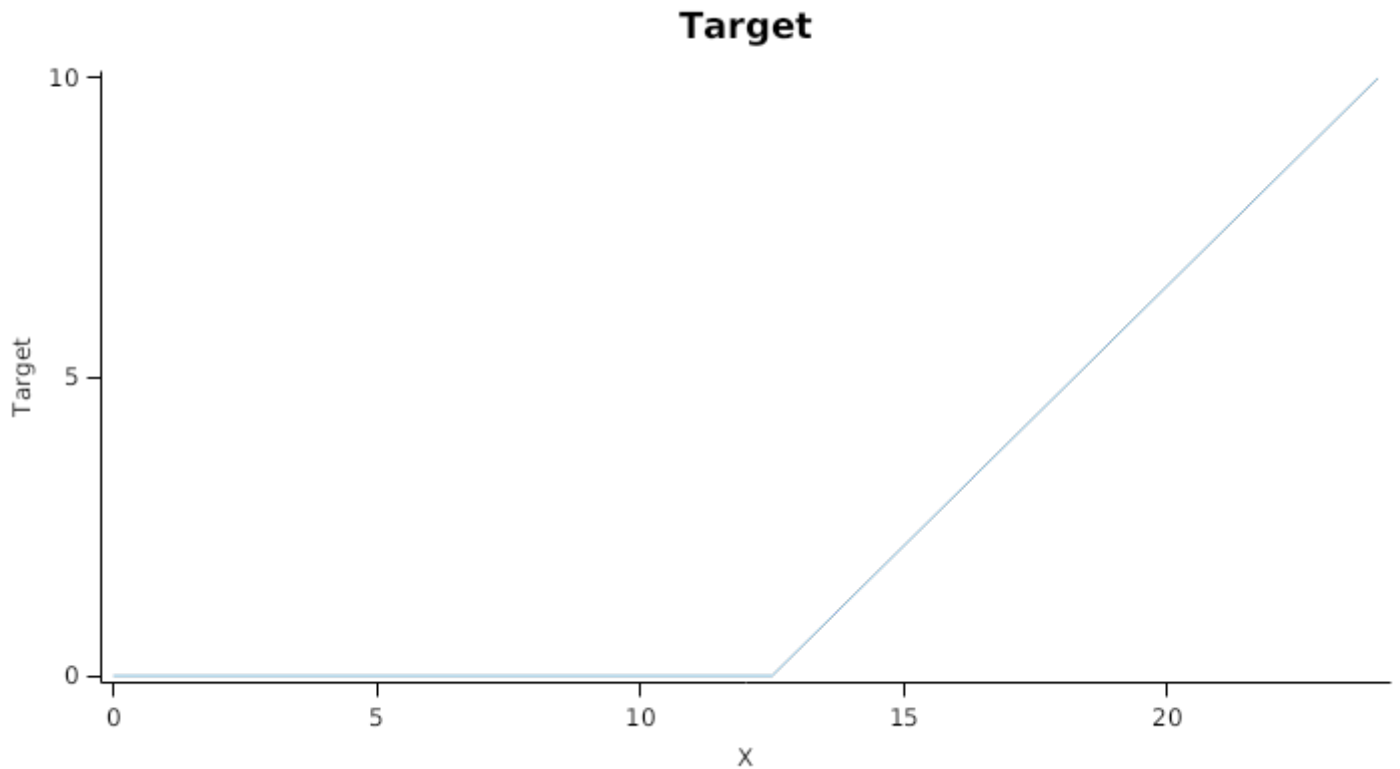
This phase goes from endjuvenile to floralinitiation.

The *Target* for completion is calculated as:

*Target* is calculated using linear interpolation

X	Target
0.0	0.0
12.5	0.0

X	Target
24.0	10.0



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

*Progression* = [Phenology].*ThermalTime*

### 1.2.6 LeafAppearance

This phase goes from floralinitiation to flagleaf and it continues until the final main-stem leaf has finished expansion. The duration of this phase is determined by leaf appearance rate (Structure.Phyllchron) and the number of leaves produced on the mainstem (Structure.FinalLeafNumber)

*ThermalTime* = [Phenology].*ThermalTime*

*FinalLeafNumber* = [Structure].*FinalLeafNumber*

*LeafNumber* = [Leaf].*ExpandedCohortNo* + [Leaf].*NextExpandingLeafProportion*

*FullyExpandedLeafNo* = [Leaf].*ExpandedCohortNo*

*InitialisedLeafNumber* = [Leaf].*InitialisedCohortNo*

### 1.2.7 FlagLeafToFlowering

This phase goes from flagleaf to flowering.

The *Target* for completion is calculated as:

Target = 50 (deg day)

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

*Progression* = [Phenology].*ThermalTime*

### 1.2.8 FloweringToGrainFilling

This phase goes from flowering to startgrainfill.

The *Target* for completion is calculated as:

Target = 120 (deg day)

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

*Progression* = *[Phenology].ThermalTime*

### **1.2.9 GrainFilling**

This phase goes from startgrainfill to endgrainfill.

The *Target* for completion is calculated as:

Target = 550 (deg day)

*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 = 10 (deg day)

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

*Progression* = *[Phenology].ThermalTime*

### **1.2.11 MaturityToHarvestRipe**

This phase goes from maturity to harvestripe.

The *Target* for completion is calculated as:

Target = 10 (deg day)

*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 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)

### **1.2.14 FloweringDAS**

Before Flowering

PreEventValue = 0

On Flowering the value is set to:

*PostEventValue* = *[Plant].DaysAfterSowing*

### **1.2.15 MaturityDAS**

Before Maturity

PreEventValue = 0

On Maturity the value is set to:

$PostEventValue = [Plant].DaysAfterSowing$

## 1.2.16 BBCH

### 1.2.16.1 BBCH

This model calculates a BBCH growth stage value based upon the current phenological growth stage within the model. The model uses information regarding germination, emergence and leaf appearance for early growth stages (BBCH stages 0 to 39).

BeginStage	Growth Phase	Description
1	Germinating	$BBCH = 5 \times FractionThroughPhase$
2	Emerging	$BBCH = 5 + 5 \times FractionThroughPhase$
3	Juvenile	$BBCH = 10 + (Leaf.AppearedCohortNo - 1)$
4	PhotoSensitive	$BBCH = 10 + (Leaf.AppearedCohortNo - 1)$
5	LeafAppearance	$BBCH = 30 + LeavesAppearedInPhase$

BBCHSTages 11-19 assume the droopy leaf method of measuring leaf appearance and to translate this to a model variable we assume droopy leaves are one fewer than the number of tips visible. We assume that the beginning of stem extension (BBCH 30) corresponds to the floral initiation stage in the model (Stage 5). Scores between 31 and 39 depend on the number of nodes visible. The model does not simulate nodes explicitly so we assume that node appearance occurs at the same rate as leaf appearance and add the number of leaves that have appeared in the LeafAppearance Phase to give an estimate of BBCH score. BBCH stage 50 occurs when the tassel is just visible. This is assumed to occur at the same time as the appearance of the tip of the flag leaf. This occurs toward the end of the leaf appearance phase. While still in the leaf appearance phase after flag leaf tip appearance BBCH score is calculated as:  $BBCH = 50 + 5 * FractionFlagLeafExpansion$ . This assumes the tassel will be half emerged when the flag leaf is full expanded. The model then uses simulated phenological growth stages for BBCH stages 55 to 99.

Stage	APSIM Name	BBCH translation
6.0	FlagLeafFullyExpanded	55 - Mid Tassel Emergence
7.0	Flowering	65 - Mid flowering
8.0	StartGrainFill	70 - Beginning of Grain development
9.0	EndGrainFill	87 - Physiological maturity
10	Maturity	99 - Harvest Product

## 1.3 Structure

The structure model simulates morphological development of the plant to inform the Leaf class when and how many leaves and branches appear and provides an estimate of height.

### 1.3.1 Plant and Main-Stem Population

The *Plant.Population* is set at sowing with information sent from a manager script in the Sow method. The *PrimaryBudNumber* is also sent with the Sow method. The main-stem population (*MainStemPopn*) for Maize is calculated as:

$$MainStemPopn = Plant.Population \times PrimaryBudNumber$$

Primary bud number is > 1 for crops like potato and grape vine where there are more than one main-stem per plant

### 1.3.2 Main-Stem leaf appearance

Each day the number of main-stem leaf tips appeared (*LeafTipsAppeared*) is calculated as:

$$LeafTipsAppeared += DeltaTips$$

Where *DeltaTips* is calculated as:

$$DeltaTips = ThermalTime / Phyllochron$$



Where *Phyllochron* is the thermal time duration between the appearance of leaf tips given by:

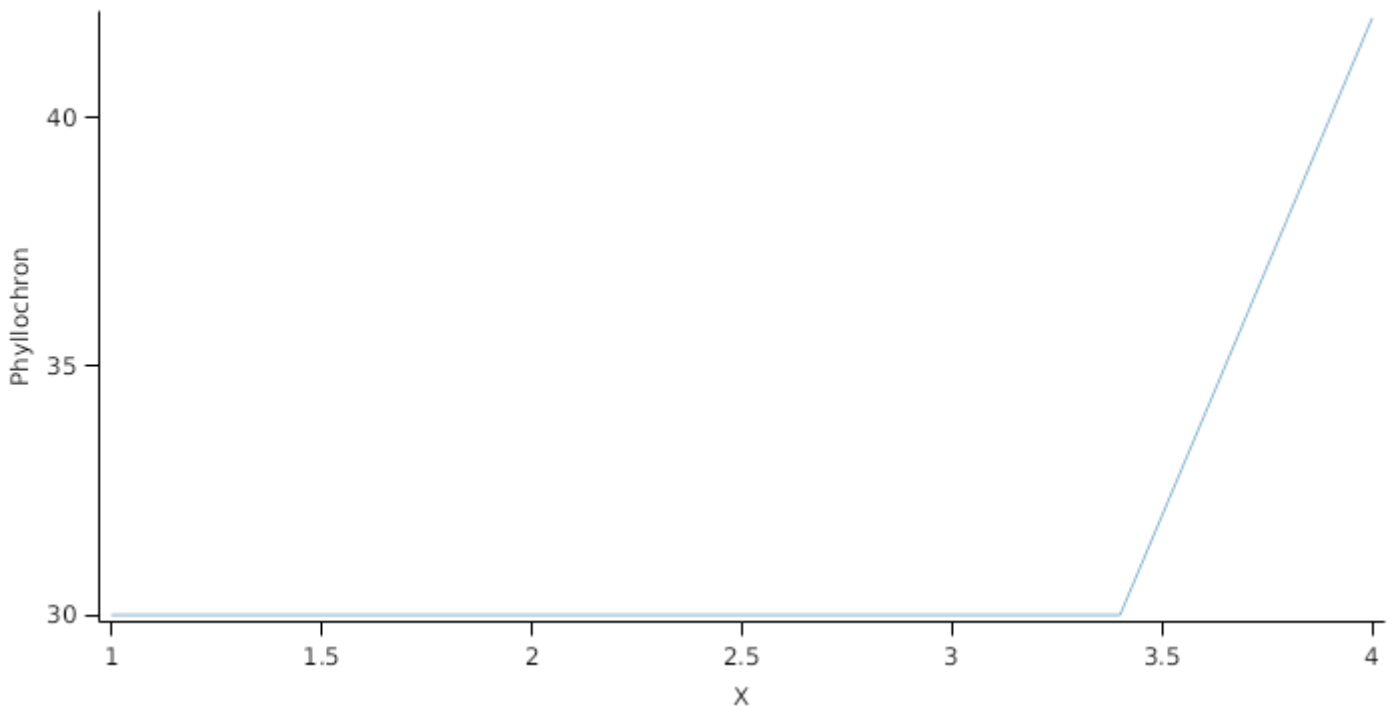
Phyllochron is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Phyllochron has a value between Emergence and FlagLeaf calculated as:

*Phyllochron* is calculated using linear interpolation

X	Phyllochron
1.0	30.0
3.4	30.0
4.0	42.0

## Phyllochron



*ThermalTime* is given by

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

*LeafTipsAppeared* continues to increase until *FinalLeafNumber* is reached where *FinalLeafNumber* is calculated as:

$$\text{FinalLeafNumber} = \text{ValueToHold} \text{ until FlorallInitiation after which the value is fixed.}$$

$$\text{ValueToHold} = \text{PrimordiaAtEmergence} + \text{PrimordiaDuringJuvenilePhase}$$

$$\text{PrimordiaAtEmergence} = 6$$

$$\text{PrimordiaDuringJuvenilePhase} = \text{AddFunction} / \text{MeanPlastochron}$$

$$\text{AddFunction} = [\text{Phenology}].\text{Juvenile.Target} + [\text{Phenology}].\text{Photosensitive.Target}$$

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

PreEmerg has a value between Germination and Emergence calculated as:

$$\text{Constant} = 15$$

PostEmerg has a value between Emergence and FlagLeaf calculated as:

$$\text{MeanPlastochron} = \text{AccumulatedPlastochron} / \text{Days}$$

*AccumulatedPlastochron* = Accumulated *Plastochron* between emergence and floralinitiation

*Plastochron* = [Structure].Phyllochron x *LeafTipsPerPrimordia*

*LeafTipsPerPrimordia* = 0.5

*Days* = Accumulated *Constant* between emergence and floralinitiation

*Constant* = 1

### 1.3.3 Branching and Branch Mortality

The total population of stems (*TotalStemPopn*) is calculated as:

*TotalStemPopn* = *MainStemPopn* + *NewBranches* - *NewlyDeadBranches*

Where:

*NewBranches* = *MainStemPopn* x *BranchingRate*

*BranchingRate* is given by:

*BranchingRate* = 0

*NewlyDeadBranches* is calculated as:

*NewlyDeadBranches* = (*TotalStemPopn* - *MainStemPopn*) x *BranchMortality*

where *BranchMortality* is given by:

*BranchMortality* = *ShadeInducedBranchMortality* + *DroughtInducedBranchMortality*

*ShadeInducedBranchMortality* = 0

*DroughtInducedBranchMortality* = 0

### 1.3.4 Height

The height of the crop is calculated by the *HeightModel*

Height is used by the MicroClimate model to calculate the aerodynamic resistance used for calculation of potential transpiration. Calculates the potential height increment and then multiplies it by the smallest of any children functions (Child functions represent stress).

## 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

*DMConversionEfficiency* = 1

*RemobilisationCost* = 0

*InitialGrainProportion* = 0.05

*MaximumPotentialGrainSize* = 0.3 (g)

*MaximumGrainsPerCob* = 700 (number)

*MinimumNConc* = 0.008 (g/g)

*WaterContent* = 0.12 (g/g)

*CarbonConcentration* = 0.4

### 1.4.2 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.

InitialPhase has a value between Flowering and StartGrainFill calculated as:

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$\text{NumberFunction} = [\text{Grain}].\text{NumberFunction}$$

$$\text{FillingDuration} = [\text{Phenology}].\text{FloweringToGrainFilling}.\text{Target}$$

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

$$\text{PotentialSizeIncrement} = [\text{Grain}].\text{InitialGrainProportion} \times [\text{Grain}].\text{MaximumPotentialGrainSize}$$

LinearPhase has a value between StartGrainFill and EndGrainFill calculated as:

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$\text{NumberFunction} = [\text{Grain}].\text{NumberFunction}$$

$$\text{FillingDuration} = [\text{Phenology}].\text{GrainFilling}.\text{Target}$$

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

$$\text{PotentialSizeIncrement} = \text{ProportionLinearPhase} \times [\text{Grain}].\text{MaximumPotentialGrainSize}$$

$$\text{ProportionLinearPhase} = 1 - [\text{Grain}].\text{InitialGrainProportion}$$

### 1.4.3 MaximumNConc

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

InitialPhase has a value between Flowering and StartGrainFill calculated as:

$$\text{InitialNConc} = 0.05$$

LinearPhase has a value between StartGrainFill and EndGrainFill calculated as:

$$\text{FinalNConc} = 0.013$$

### 1.4.4 NFillingRate

NFillingRate 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 Flowering and EndGrainFill calculated as:

Filling rate is calculated from grain number, a maximum mass to be filled and the duration of the filling process.

$$\text{NumberFunction} = [\text{Grain}].\text{NumberFunction}$$

$$\text{FillingDuration} = [\text{Phenology}].\text{FloweringToGrainFilling}.\text{Target} + [\text{Phenology}].\text{GrainFilling}.\text{Target}$$

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

$$\text{PotentialSizeIncrement} = [\text{Grain}].\text{MaximumNConc}.\text{LinearPhase}.\text{FinalNConc} \times [\text{Grain}].\text{MaximumPotentialGrainSize}$$

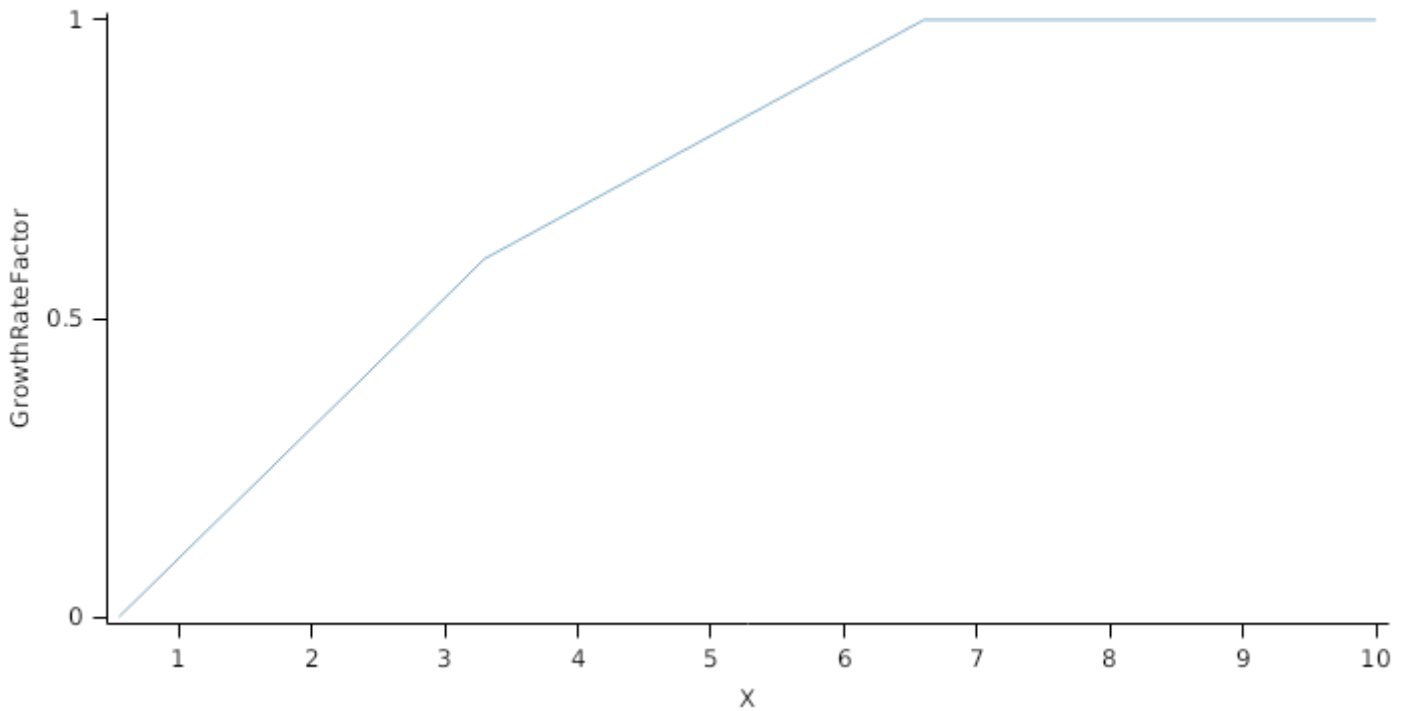
### 1.4.5 NumberFunction

$$\text{NumberFunction} = [\text{MaximumGrainsPerCob}] \times \text{GrowthRateFactor} \times [\text{Maize}].\text{Population}$$

GrowthRateFactor is calculated using linear interpolation

X	GrowthRateFactor
0.6	0.0
3.3	0.6
6.6	1.0
10.0	1.0

## GrowthRateFactor



### 1.4.6 YieldBuPerAcre

$$\text{YieldBuPerAcre} = [\text{Grain}].\text{Live.Wt} \times 10 \times 56 \times 0.4536 \times 2.471$$

### 1.4.7 GrowthGrainDevelopment

*GrowthGrainDevelopment* = Accumulated *DailyBiomassProduction* between flagleaf and startgrainfill

$$\text{DailyBiomassProduction} = [\text{Arbitrator}].\text{DM.TotalFixationSupply}$$

### 1.4.8 DaysGrainDevelopment

*DaysGrainDevelopment* = Accumulated *LenthOfADay* between flagleaf and startgrainfill

$$\text{LenthOfADay} = 1 \text{ (days/day)}$$

### 1.4.9 GrowthRateGrainDevelopment

$$\text{GrowthRateGrainDevelopment} = [\text{GrowthGrainDevelopment}] / [\text{DaysGrainDevelopment}] / [\text{Maize}].\text{Population}$$

### 1.4.10 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	0	0	0
Cut	100	0	0	0
Prune	0	0	80	0
Graze	60	0	20	0

### 1.4.11 DMDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

#### 1.4.12 NDemandPriorityFactors

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

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 ( $\text{kNO}_3$  or  $\text{kNH}_4$ ), the concentration of N form (ppm), and a soil moisture factor ( $\text{NUptakeSWFactor}$ ) which typically decreases as the soil dries.  $\text{NO}_3 \text{ uptake} = \text{NO}_3 \times \text{kNO}_3 \times \text{NO}_{3\text{ppm},i} \times \text{NUptakeSWFactor}$   $\text{NH}_4 \text{ uptake} = \text{NH}_4 \times \text{kNH}_4 \times \text{NH}_{4\text{ppm},i} \times \text{NUptakeSWFactor}$  As can be seen from the above equations, the values of  $\text{kNO}_3$  and  $\text{kNH}_4$  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 ( $\text{MaxDailyNUptake}$ ) 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.  $SW\ uptake = (SW_i - LL_i) \times KL_i \times KLModifier$

### 1.5.6 Constants

DMConversionEfficiency = 1

RemobilisationCost = 0

No effect of soil water content on root growth is currently captured in the model.

SoilWaterEffect = 1

Maximum daily N uptake is set to an arbitrarily high value.

MaxDailyNUptake = 20 (g/m<sup>2</sup>/d)

Jones and Kiniry (1986) used a constant root senescence fraction in the CERES-Maize model.

SenescenceRate = 0.005 (/d)

MaximumRootDepth = 1000000

MaximumNConc = 0.01 (g/g)

MinimumNConc = 0.01 (g/g)

SpecificRootLength = 40 (m/g)

KNO<sub>3</sub> = 0.03

KNH<sub>4</sub> = 0.02

MaintenanceRespirationFunction = 0

CarbonConcentration = 0.4

RootDepthStressFactor = 1

### 1.5.7 RootShape

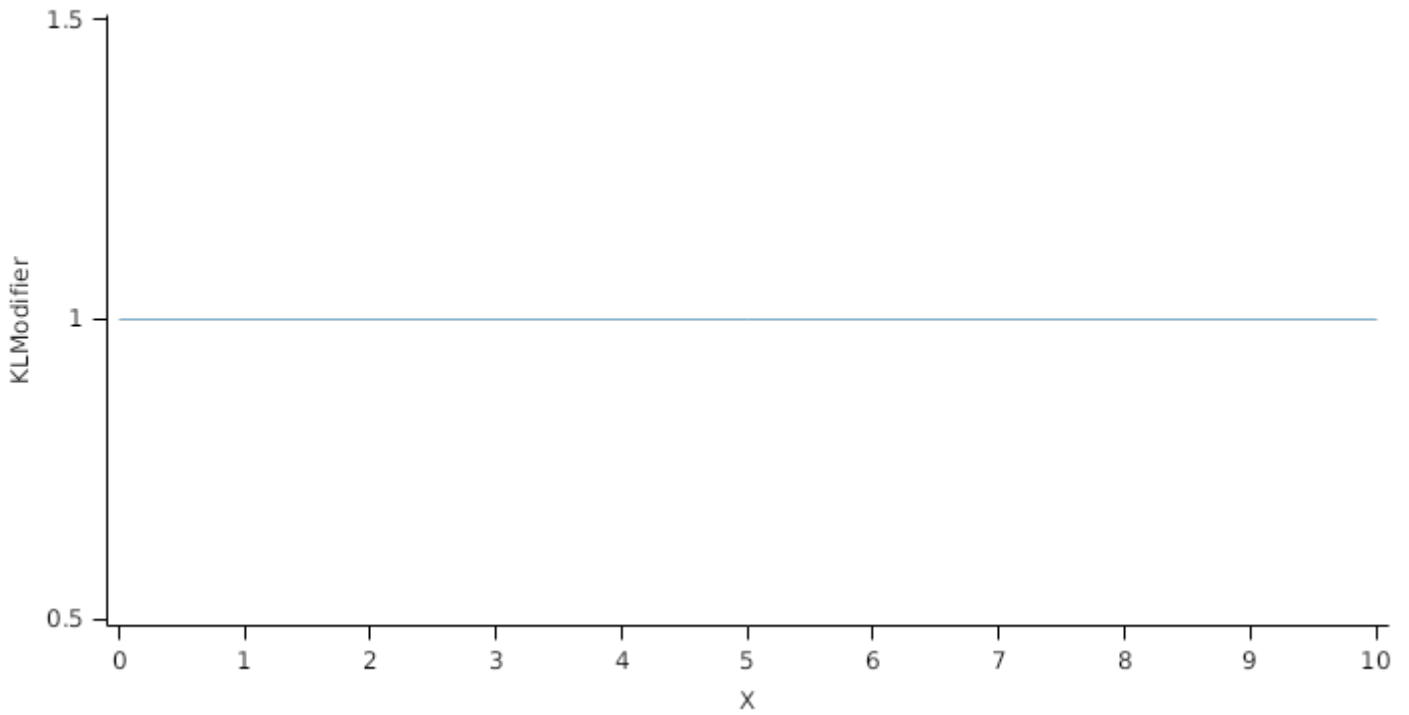
This model calculates the proportion of each soil layer occupied by roots.

### 1.5.8 KLModifier

*KLModifier* is calculated using linear interpolation

X	KLModifier
0.0	1.0
10.0	1.0

## KLModifier



### 1.5.9 RootFrontVelocity

$$\text{RootFrontVelocity} = \text{PotentialRootFrontVelocity} \times \text{TemperatureFactor} \times \text{WaterFactor}$$

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

[Hammer et al., 2009](#) used a constant extraction front velocity of 2.5cm per day in their analysis of historical maize yield trends in the US corn belt.

PreFlowering has a value between Germination and Flowering calculated as:

$$\text{RootFrontVelocity} = 25 \text{ (mm/d)}$$

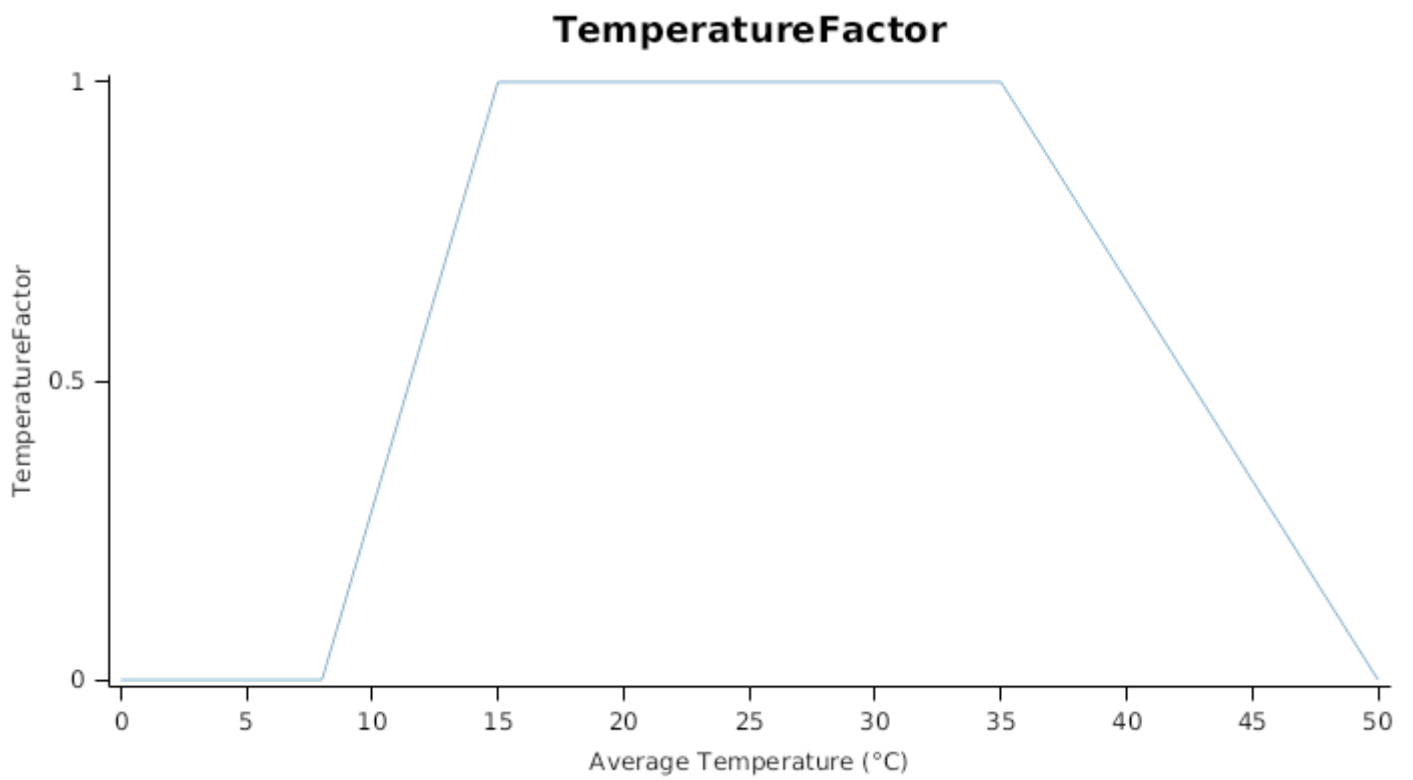
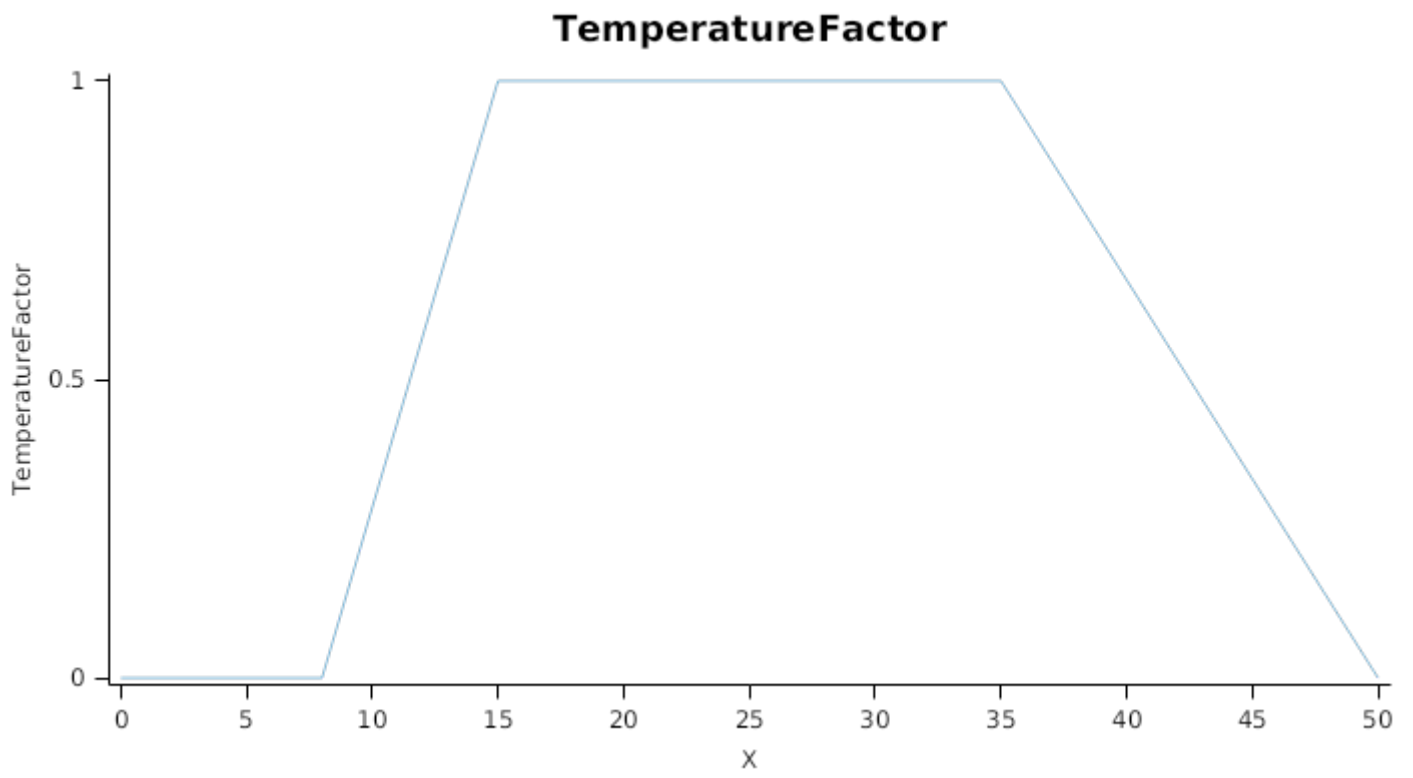
PostFlowering has a value between Flowering and Maturity calculated as:

$$\text{RootFrontVelocity} = 0 \text{ (mm/d)}$$

*TemperatureFactor* 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.

$$\text{MaximumTemperatureWeighting} = 0.5$$

X	TemperatureFactor
0.0	0.0
8.0	0.0
15.0	1.0
35.0	1.0
50.0	0.0

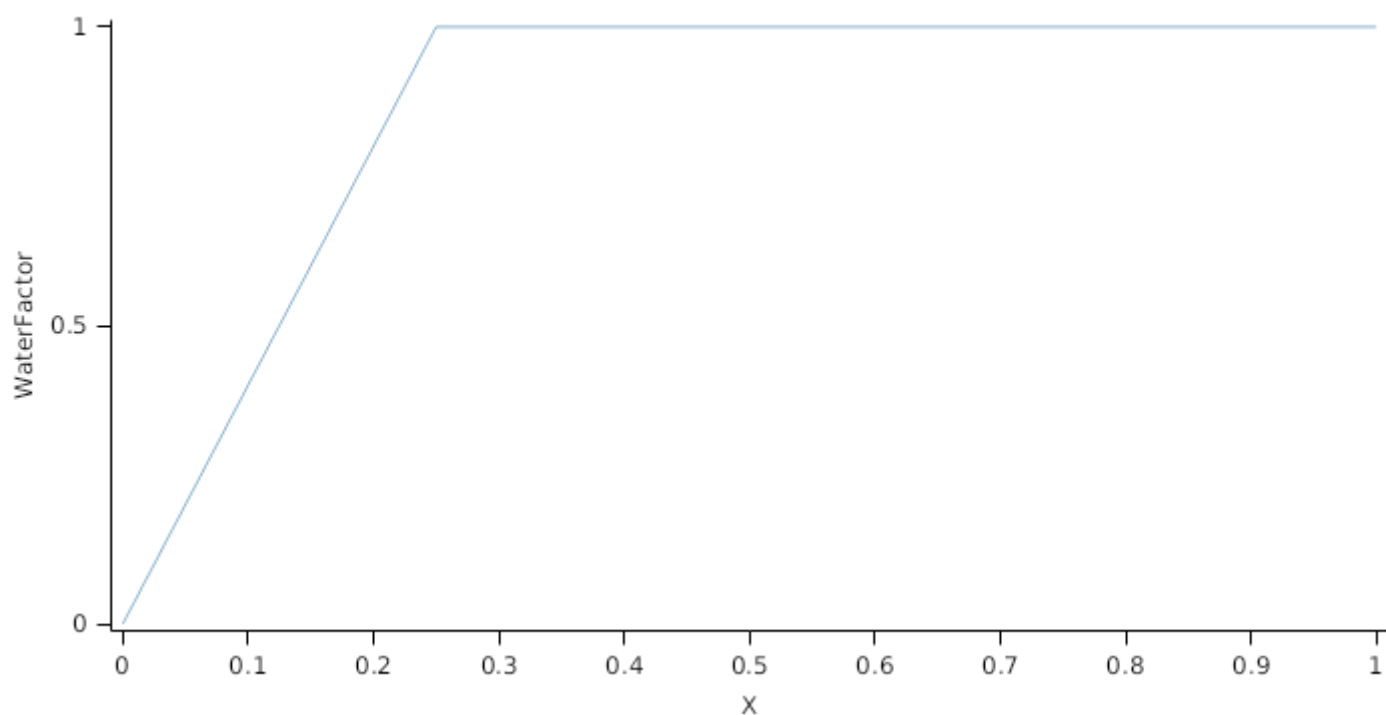


*WaterFactor* is calculated using linear interpolation

X	WaterFactor
0.0	0.0
0.2	1.0
1.0	1.0



## WaterFactor



### 1.5.10 NitrogenDemandSwitch

NitrogenDemandSwitch has a value between Germination and Maturity calculated as:

Constant = 1

### 1.5.11 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.

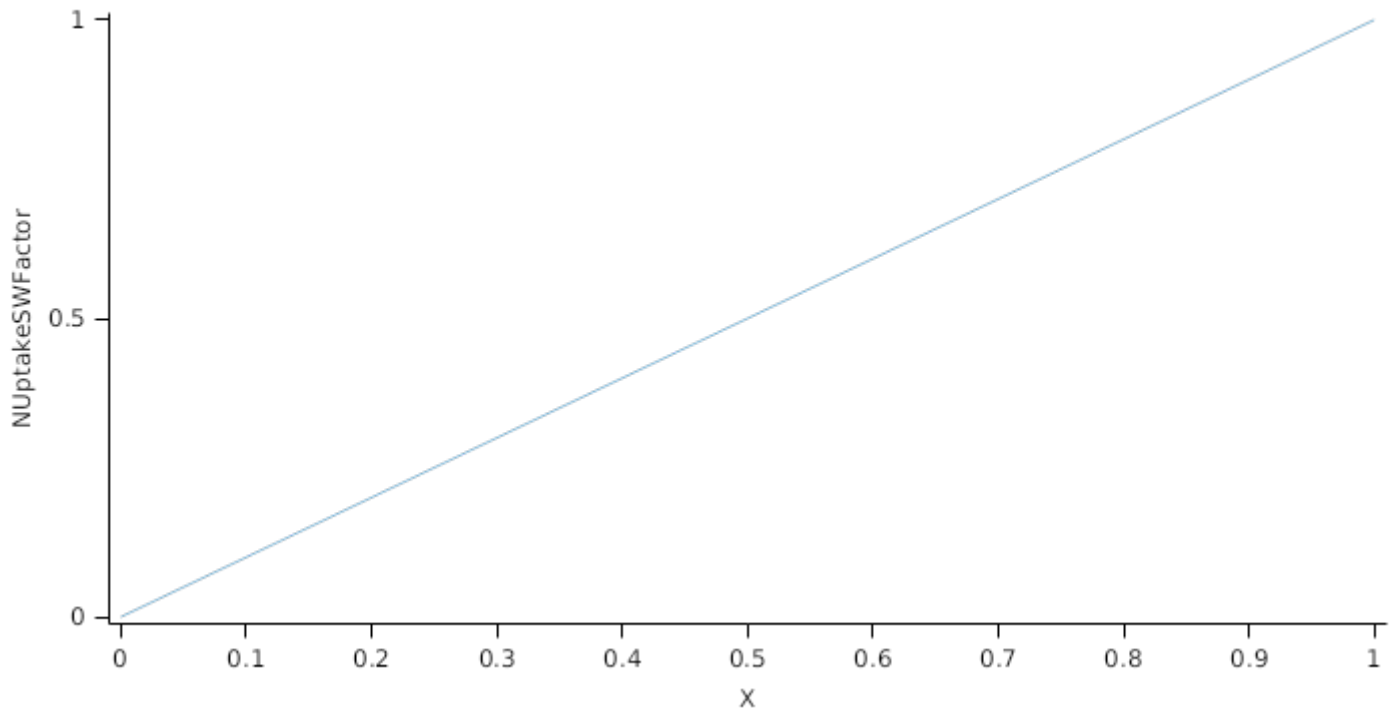
Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	20	0
Cut	0	0	30	0
Prune	0	0	10	0
Graze	0	0	15	0

### 1.5.12 NUptakeSWFactor

*NUptakeSWFactor* is calculated using linear interpolation

X	NUptakeSWFactor
0.0	0.0
1.0	1.0

## NUptakeSWFactor



### 1.5.13 DMDemands

#### 1.5.13.1 DMDemands

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

$Structural = DMDemandFunction \times StructuralFraction$

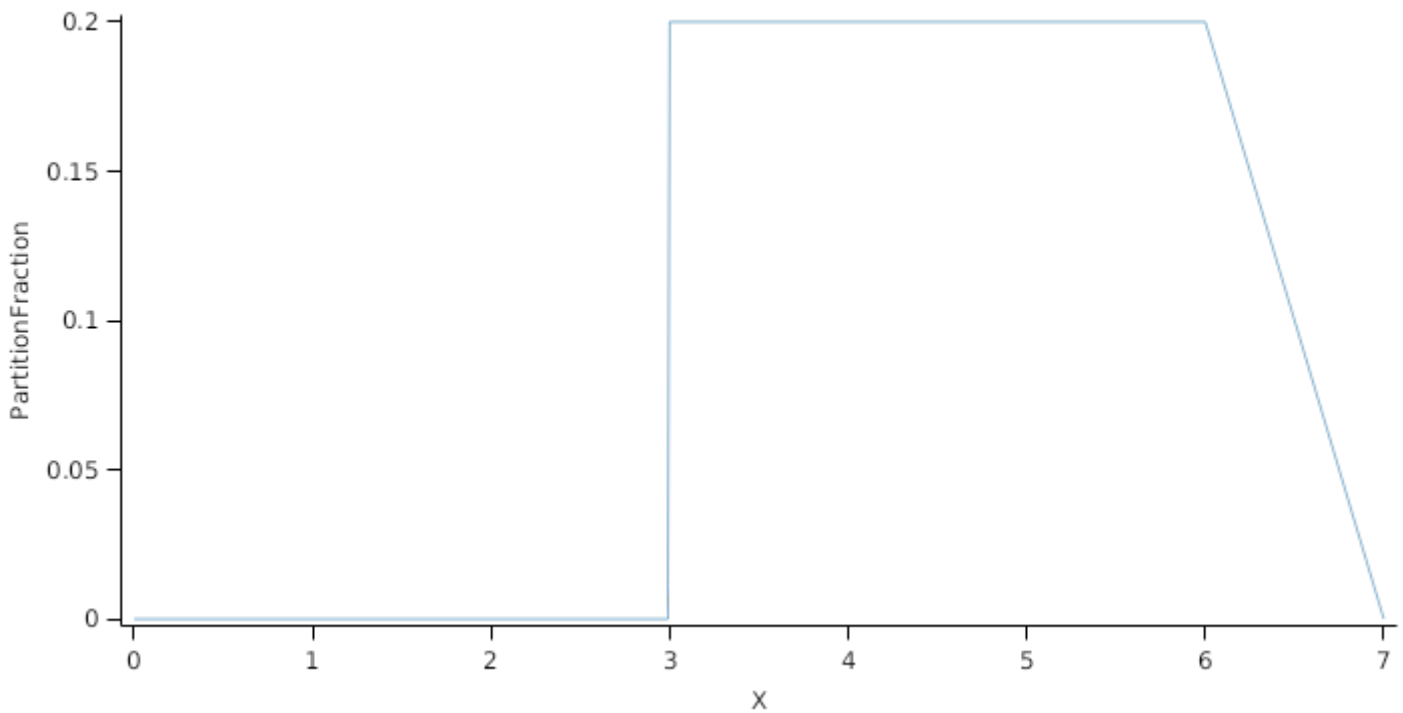
Returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs.

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

*PartitionFraction* is calculated using linear interpolation

X	PartitionFraction
0.0	0.0
3.0	0.0
3.0	0.2
6.0	0.2
7.0	0.0

## PartitionFraction



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

### 1.5.14 NDemands

#### 1.5.14.1 NDemands

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

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

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

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

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

### 1.5.15 CriticalINConc

$CriticalNConc = [Root].MinimumNConc$

### 1.5.16 InitialWt

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

Structural = 0.2 (g/plant)

Metabolic = 0

Storage = 0

### 1.6 Leaf

The leaves are modelled as a set of leaf cohorts and the properties of each of these cohorts are summed to give overall values for the leaf organ.

A cohort represents all the leaves of a given main- stem node position including all of the branch leaves appearing at the same time as the given main-stem leaf (Lawless et al., 2005).

The number of leaves in each cohort is the product of the number of plants per m<sup>2</sup> and the number of branches per plant. The *Structure* class models the appearance of main-stem leaves and branches. Once cohorts are initiated the *Leaf* class models the area and biomass dynamics of each.

It is assumed all the leaves in each cohort have the same size and biomass properties. The modelling of the status and function of individual cohorts is delegated to *LeafCohort* classes.

#### 1.6.1 Dry Matter Fixation

The most important DM supply from leaf is the photosynthetic fixation supply. Radiation interception is calculated from LAI using an extinction coefficient of:

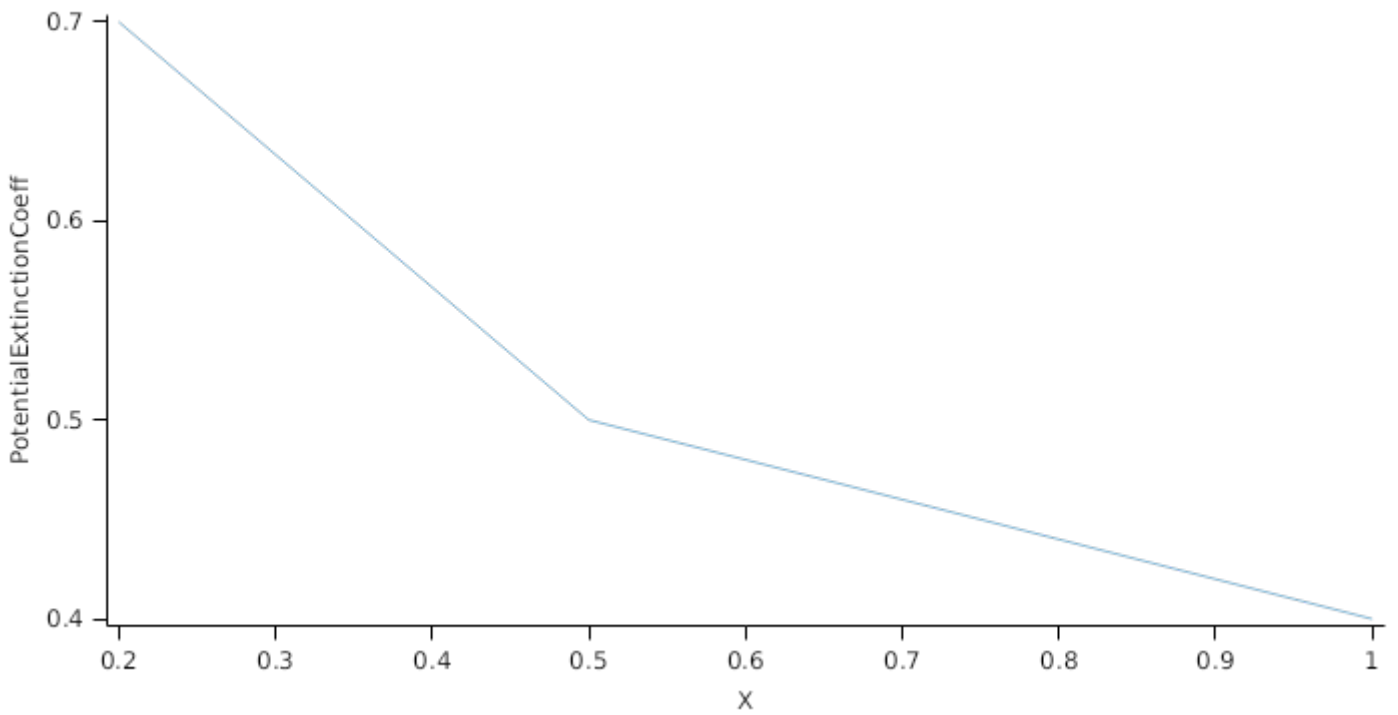
Extinction coefficient is specified in terms of a potential extinction coefficient which is then discounted during periods of water stress to account for the impact of leaf curling.

$ExtinctionCoeff = PotentialExtinctionCoeff \times WaterStressImpact$

*PotentialExtinctionCoeff* is calculated using linear interpolation

X	PotentialExtinctionCoeff
0.2	0.7
0.5	0.5
1.0	0.4

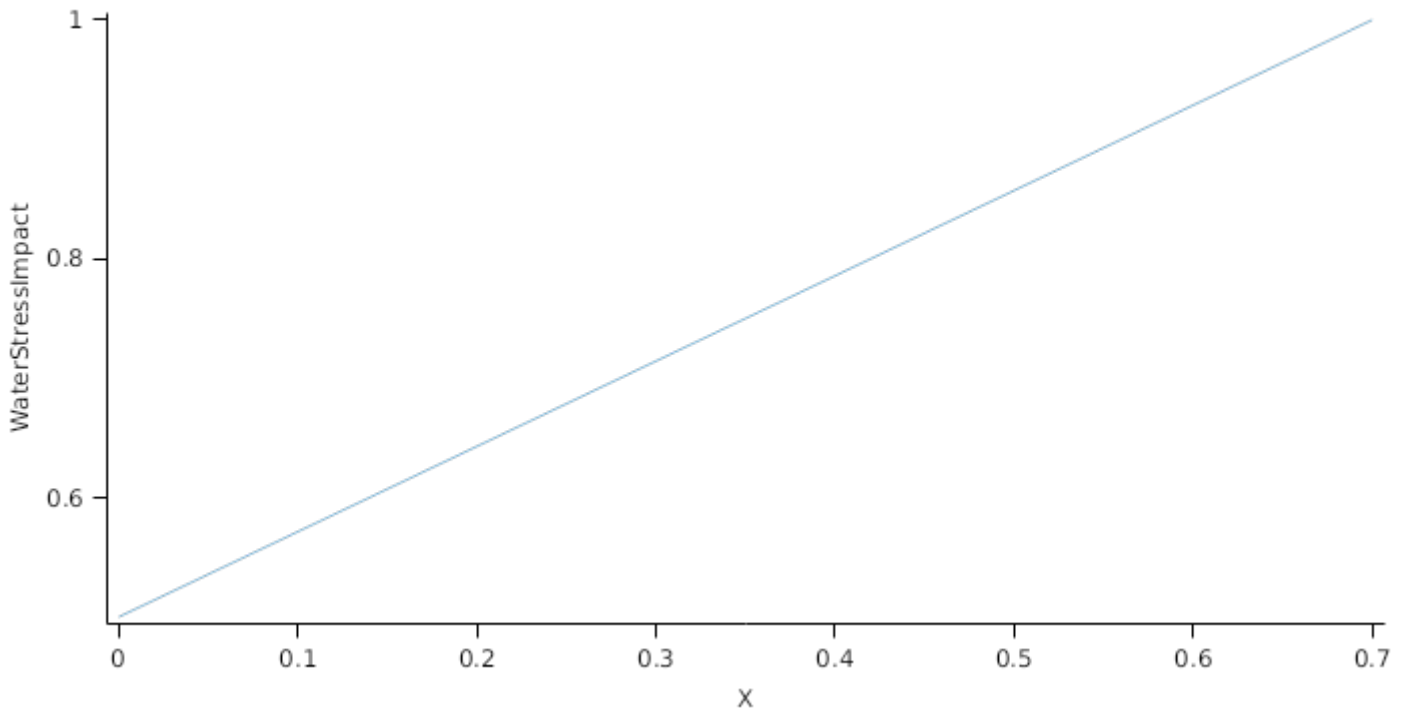
## PotentialExtinctionCoeff



*WaterStressImpact* is calculated using linear interpolation

X	WaterStressImpact
0.0	0.5
0.7	1.0

## WaterStressImpact



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 (FCO2).

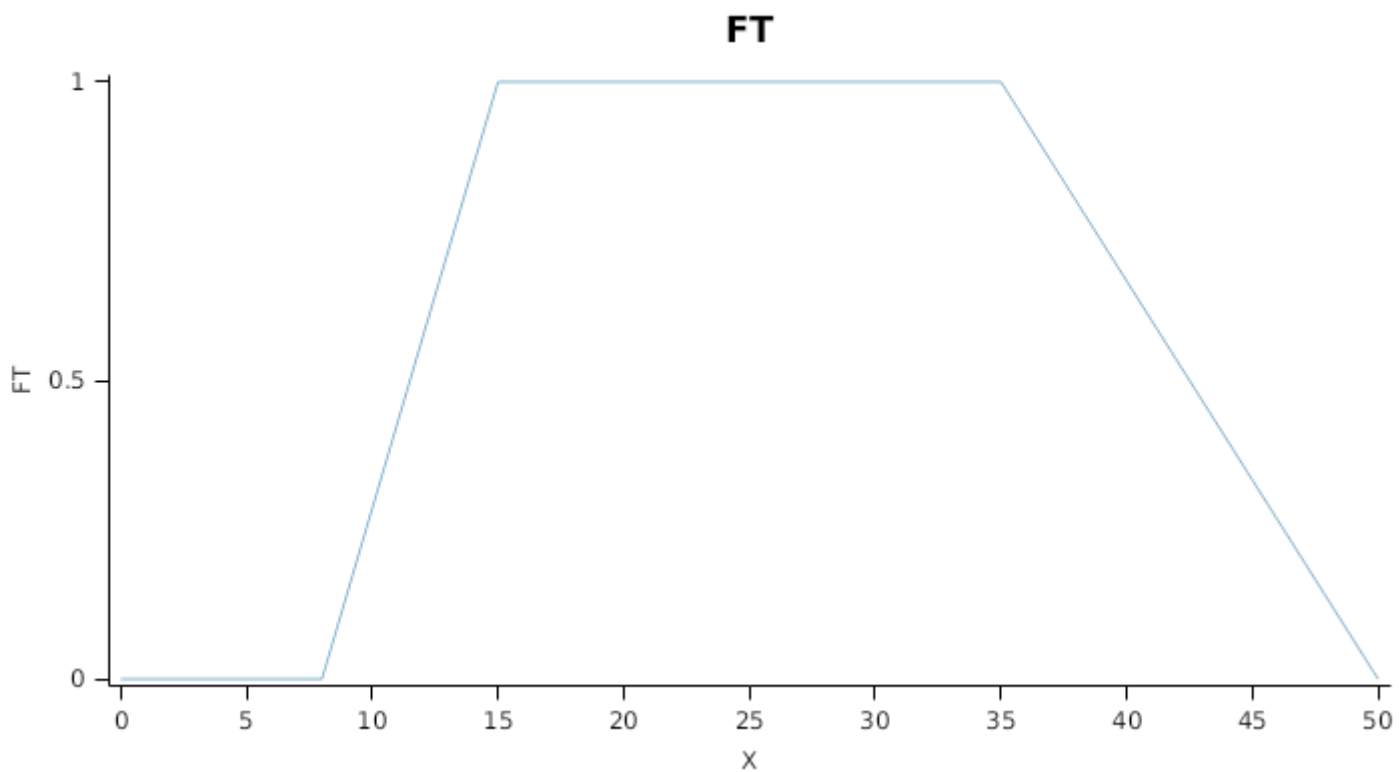
NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

RUE = 2

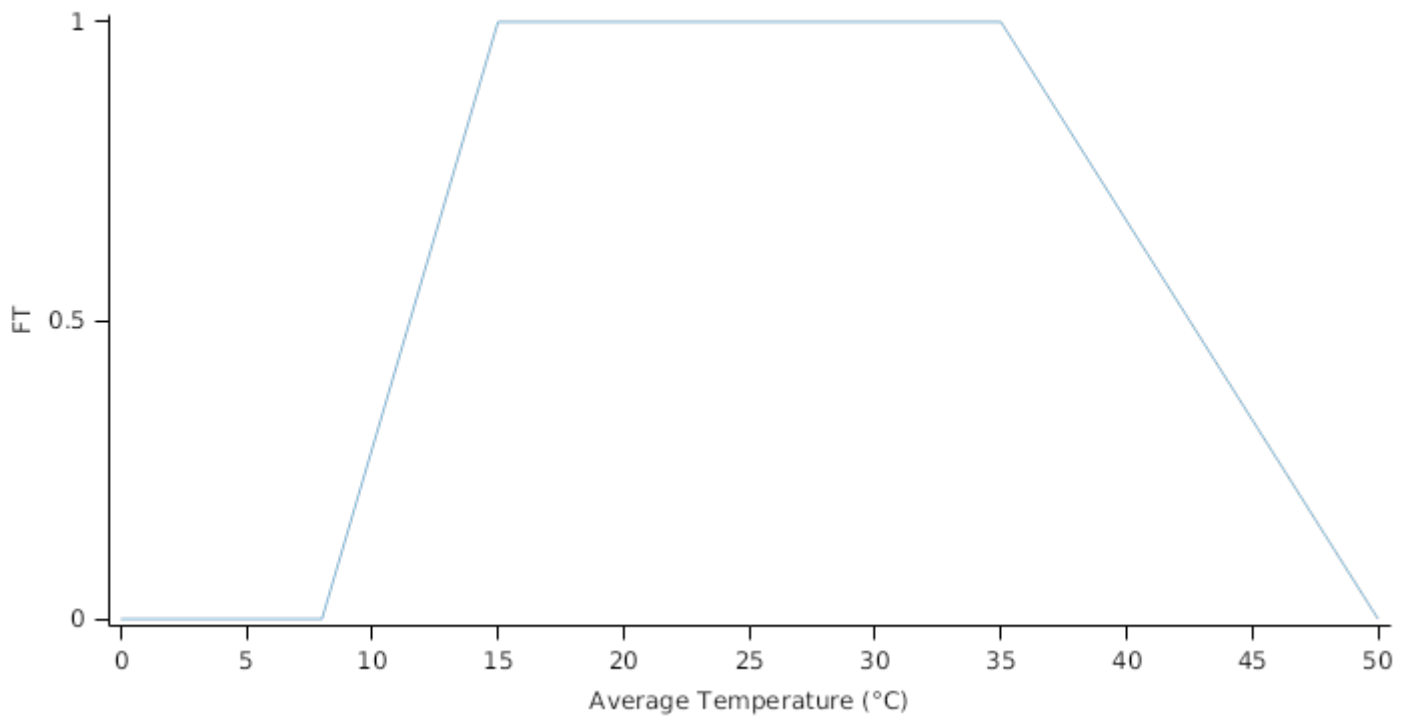
FT is calculated as a function of daily min and max temperatures, these are weighted toward max temperature according to the specified MaximumTemperatureWeighting factor. A value equal to 1.0 means it will use max temperature, a value of 0.5 means average temperature.

MaximumTemperatureWeighting = 0.75

X	FT
0.0	0.0
8.0	0.0
15.0	1.0
35.0	1.0
50.0	0.0



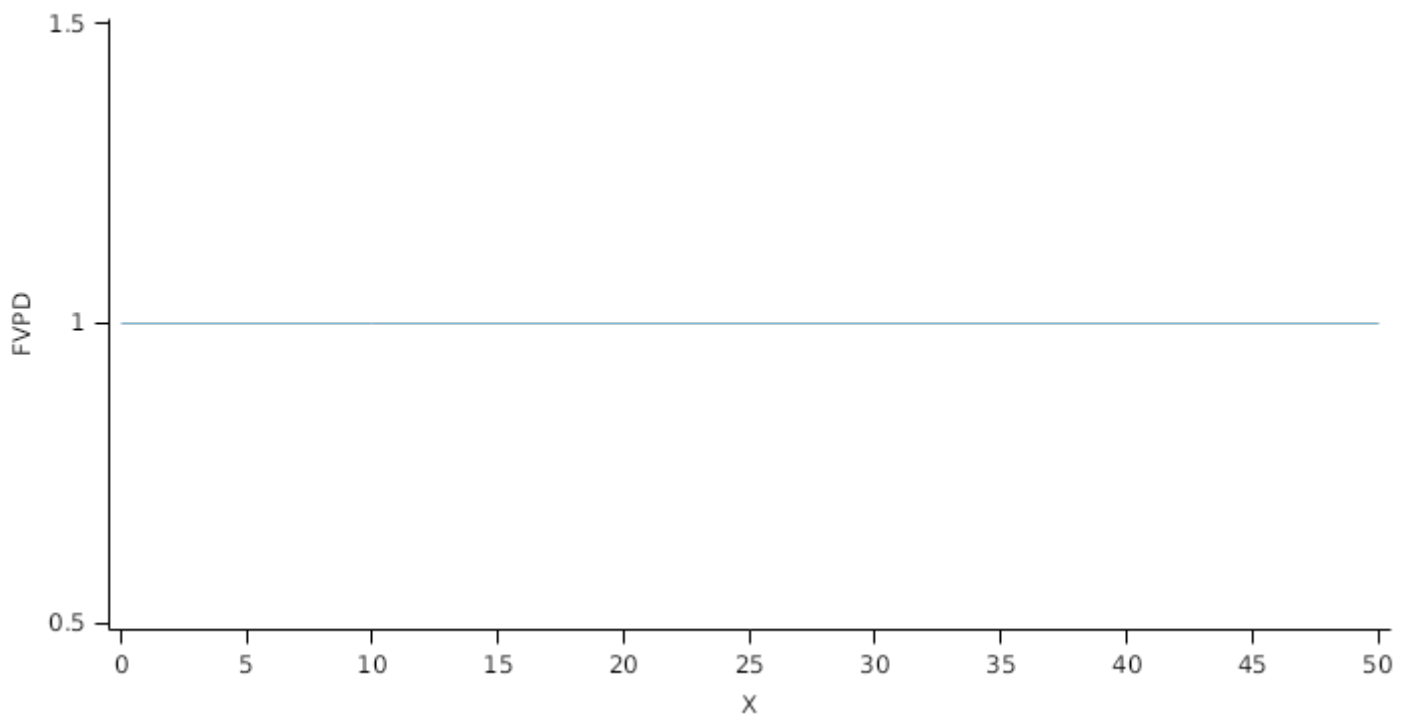
### FT



FVPD is calculated using linear interpolation

X	FVPD
0.0	1.0
10.0	1.0
50.0	1.0

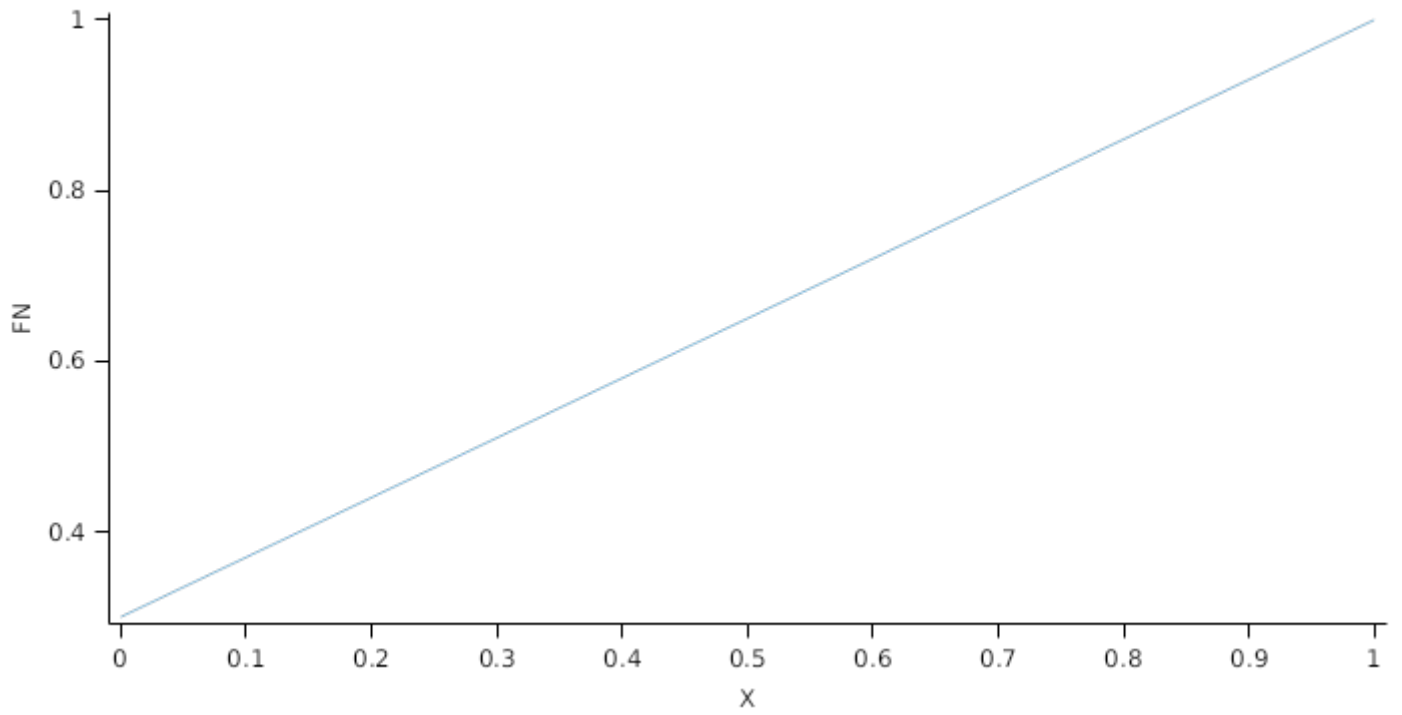
### FVPD



FN is calculated using linear interpolation

X	FN
0.0	0.3
1.0	1.0

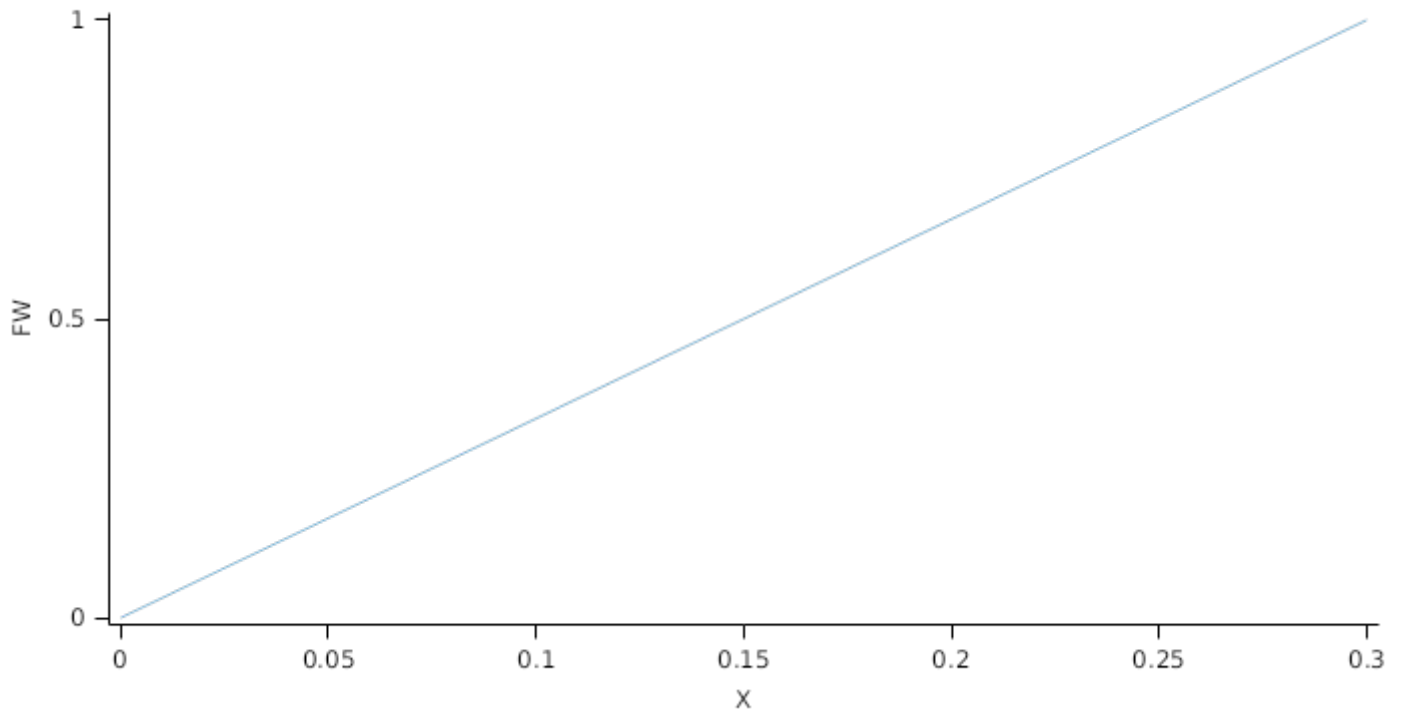
### FN



FW is calculated using linear interpolation

X	FW
0.0	0.0
0.3	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,

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

$$RadnInt = [Leaf].RadiationIntercepted$$

### 1.6.2 Constants

RemobilisationCost = 0

StructuralFraction = 0.7

DMConversionEfficiency = 1 (0-1)

CarbonConcentration = 0.4

WidthFunction = 0

### 1.6.3 CO<sub>2</sub>internal

$$CO_2internal = (163 - [IWeather].MeanT) / (5 - 0.1 \times [IWeather].MeanT)$$

### 1.6.4 StomatalConductanceCO<sub>2</sub>Modifier

$$StomatalConductanceCO_2Modifier = [Leaf].Photosynthesis.FCO_2 / RelativeCO_2Gradient$$

$$RelativeCO_2Gradient = ([IWeather].CO_2 - [Leaf].CO_2internal) / (350 - [Leaf].CO_2internal)$$

### 1.6.5 InitialLeaves[1]

Area = 380

### 1.6.6 InitialLeaves[2]

Area = 0

### 1.6.7 CohortParameters

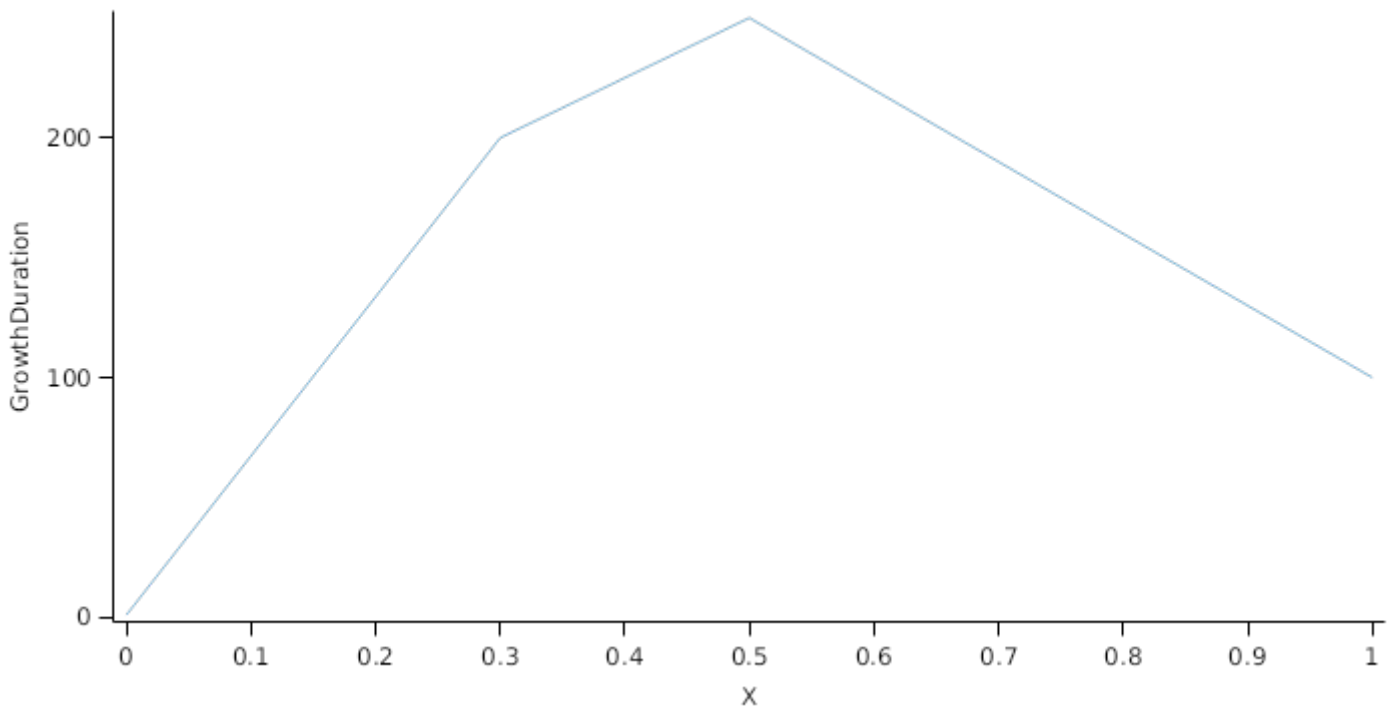
#### 1.6.7.1 Potential Leaf Area index

Leaf area index is calculated as the sum of the area of each cohort of leaves. The appearance of a new cohort of leaves occurs each time Structure.LeafTipsAppeared increases by one. From tip appearance the area of each cohort will increase for a certain number of degree days defined by the *GrowthDuration*

*GrowthDuration* is calculated using linear interpolation

X	GrowthDuration
0.0	1.0
0.3	200.0
0.5	250.0
1.0	100.0

## GrowthDuration



If no stress occurs the leaves will reach a Maximum area (*MaxArea*) at the end of the *GrowthDuration*. The *MaxArea* is defined by:

### 1.6.7.1.1 MaxArea

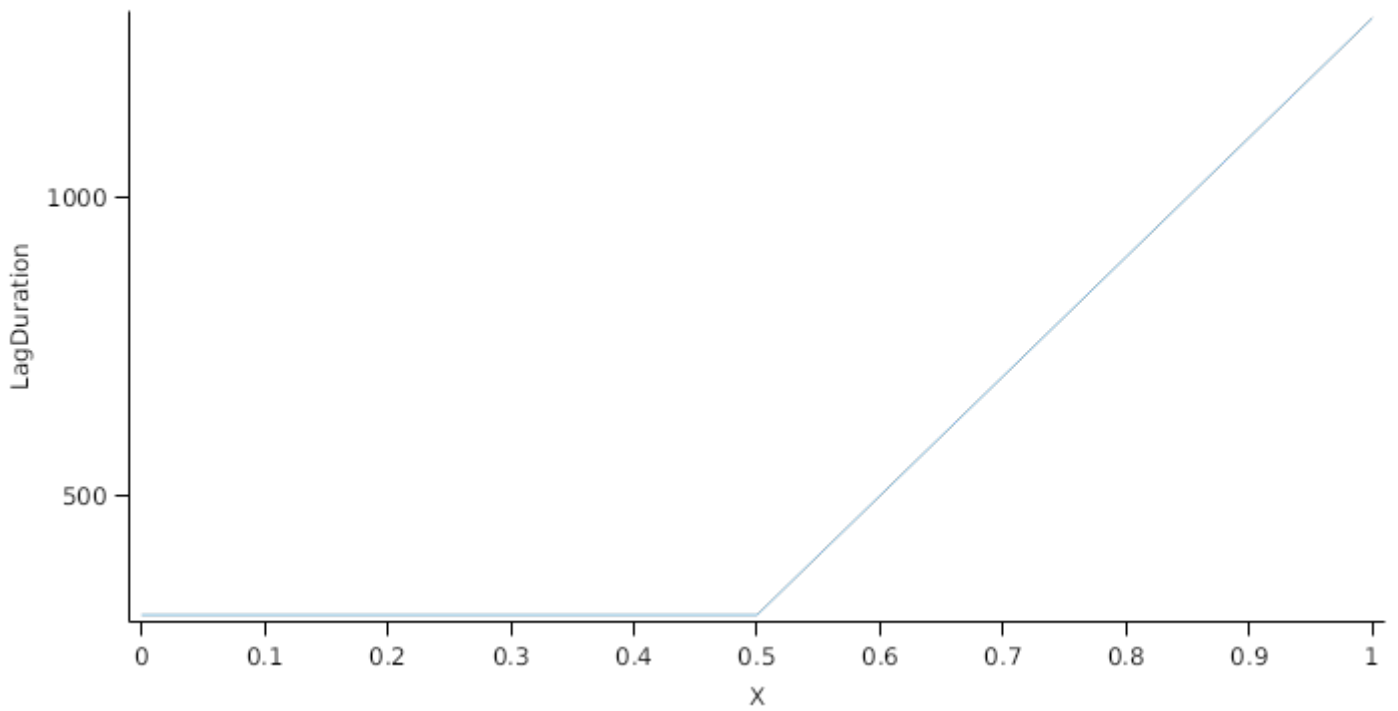
Calculates the maximum leaf size (mm<sup>2</sup>/leaf) given its node position (Elings, 2000 - Agronomy Journal 92, 436-444)

In the absence of stress the leaf will remain at *MaxArea* for a number of degree days set by the *LagDuration* and then area will senesce to zero at the end of the *SenescenceDuration*

*LagDuration* is calculated using linear interpolation

X	LagDuration
0.0	300.0
0.5	300.0
0.7	700.0
0.8	900.0
1.0	1300.0

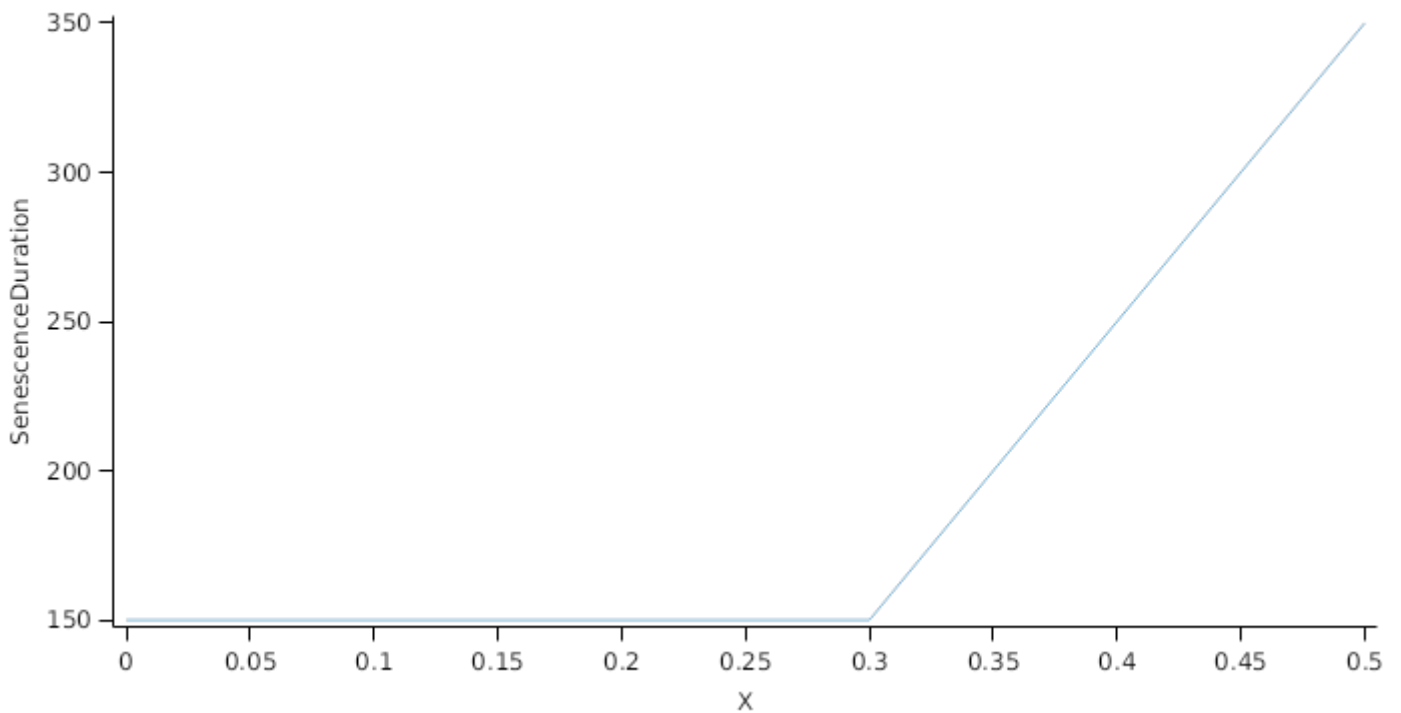
## LagDuration



*SenescenceDuration* is calculated using linear interpolation

X	SenescenceDuration
0.0	150.0
0.3	150.0
0.5	350.0

## SenescenceDuration

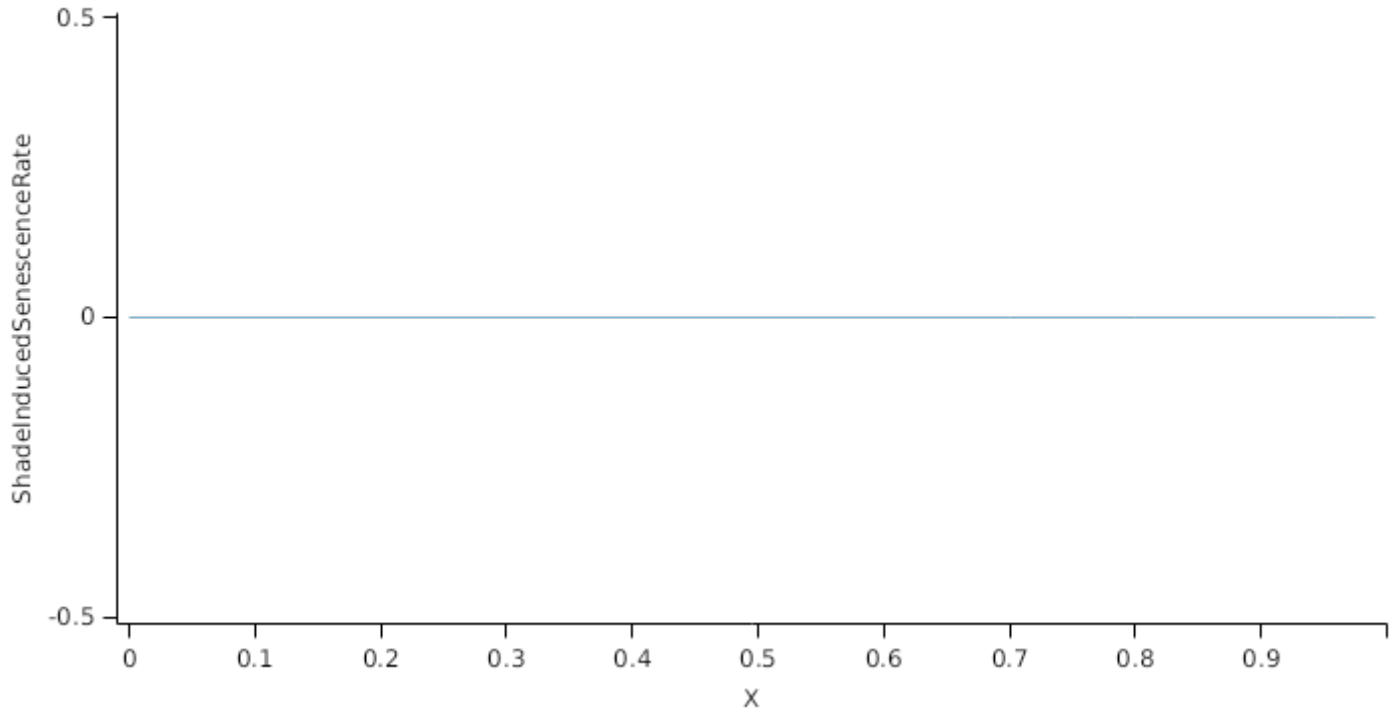


Mutual shading can cause premature senescence of cohorts if the leaf area above them becomes too great. Each cohort models the proportion of its area that is lost to shade induced senescence each day as:

*ShadeInducedSenescenceRate* is calculated using linear interpolation

X	ShadeInducedSenescenceRate
0.0	0.0
0.7	0.0
0.8	0.0
1.0	0.0
1.0	0.0

**ShadeInducedSenescenceRate**



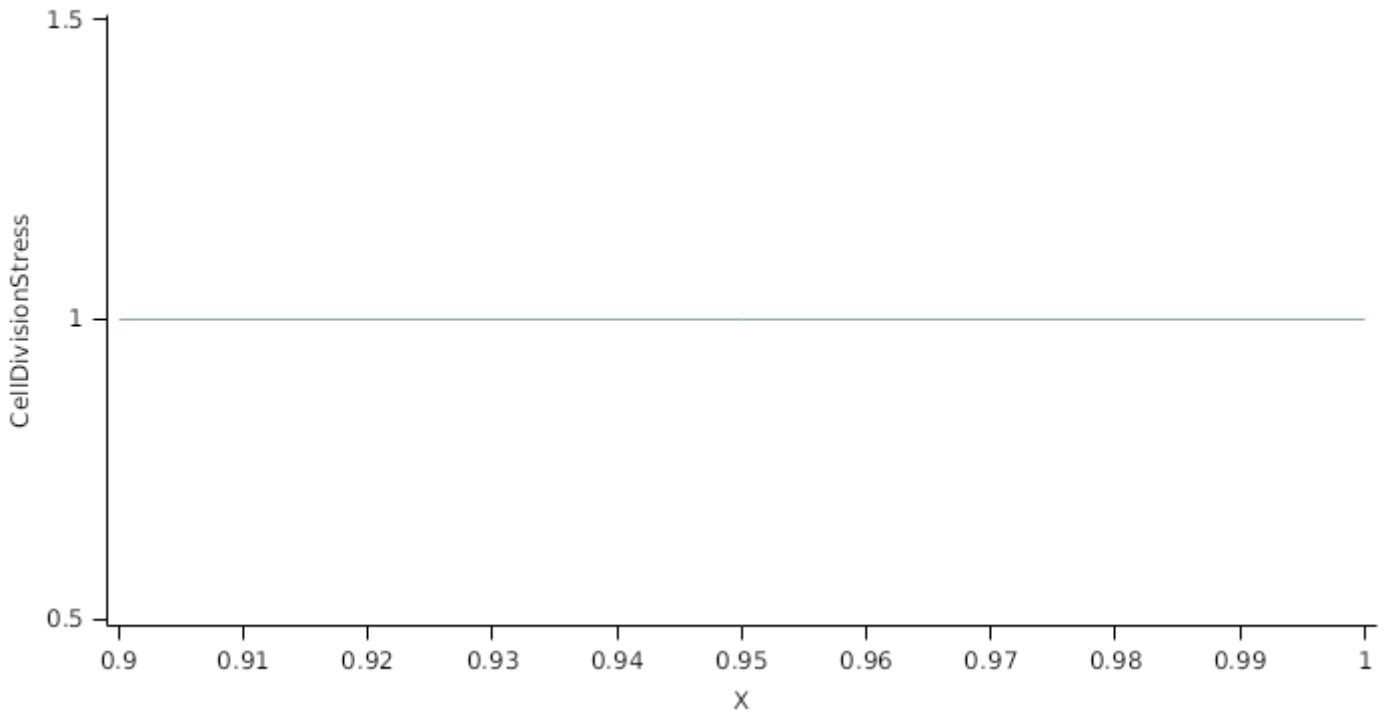
### 1.6.7.2 Stress effects on Leaf Area Index

Stress reduces leaf area in a number of ways. Firstly, stress occurring prior to the appearance of the cohort can reduce cell division, so reducing the maximum leaf size. Leaf captures this by multiplying the *MaxSize* of each cohort by a *CellDivisionStress* factor which is calculated as:

*CellDivisionStress* is calculated using linear interpolation

X	CellDivisionStress
0.9	1.0
1.0	1.0

## CellDivisionStress



Leaf.FN quantifies the N stress status of the plant and represents the concentration of metabolic N relative the maximum potential metabolic N content of the leaf calculated as  $(Leaf.NConc - MinimumNConc)/(CriticalNConc - MinimumNConc)$ .

Leaf.FW quantifies water stress and is calculated as  $Leaf.Transpiration/Leaf.WaterDemand$ , where *Leaf.Transpiration* is the minimum of *Leaf.WaterDemand* and *Root.WaterUptake*

Stress during the  $\langle i \rangle$ GrowthDuration\* of the cohort reduces the size increase of the cohort by multiplying the potential increase by a *ExpansionStress* factor:

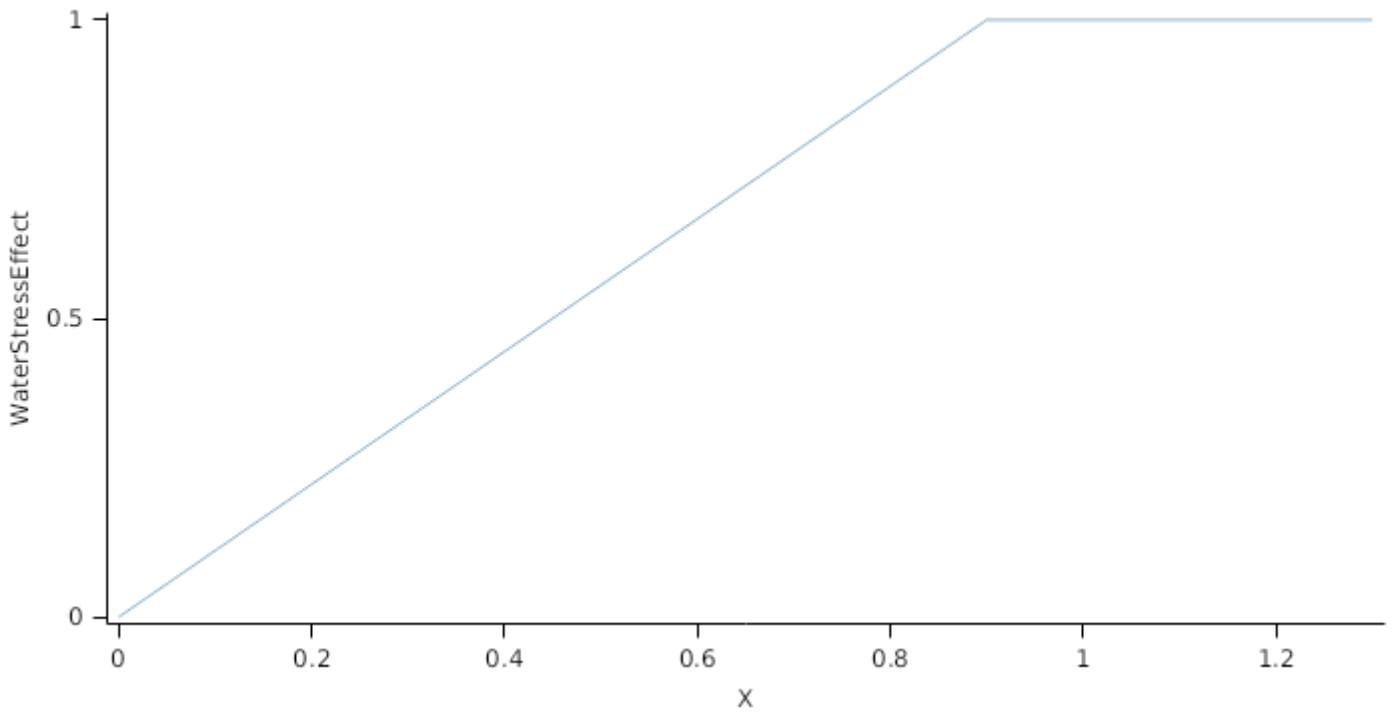
$$ExpansionStress = \text{Min}(WaterStressEffect, TemperatureEffect, NitrogenStressEffect)$$

Where:

*WaterStressEffect* is calculated using linear interpolation

X	WaterStressEffect
0.0	0.0
0.9	1.0
1.3	1.0

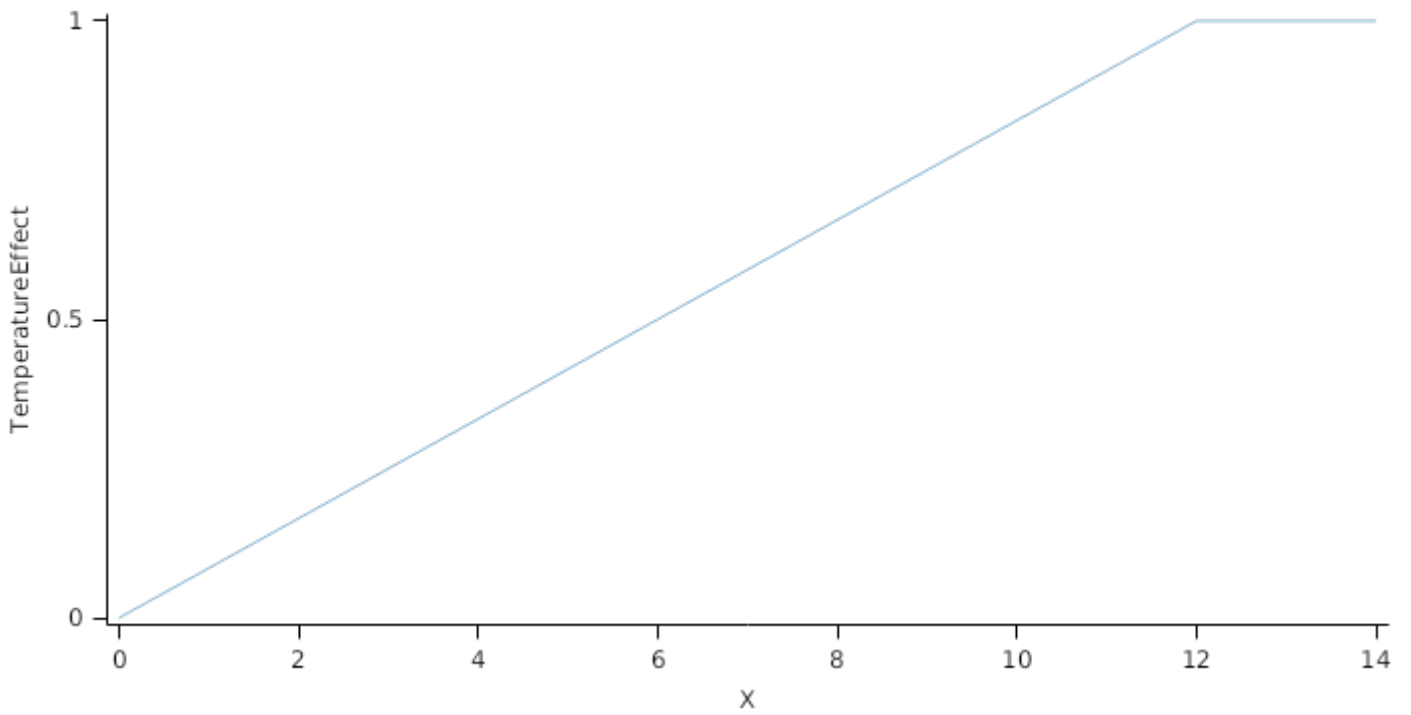
### WaterStressEffect



*TemperatureEffect* is calculated using linear interpolation

X	TemperatureEffect
0.0	0.0
12.0	1.0
14.0	1.0

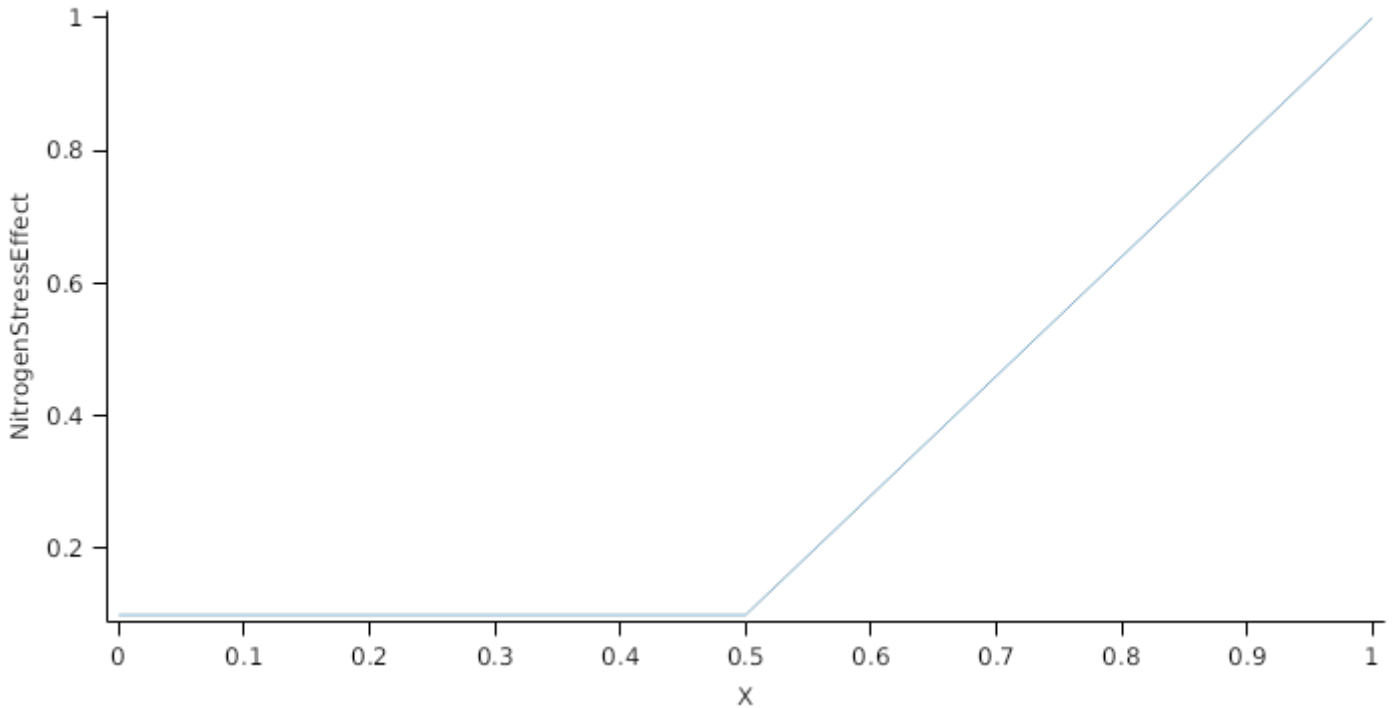
### TemperatureEffect



*NitrogenStressEffect* is calculated using linear interpolation

X	NitrogenStressEffect
0.0	0.1
0.5	0.1
1.0	1.0

## NitrogenStressEffect



Stresses can also accelerate the onset and rate of senescence in a number of ways. Nitrogen shortage will cause N to be retranslocated out of lower order leaves to support the expansion of higher order leaves and other organs. When this happens the lower order cohorts will have their area reduced in proportion to the amount of N that is remobilised out of them.

Water stress hastens senescence by increasing the rate of thermal time accumulation in the lag and senescence phases. This is done by multiplying thermal time accumulation by *DroughtInducedLagAcceleration* and *DroughtInducedSenescenceAcceleration* factors, respectively.

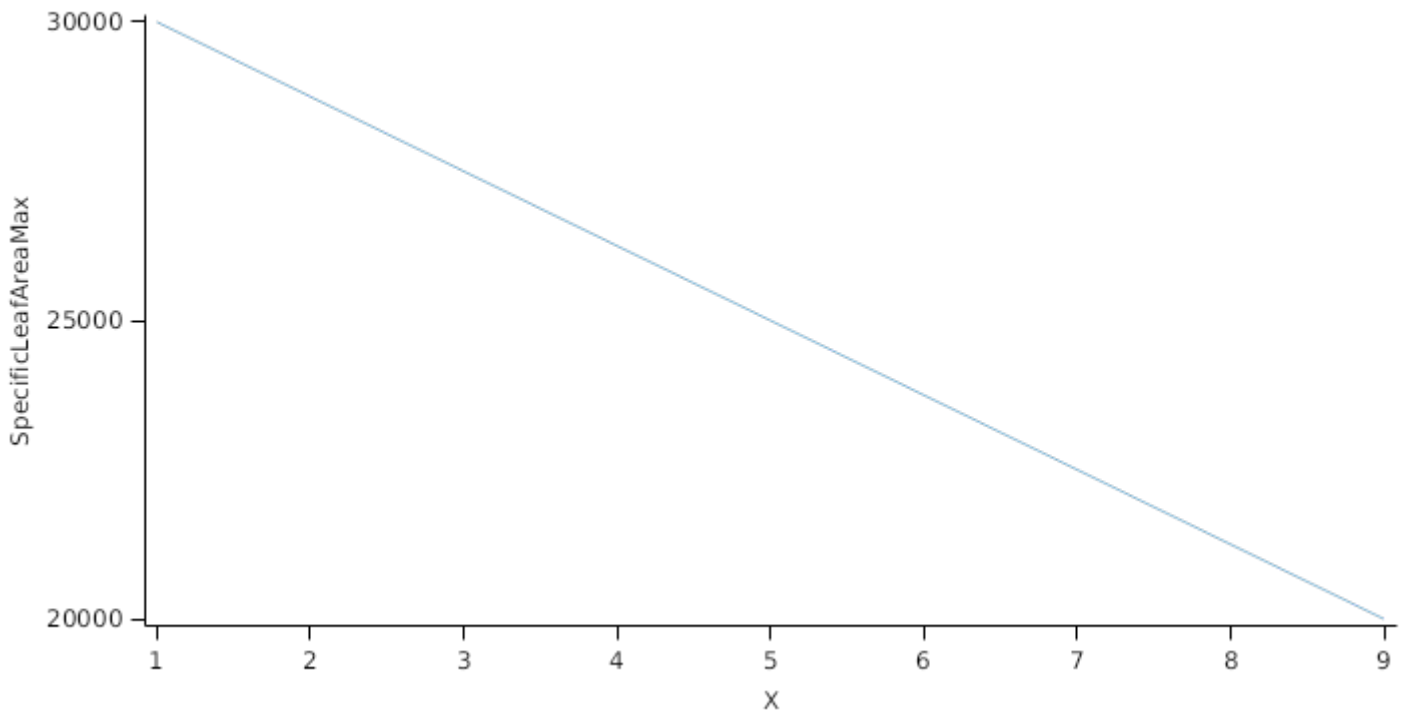
### 1.6.7.3 Dry matter Demand

Leaf calculates the DM demand from each cohort as a function of the potential size increment (*DeltaPotentialArea*) and specific leaf area bounds. Under non-stressed conditions the demand for non-storage DM is calculated as *DeltaPotentialArea* divided by the mean of *SpecificLeafAreaMax* and *SpecificLeafAreaMin*. Under stressed conditions it is calculated as *DeltaWaterConstrainedArea* divided by *SpecificLeafAreaMin*.

*SpecificLeafAreaMax* is calculated using linear interpolation

X	SpecificLeafAreaMax
1.0	30000.0
9.0	20000.0

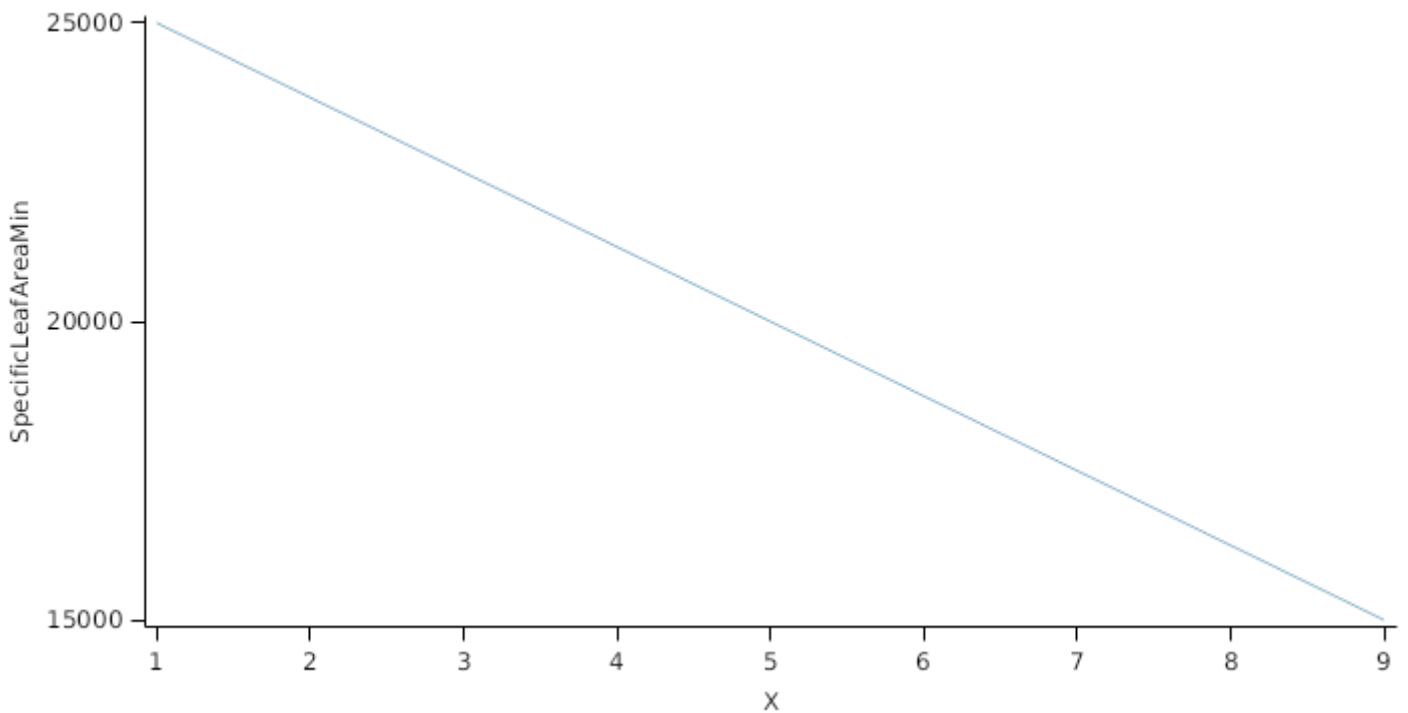
## SpecificLeafAreaMax



*SpecificLeafAreaMin* is calculated using linear interpolation

X	SpecificLeafAreaMin
1.0	25000.0
9.0	15000.0

## SpecificLeafAreaMin



Non-storage DM Demand is then separated into structural and metabolic DM demands using the *StructuralFraction*:

$StructuralFraction = 0.3$

The storage DM demand is calculated from the sum of metabolic and structural DM (including today's demands) multiplied by a *NonStructuralFraction*



### 1.6.7.4 Nitrogen Demand

Leaf calculates the N demand from each cohort as a function of the potential DM increment and N concentration bounds.

Structural N demand =  $PotentialStructuralDMAAllocation * MinimumNConc$  where:

MinimumNConc = 0.005 (g/g)

Metabolic N demand is calculated as  $PotentialMetabolicDMAAllocation * (CriticalNConc - MinimumNConc)$  where:

CriticalNConc = 0.01 (g/g)

Storage N demand is calculated as the sum of metabolic and structural wt (including today's demands) multiplied by  $LuxuryNConc (MaximumNConc - CriticalNConc)$  less the amount of storage N already present.  $MaximumNConc$  is given by:

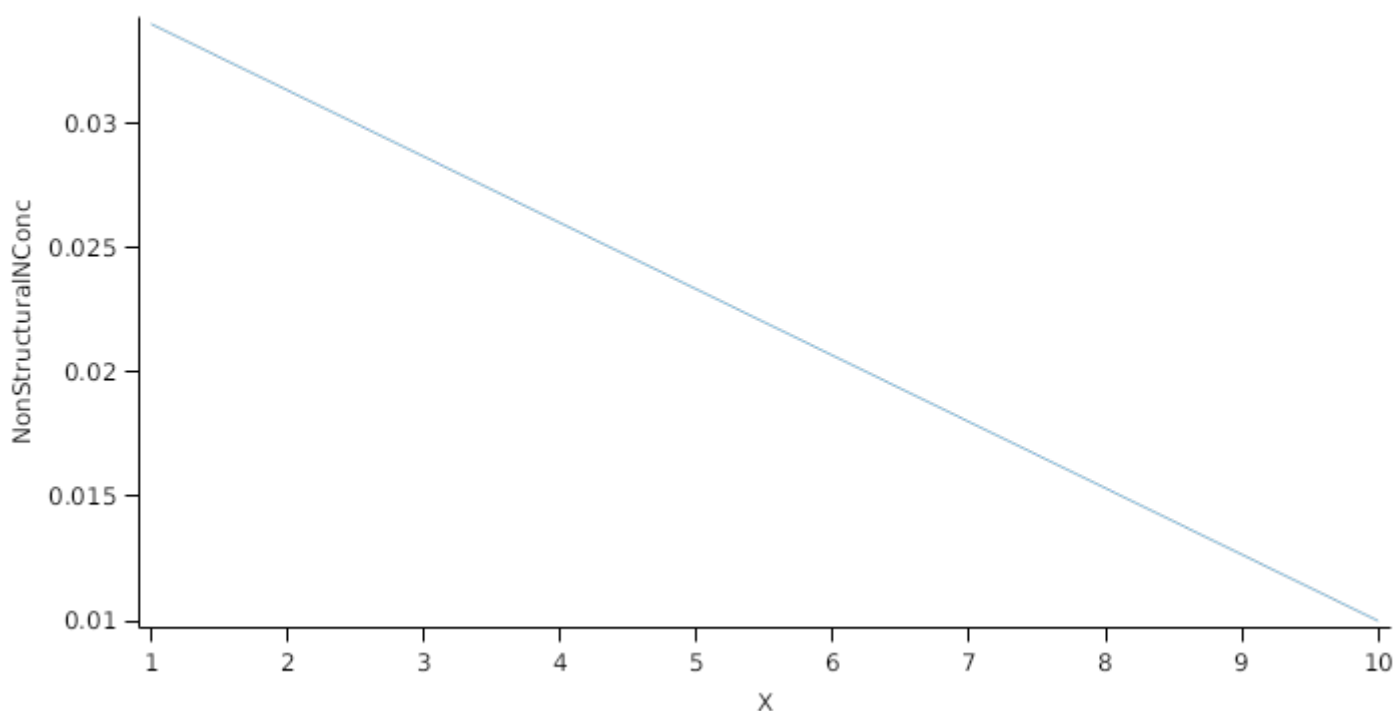
$MaximumNConc = [Leaf].CohortParameters.CriticalNConc + StorageNConc$

$StorageNConc = NonStructuralNConc * WaterStressEffect$

$NonStructuralNConc$  is calculated using linear interpolation

X	NonStructuralNConc
1.0	0.0
10.0	0.0

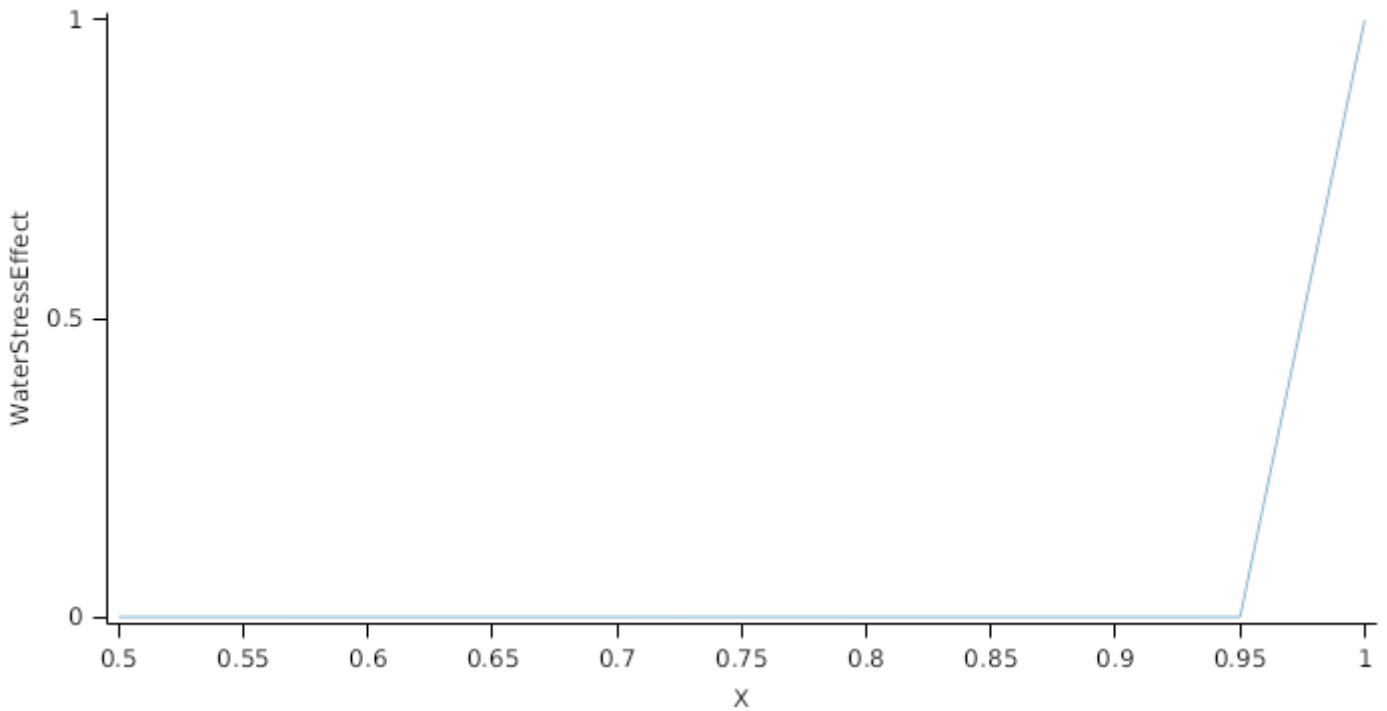
### NonStructuralNConc



$WaterStressEffect$  is calculated using linear interpolation

X	WaterStressEffect
0.5	0.0
0.9	0.0
1.0	1.0

## WaterStressEffect



### 1.6.7.5 Drymatter supply

In addition to photosynthesis, the leaf can also supply DM by reallocation of senescing DM and retranslocation of storage DM: Reallocation supply is a proportion of the metabolic and non-structural DM that would be senesced each day where the proportion is set by:

DMReallocationFactor = 1

Retranslocation supply is calculated as a proportion of the amount of storage DM in each cohort where the proportion is set by :

DMRetranslocationFactor = 0.5

### 1.6.7.6 Nitrogen supply

Nitrogen supply from the leaf comes from the reallocation of metabolic and storage N in senescing material and the retranslocation of metabolic and storage N. Reallocation supply is a proportion of the Metabolic and Storage DM that would be senesced each day where the proportion is set by:

NReallocationFactor = 1

Retranslocation supply is calculated as a proportion of the amount of storage and metabolic N in each cohort where the proportion is set by :

NRetranslocationFactor = 0

### 1.6.7.7 Constants

DMReallocationFactor = 1

RemobilisationCost = 0

LeafSizeShapeParameter = 0.01

MaintenanceRespirationFunction = 0

SenescingLeafRelativeSize = 1 (0-1)

NReallocationFactor = 1

NRetranslocationFactor = 0

DMRetranslocationFactor = 0.5

DetachmentLagDuration = 1000000

DetachmentDuration = 1000000

CriticalNConc = 0.01 (g/g)

MinimumNConc = 0.005 (g/g)

StructuralFraction = 0.3

StorageFraction = 0.1

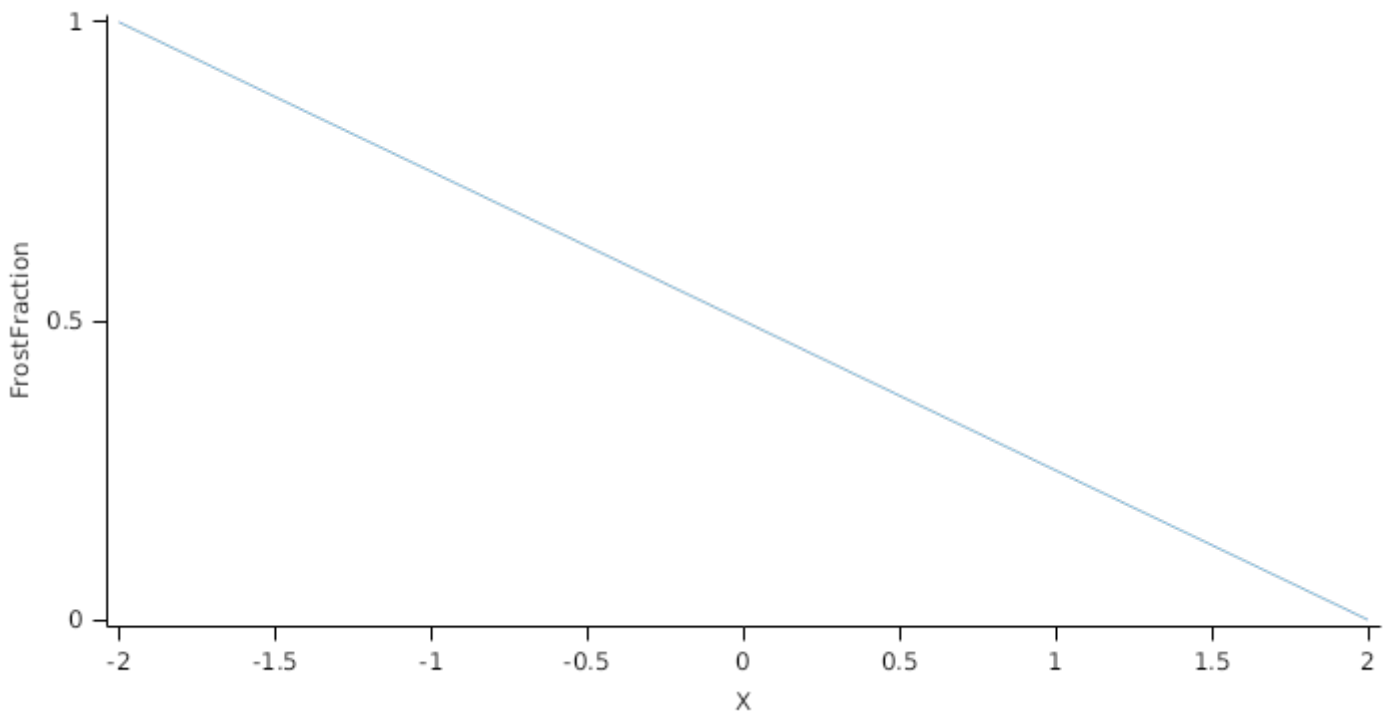
InitialNConc = 0 (g/g)

### 1.6.8 FrostFraction

*FrostFraction* is calculated using linear interpolation

X	FrostFraction
-2.0	1.0
2.0	0.0

### FrostFraction



### 1.6.9 ThermalTime

*ThermalTime* = [Phenology].ThermalTime

### 1.6.10 FRGRFunction

*FRGRFunction* = Min(*FT*, *FN*, *FVPD*)

Where:

*FT* = [Leaf].Photosynthesis.*FT*

*FN* = [Leaf].Photosynthesis.*FN*

*FVPD* = [Leaf].Photosynthesis.*FVPD*

### 1.6.11 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	30	0
Cut	80	0	0	0
Prune	0	0	60	0
Graze	60	0	10	0

### 1.6.12 DepthFunction

$DepthFunction = [Leaf].Height$

### 1.6.13 DMDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

### 1.6.14 NDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

## 1.7 Husk

### 1.7.1 Husk

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.

$Structural = DMDemandFunction \times StructuralFraction$

Demand is calculated from the product of growth rate, thermal time and population.

$ThermalTime = [Phenology].ThermalTime$

StartStage = 5.5

GrowthDuration = 300

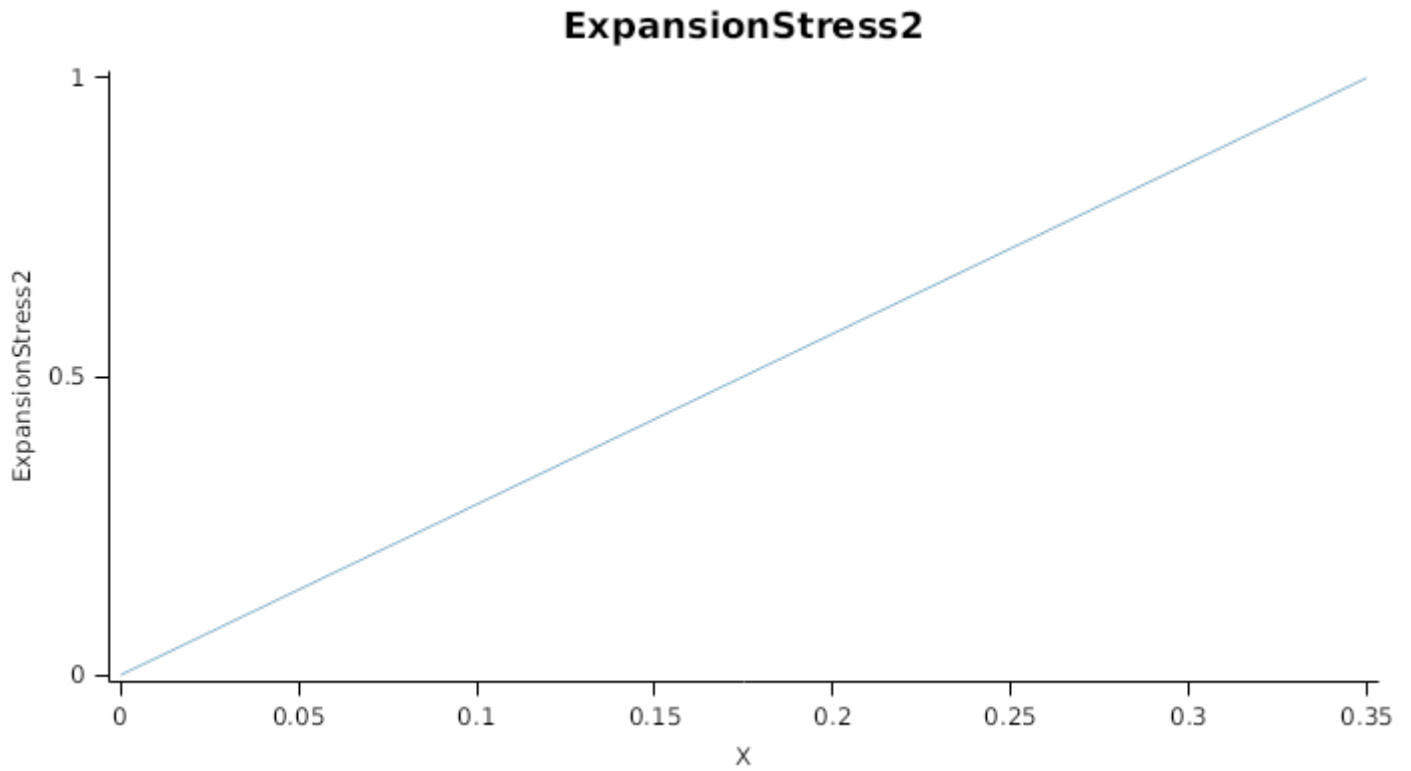
MaximumOrganWt = 25

$OrganPopulation = [Rachis].CobNumber$

SenescenceRate = 0 (/d)

*ExpansionStress2* is calculated using linear interpolation

X	ExpansionStress2
0.0	0.0
0.3	1.0



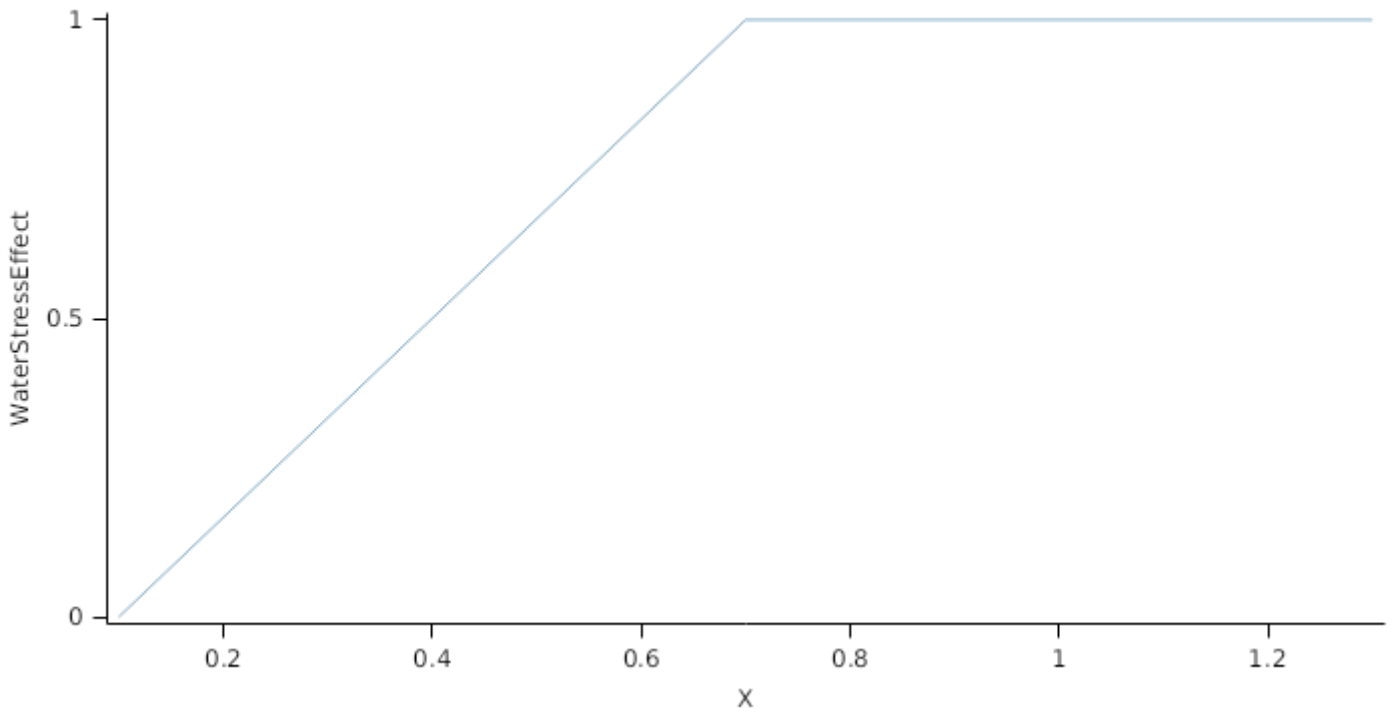
$ExpansionStress = \text{Min}(WaterStressEffect, TemperatureEffect, NitrogenStressEffect)$

Where:

*WaterStressEffect* is calculated using linear interpolation

X	WaterStressEffect
0.1	0.0
0.7	1.0
1.3	1.0

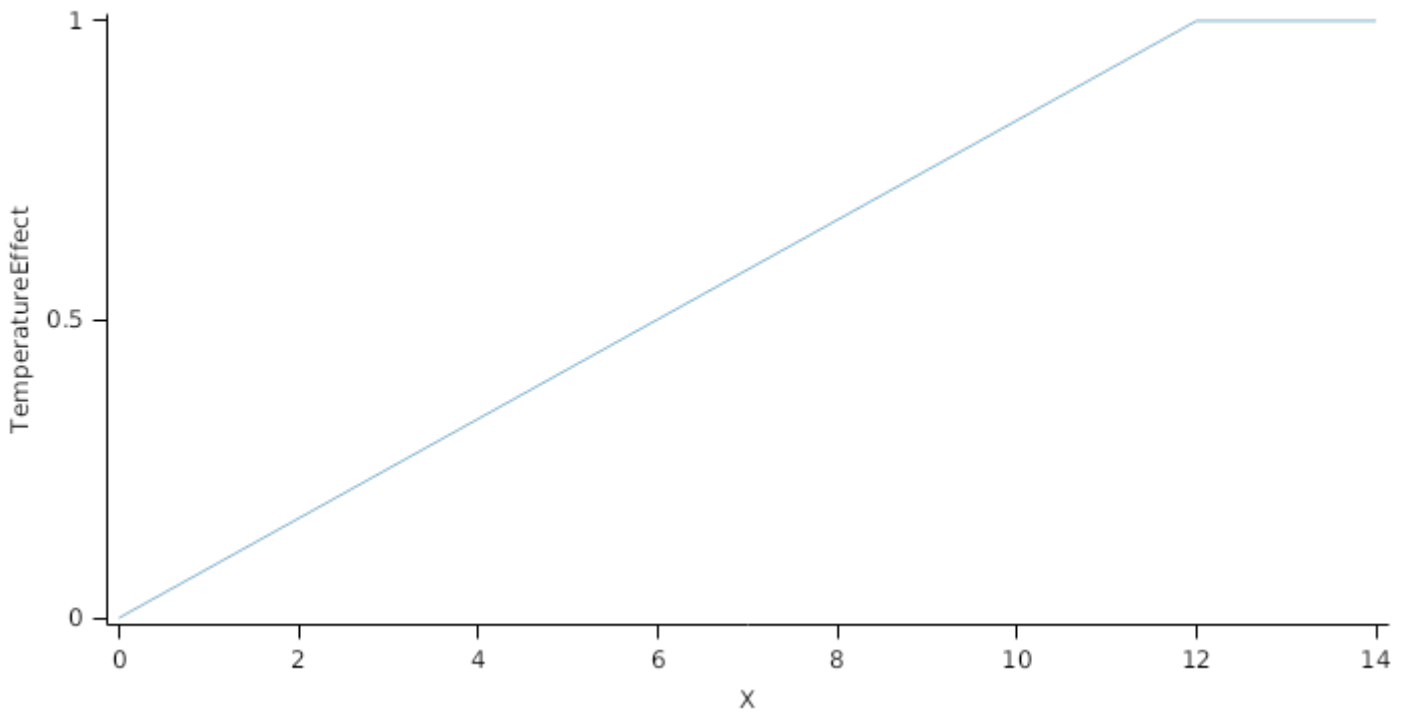
### WaterStressEffect



*TemperatureEffect* is calculated using linear interpolation

X	TemperatureEffect
0.0	0.0
12.0	1.0
14.0	1.0

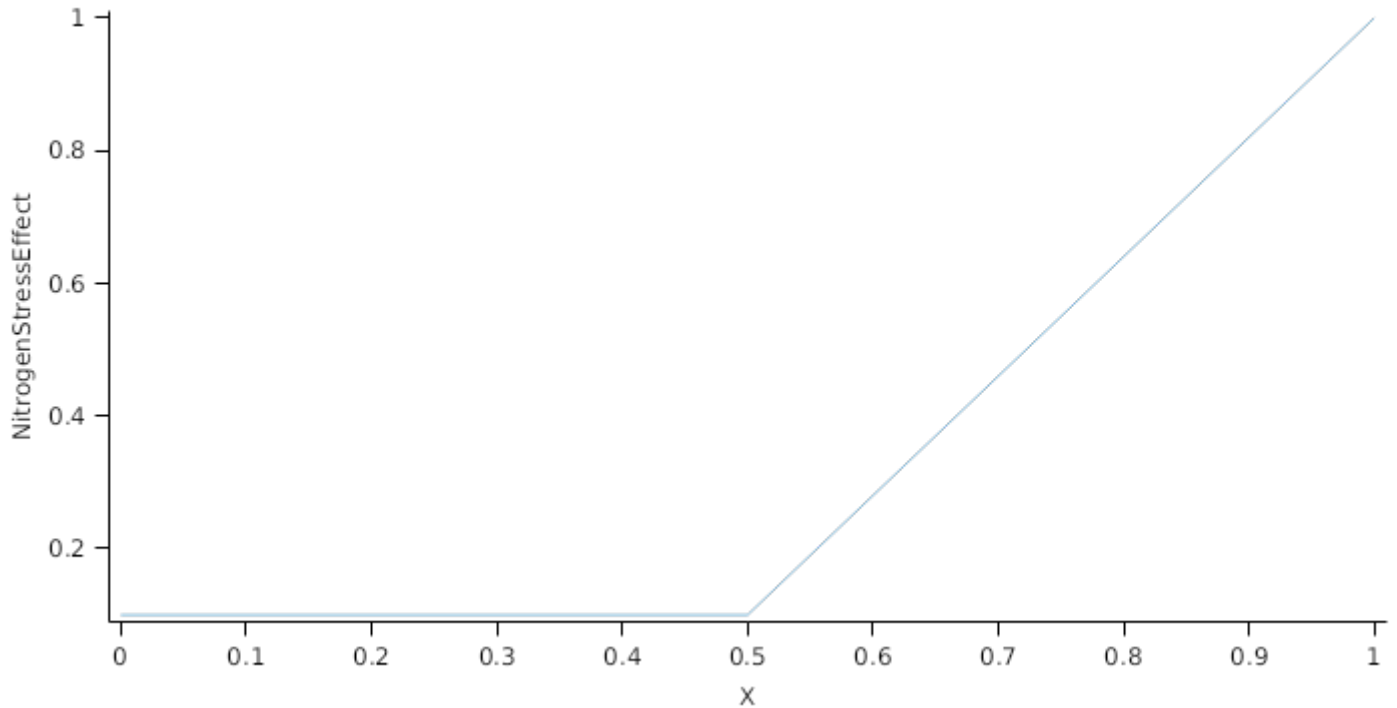
### TemperatureEffect



*NitrogenStressEffect* is calculated using linear interpolation

X	NitrogenStressEffect
0.0	0.1
0.5	0.1
1.0	1.0

## NitrogenStressEffect



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

PreStemElongation has a value between Emergence and StartGrainFill calculated as:

StemFraction = 0.5

StemElongation has a value between StartGrainFill and HarvestRipe calculated as:

StemFraction = 1

Metabolic = 0

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

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

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.

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

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

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

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

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

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

$NitrogenDemandSwitch = [Husk].nitrogenDemandSwitch$

$MaxNconc = [Husk].maximumNconc$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

#### 1.7.4 N Concentration Thresholds

MinimumNConc = 0.005 (g/g)

$CriticalNConc = [Husk].MinimumNConc$

MaximumNConc = 0.007 (g/g)

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

NitrogenDemandSwitch has a value between Emergence and StartGrainFill calculated as:

Constant = 1

#### 1.7.5 Dry Matter Supply

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

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

#### 1.7.6 Nitrogen Supply

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

Husk can retranslocate up to 10% of non-structural N each day if required by the plant arbitrator to meet N demands.

#### 1.7.7 Senescence and Detachment

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

Husk 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	50	0	10	0
Cut	80	0	0	0
Prune	0	0	60	0
Graze	60	0	20	0

### 1.8 Rachis

#### 1.8.1 Rachis



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.

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

Demand is calculated from the product of growth rate, thermal time and population.

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

$$\text{StartStage} = 5.5$$

$$\text{GrowthDuration} = 300$$

$$\text{MaximumOrganWt} = 23$$

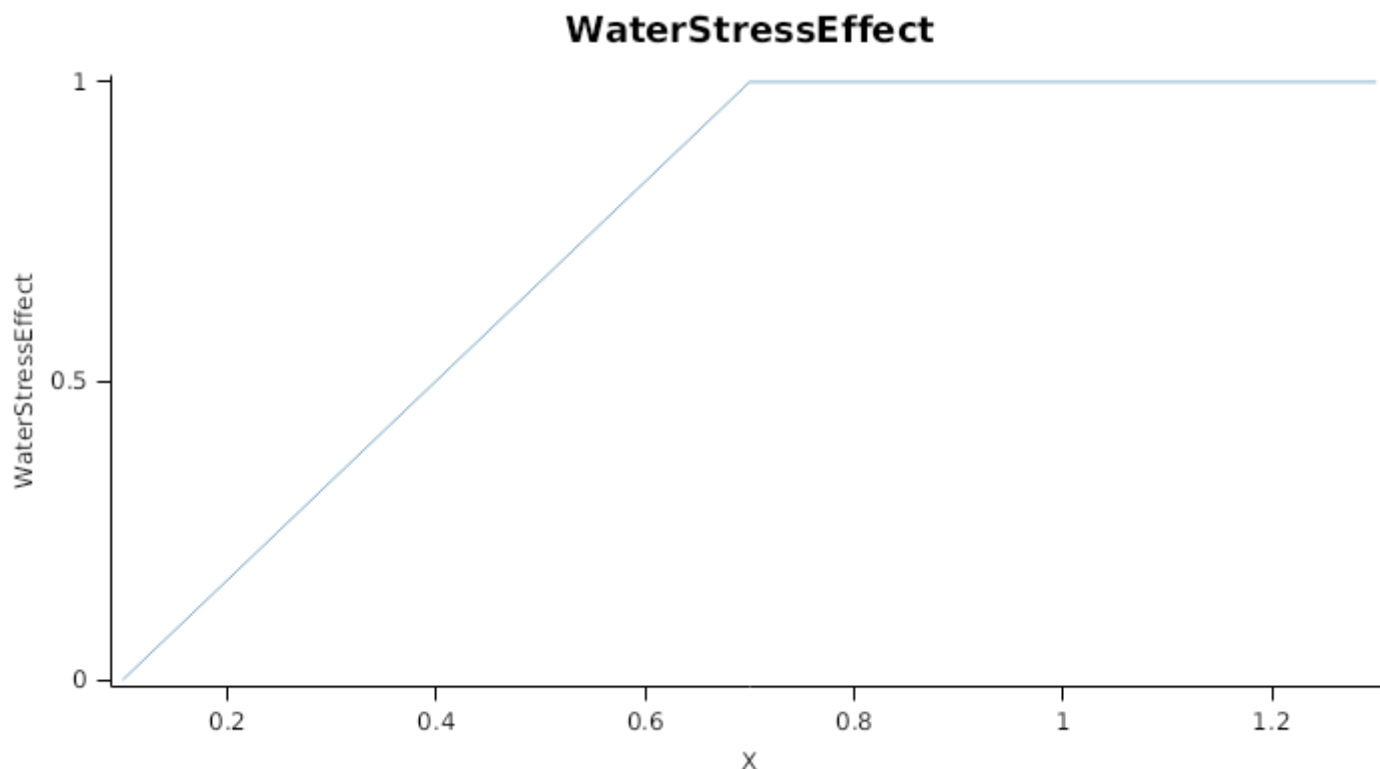
$$\text{OrganPopulation} = [\text{Rachis}].\text{CobNumber}$$

$$\text{ExpansionStress} = \text{Min}(\text{WaterStressEffect}, \text{TemperatureEffect}, \text{NitrogenStressEffect})$$

Where:

*WaterStressEffect* is calculated using linear interpolation

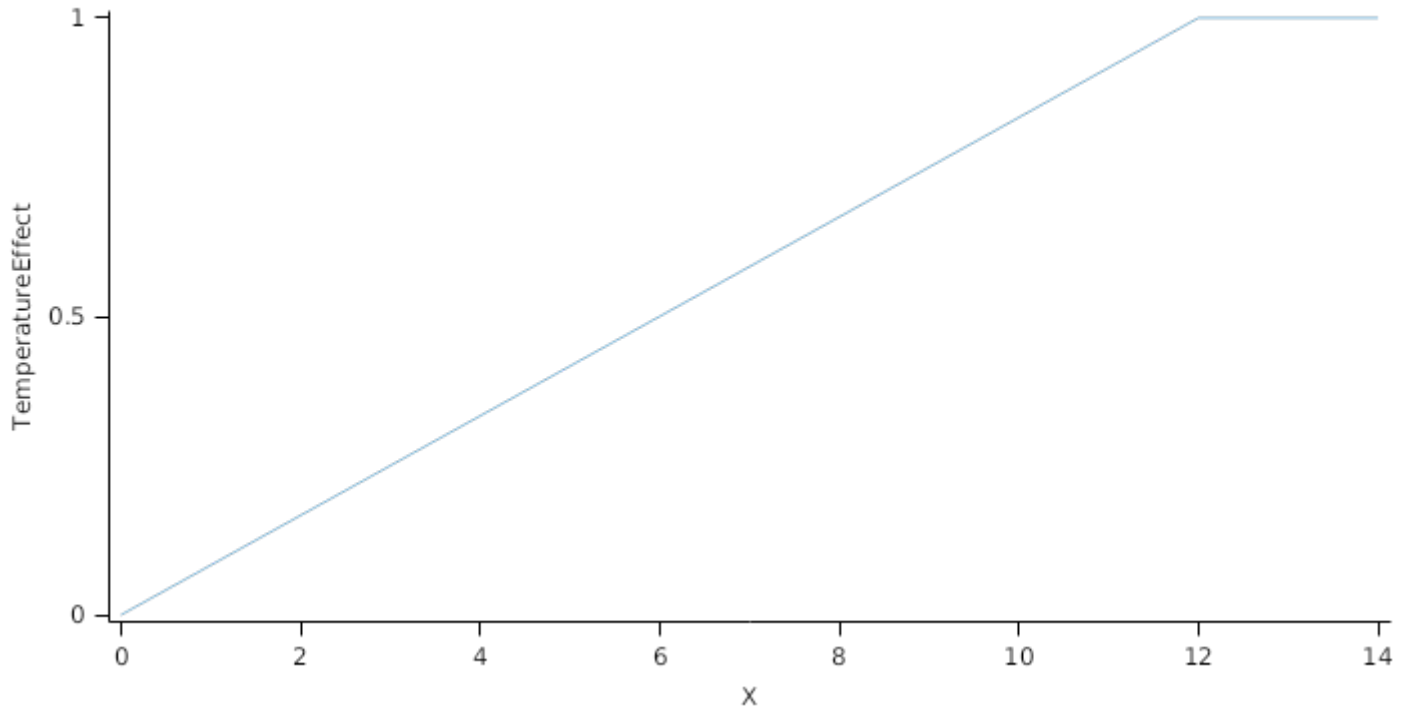
X	WaterStressEffect
0.1	0.0
0.7	1.0
1.3	1.0



*TemperatureEffect* is calculated using linear interpolation

X	TemperatureEffect
0.0	0.0
12.0	1.0
14.0	1.0

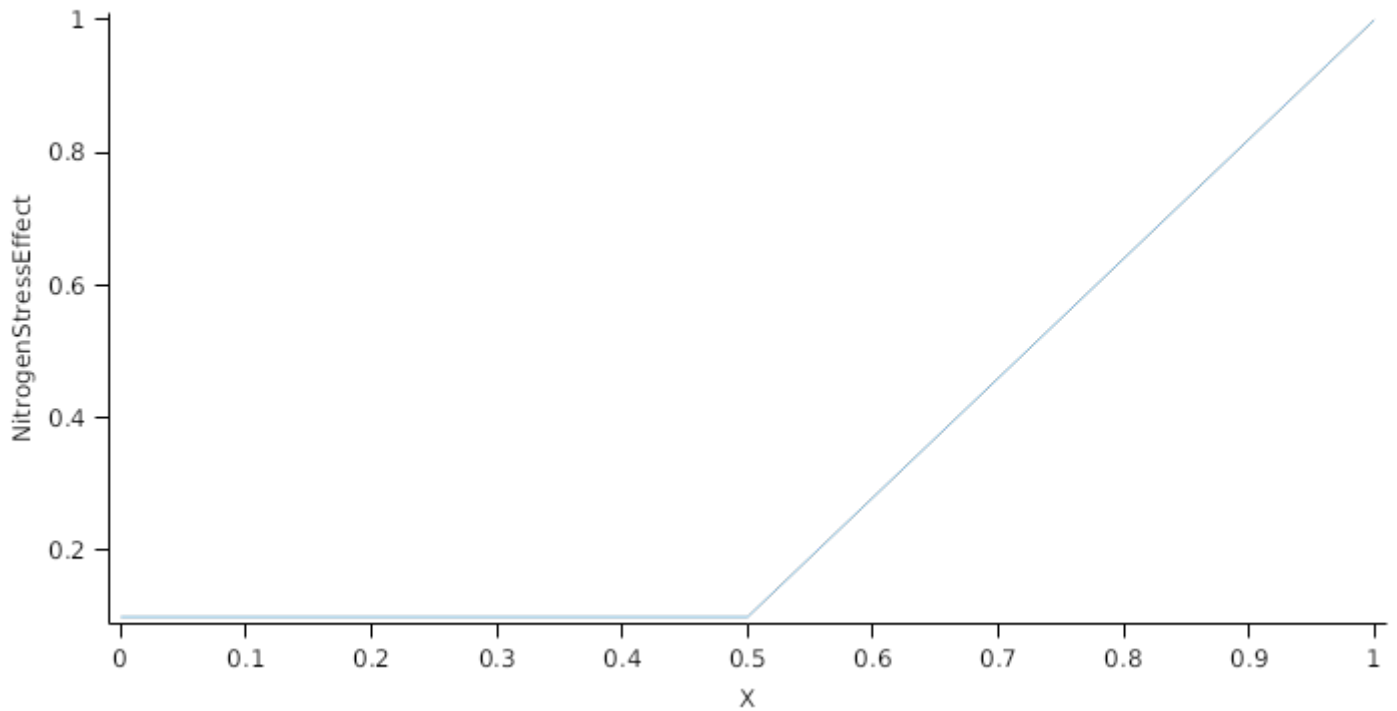
**TemperatureEffect**



*NitrogenStressEffect* is calculated using linear interpolation

X	NitrogenStressEffect
0.0	0.1
0.5	0.1
1.0	1.0

## NitrogenStressEffect



StructuralFraction = 1 (g/g)

Metabolic = 0

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

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

QStructuralPriority = 1

QMetabolicPriority = 1

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.

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

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

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

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

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

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

$NitrogenDemandSwitch = [Rachis].nitrogenDemandSwitch$

$MaxNconc = [Rachis].maximumNconc$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

#### 1.8.4 N Concentration Thresholds

MinimumNConc = 0.003 (g/g)

$CriticalNConc = [Rachis].MinimumNConc$

MaximumNConc = 0.008 (g/g)

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

NitrogenDemandSwitch has a value between Emergence and StartGrainFill calculated as:

Constant = 1

#### 1.8.5 Dry Matter Supply

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

Rachis does not retranslocate non-structural DM.

#### 1.8.6 Nitrogen Supply

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

Rachis can retranslocate up to 10% of non-structural N each day if required by the plant arbitrator to meet N demands.

#### 1.8.7 Senescence and Detachment

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

Rachis 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	50	0	10	0
Cut	80	0	0	0
Prune	0	0	60	0
Graze	60	0	20	0

### 1.9 Stem

#### 1.9.1 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.

#### 1.9.2 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.

##### 1.9.2.1 DMDemands

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

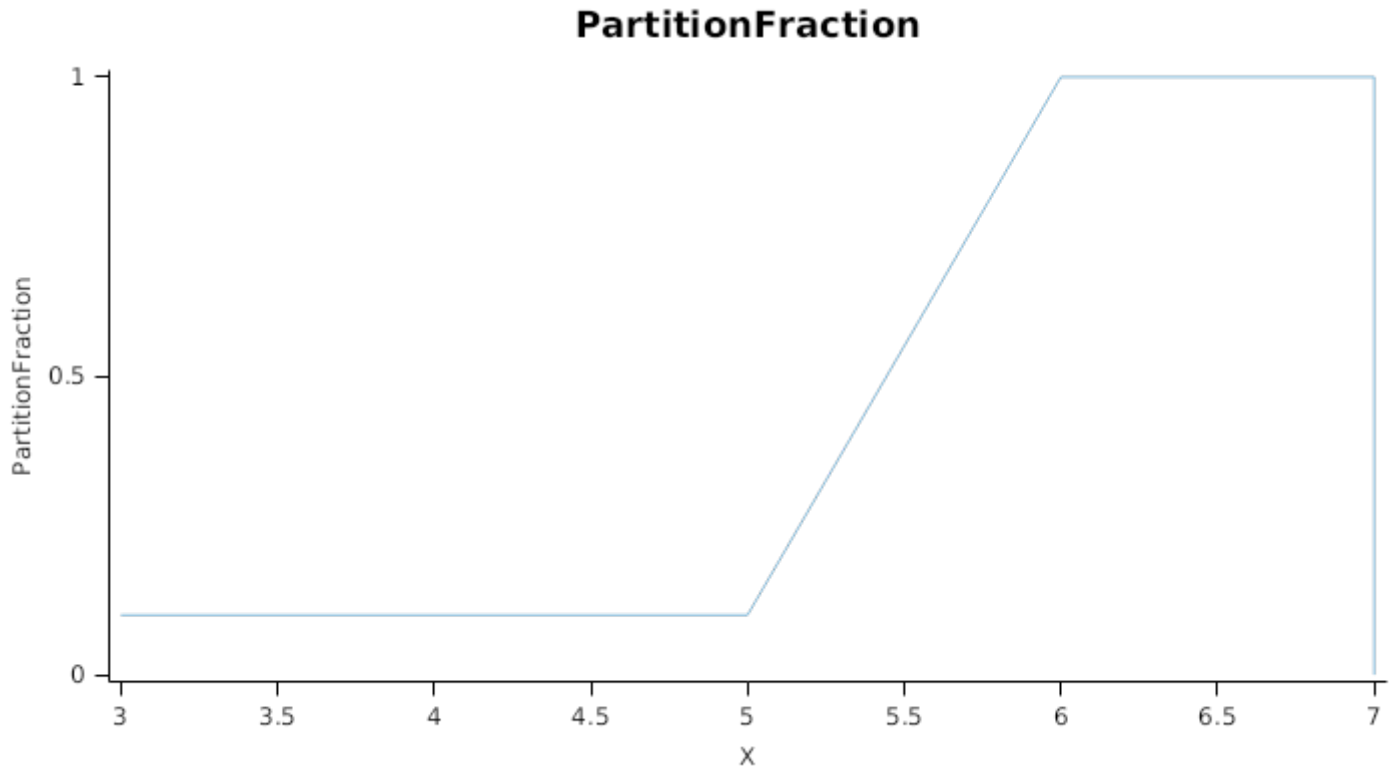
$Structural = DMDemandFunction \times StructuralFraction$

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 linear interpolation

X	PartitionFraction
3.0	0.1
5.0	0.1
6.0	1.0
7.0	1.0
7.0	0.0



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

PreStemElongation has a value between Emergence and StartGrainFill calculated as:

StemFraction = 0.5

StemElongation has a value between StartGrainFill and HarvestRipe calculated as:

StemFraction = 1

Metabolic = 0

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

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

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

### 1.9.3 Nitrogen Demand

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

#### 1.9.3.1 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.

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

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

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

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

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

#### 1.9.4 N Concentration Thresholds

$$\text{MinimumNConc} = 0.004 \text{ (g/g)}$$

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

$$\text{MetabolicNConc} = \text{NonStructuralNConc} \times \text{Proportion}$$

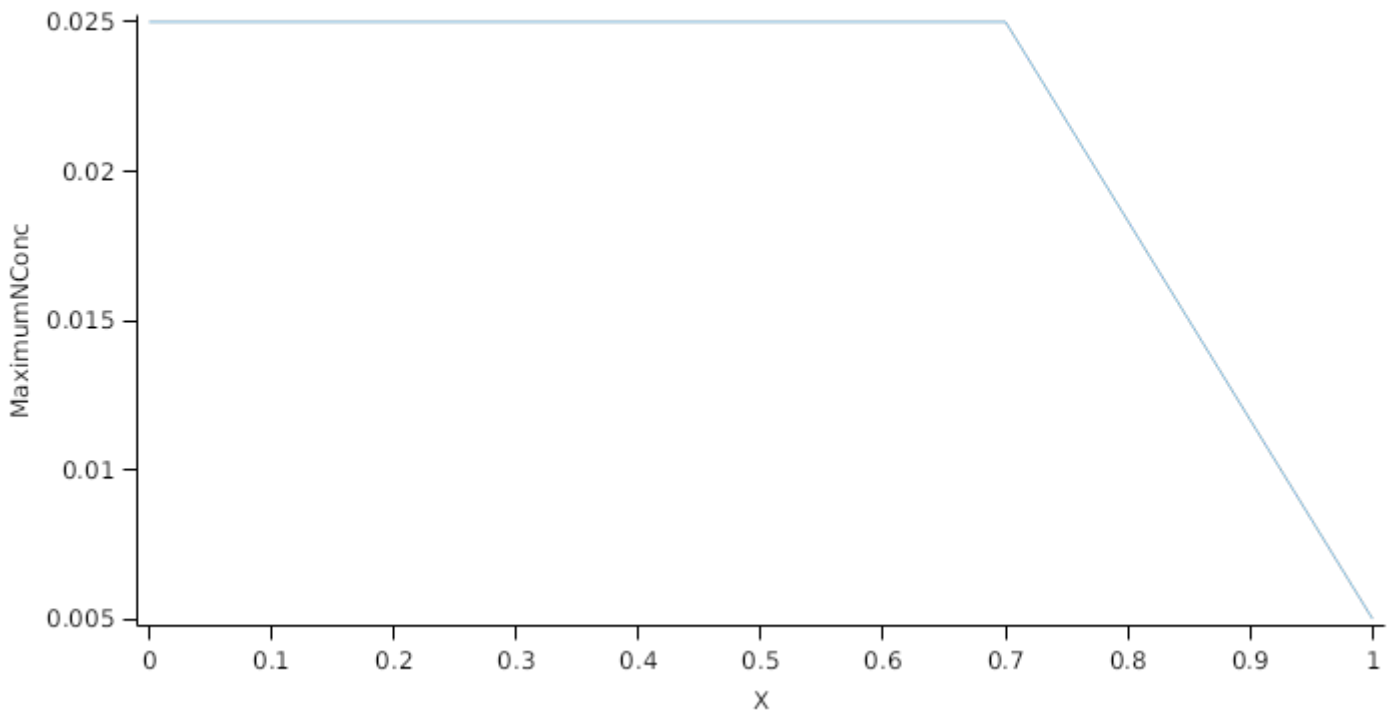
$$\text{NonStructuralNConc} = [\text{Stem}].\text{MaximumNConc} - [\text{Stem}].\text{MinimumNConc}$$

$$\text{Proportion} = 0.2$$

*MaximumNConc* is calculated using linear interpolation

X	MaximumNConc
0.0	0.0
0.7	0.0
1.0	0.0

## MaximumNConc



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

NitrogenDemandSwitch has a value between Emergence and StartGrainFill calculated as:

Constant = 1

### 1.9.5 Dry Matter Supply

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

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

### 1.9.6 Nitrogen Supply

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

The proportion of non-structural N that is allocated each day is quantified by the NReallocationFactor.

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

PreStemElongation has a value between Emergence and FlagLeaf calculated as:

StemFraction = 0

StemElongation has a value between FlagLeaf and HarvestRipe calculated as:

NRetranslocationFactor = 0.05

### 1.9.7 Senescence and Detachment

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

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	50	0	10	0
Cut	80	0	0	0
Prune	0	0	60	0
Graze	60	0	20	0

## 1.10 AboveGround

### 1.10.1 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
- \* Husk
- \* Grain
- \* Rachis

## 1.11 BelowGround

### 1.11.1 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

## 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
- \* Husk
- \* Grain
- \* Rachis
- \* Root

## 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
- \* Husk
- \* Grain
- \* Rachis
- \* Root

## 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
- \* Husk
- \* Grain
- \* Rachis
- \* Root

## **1.15 EarLive**

### **1.15.1 EarLive**

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

EarLive summarises the following biomass objects:

- \* Husk
- \* Grain
- \* Rachis

## **1.16 AboveGroundLive**

### **1.16.1 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
- \* Husk
- \* Grain
- \* Rachis

## **1.17 AboveGroundDead**

### **1.17.1 AboveGroundDead**

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

AboveGroundDead summarises the following biomass objects:

- \* Leaf
- \* Stem
- \* Husk
- \* Grain
- \* Rachis

## **1.18 Future Development Needs**

#Future Development requirements

- \* Phenology not responding to stress events. Requires better data sets for quantifying the effects of water and nitrogen on leaf appearance and the timing of development stages.
- \* Phosphorus response
- \* Improved parameterisation of supply and demand for N and DM from organs and arbitration of these resources.
- \* Heat Stress responses in grain number
- \* More validation under a wider range of environments with more detailed datasets
- \* Seedling mortality
- \* Water demand (MicroClimate) needs validation

## **1.19 CultivarFolder**

### **1.19.1 Hycorn\_53**

Hycorn\_53 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 190

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 520

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 25

[Grain].MaximumGrainsPerCob.FixedValue = 550

### **1.19.2 Pioneer\_33M54**

Pioneer\_33M54 overrides the following properties:

[Phenology].GrainFilling.Target.FixedValue = 550

[Phenology].Juvenile.Target.FixedValue = 270

### **1.19.3 Pioneer\_38H20**

Pioneer\_38H20 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 210

[Phenology].GrainFilling.Target.FixedValue = 550

### **1.19.4 Pioneer\_34K77**

Pioneer\_34K77 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 280

[Phenology].GrainFilling.Target.FixedValue = 550

### **1.19.5 Pioneer\_39V43**

Pioneer\_39V43 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 100

[Phenology].GrainFilling.Target.FixedValue = 500

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.X = 1, 4, 4.2, 10.5, 11

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.Y = 26, 26, 46, 46, 70

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 18

[Structure].HeightModel.PotentialHeight.XYPairs.X = 3, 4, 5, 6

[Structure].HeightModel.PotentialHeight.XYPairs.Y = 20, 40, 50, 1300

### **1.19.6 Atrium**

Atrium overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 150

[Phenology].GrainFilling.Target.FixedValue = 500

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.X = 1, 4, 4.2, 10.5, 11

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.Y = 26, 26, 46, 46, 70

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 18

[Structure].HeightModel.PotentialHeight.XYPairs.X = 3, 4, 5, 6

[Structure].HeightModel.PotentialHeight.XYPairs.Y = 20, 40, 50, 2370

### **1.19.7 Laila**

Laila overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 211

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 0

[Phenology].FlagLeafToFlowering.Target.FixedValue = 1

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 812

[Phenology].Maturing.Target.FixedValue = 1

[Phenology].MaturityToHarvestRipe.Target.FixedValue = 1

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 36

[Grain].MaximumGrainsPerCob.FixedValue = 720

### **1.19.8 GH\_5019WX**

GH\_5019WX overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 225

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24.0

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124

[Phenology].FlagLeafToFlowering.Target.FixedValue = 101

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 694

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 25

[Grain].MaximumGrainsPerCob.FixedValue = 550

### **1.19.9 Hycorn\_40**

Hycorn\_40 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 190

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124

[Phenology].FlagLeafToFlowering.Target.FixedValue = 50

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 713

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 25

[Grain].MaximumGrainsPerCob.FixedValue = 550

### **1.19.10 GH\_5009**

GH\_5009 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 190

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 300

[Phenology].FlagLeafToFlowering.Target.FixedValue = 50

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 713

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 25

[Grain].MaximumGrainsPerCob.FixedValue = 550

#### **1.19.11 Dekalb\_XL82**

Dekalb\_XL82 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 211

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 472

[Phenology].FlagLeafToFlowering.Target.FixedValue = 101

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 813

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 25

[Grain].MaximumGrainsPerCob.FixedValue = 550

#### **1.19.12 malawi\_local**

malawi\_local overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 280

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 600

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 360

[Grain].MaximumPotentialGrainSize.FixedValue = 0.217

#### **1.19.13 mh19**

mh19 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 290

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 570

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.14 mh17**

mh17 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 290

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 570

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.15 mh16**

mh16 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 290

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 570

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.16 mh12**

mh12 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 290

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 570

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.17 sc623**

sc623 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 280

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.18 sc625**

sc625 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 280

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 570

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 504

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.19 sc601**

sc601 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 280

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 560

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 495

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.20 CG4141**

CG4141 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 233

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 468

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

### **1.19.21 mh18**

mh18 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 260

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 486

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

### **1.19.22 r215**

r215 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 250

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 468

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

### **1.19.23 Melkassa**

Melkassa overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 230

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 160

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 387

[Grain].MaximumPotentialGrainSize.FixedValue = 0.33

#### **1.19.24 sr52**

sr52 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 250

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 468

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.25 sc501**

sc501 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 250

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 468

[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

#### **1.19.26 r201**

r201 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 230

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170



[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 450

[Grain].MaximumPotentialGrainSize.FixedValue = 0.375

#### **1.19.27 sc401**

sc401 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 230

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 450

[Grain].MaximumPotentialGrainSize.FixedValue = 0.375

#### **1.19.28 NSCM\_41**

NSCM\_41 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 230

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 170

[Phenology].GrainFilling.Target.FixedValue = 550

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 450

[Grain].MaximumPotentialGrainSize.FixedValue = 0.375

#### **1.19.29 Makueni**

Makueni overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 150

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 115

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 120

[Phenology].GrainFilling.Target.FixedValue = 530

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 432

[Grain].MaximumPotentialGrainSize.FixedValue = 0.375

### **1.19.30 Katumani**

Katumani overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 150

[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24

[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 115

[Phenology].FlagLeafToFlowering.Target.FixedValue = 10

[Phenology].FloweringToGrainFilling.Target.FixedValue = 120

[Phenology].GrainFilling.Target.FixedValue = 530

[Rachis].DMDemands.Structural.DMDemandFunction.MaximumOrganWt.FixedValue = 14

[Grain].MaximumGrainsPerCob.FixedValue = 405

[Grain].MaximumPotentialGrainSize.FixedValue = 0.375

### **1.19.31 Pioneer\_39G12**

Pioneer\_39G12 overrides the following properties:

[Phenology].Juvenile.Target.FixedValue = 180

[Phenology].GrainFilling.Target.FixedValue = 500

[Grain].MaximumGrainsPerCob.FixedValue = 550

[Grain].NumberFunction.GrowthRateFactor.XYPairs.X = 0.2,2,4,10

[Grain].NumberFunction.GrowthRateFactor.XYPairs.Y = 0,0.6,1,1

[Grain].MaximumPotentialGrainSize.FixedValue = 0.25

## **1.20 Spike**

### **1.20.1 Spike**

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

Spike summarises the following biomass objects:

- \* Husk
- \* Rachis

## **1.21 MortalityRate**

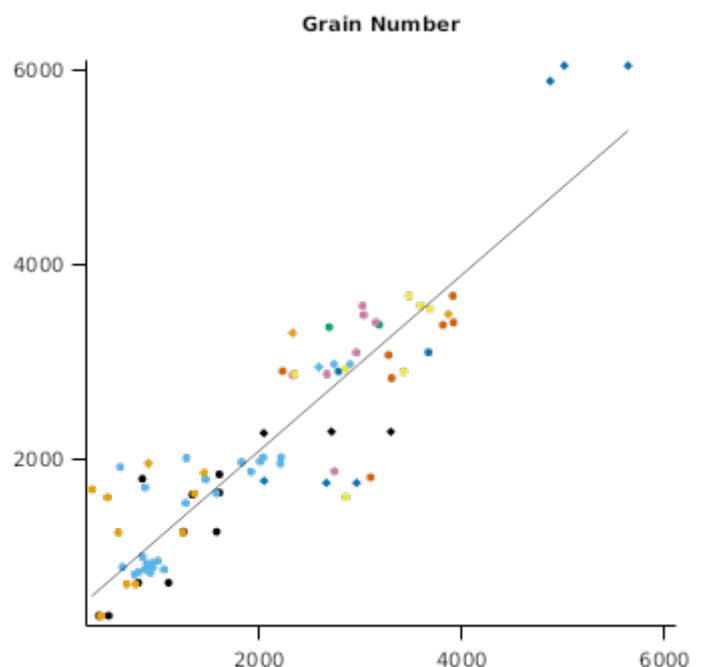
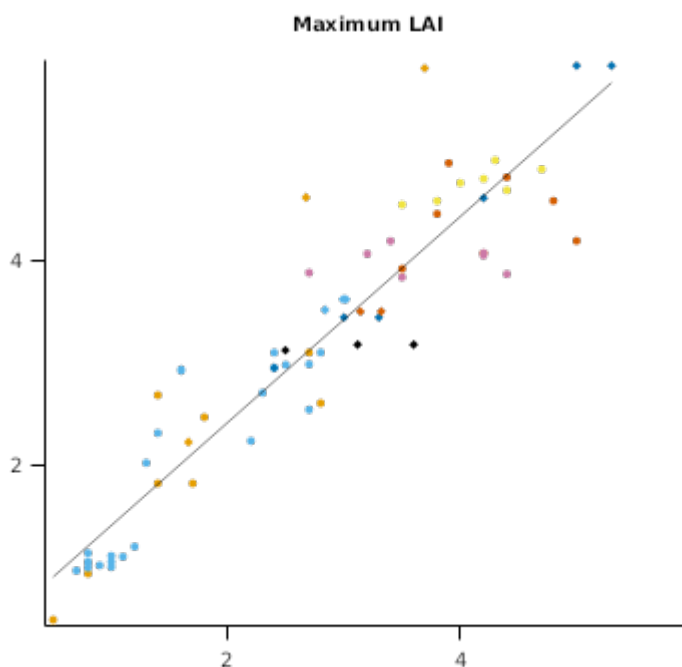
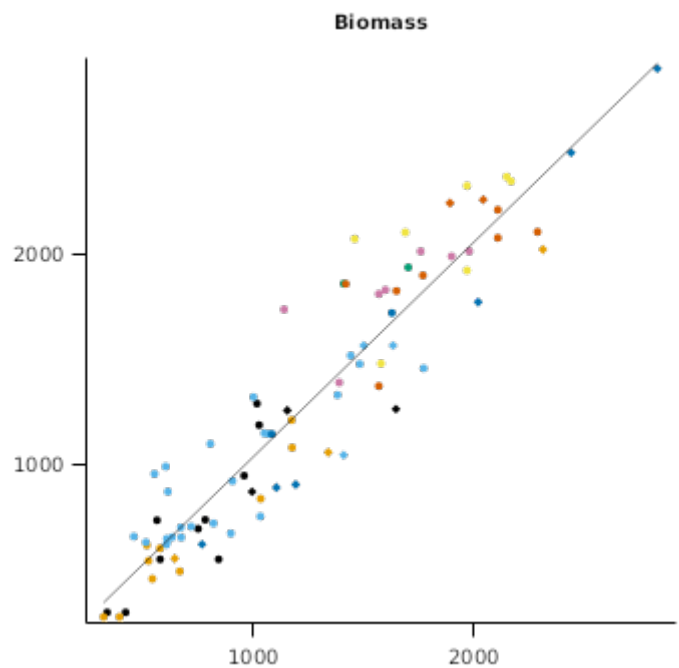
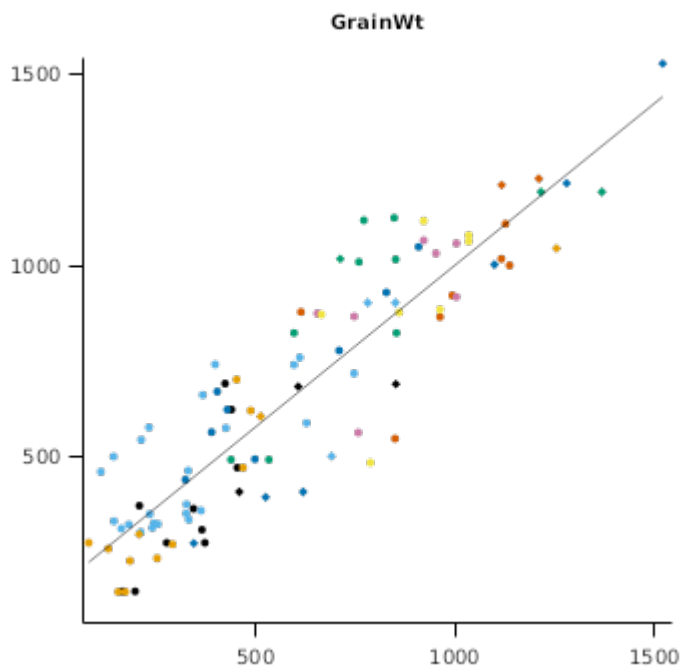
MortalityRate = 0

The APSIM Maize model has been tested on a range of datasets from around the world to provide tests across a range of environmental conditions. Tests include ranges in plant population, nitrogen management and irrigation levels, as well as including historical datasets from africa using cultivars developed during the 1960's, through to modern varieties grown in the American mid-West.

# **2 Validation**

## **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).



## 2.2 Africa

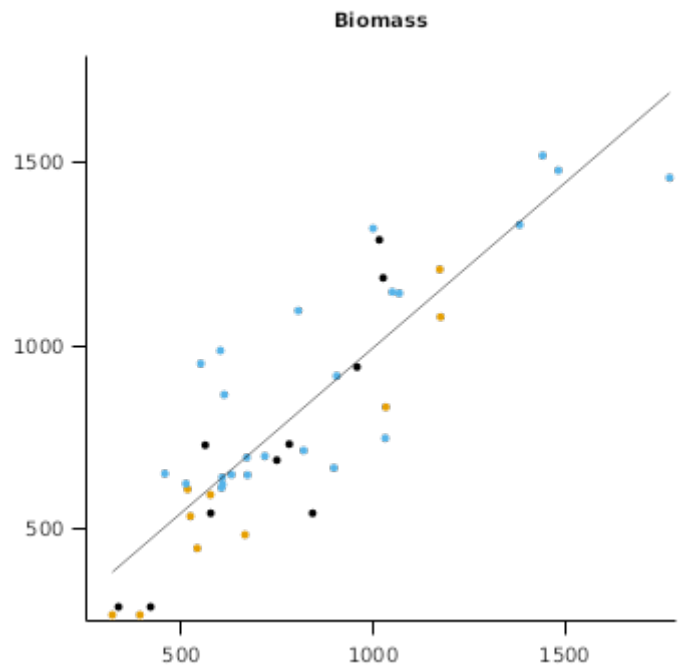
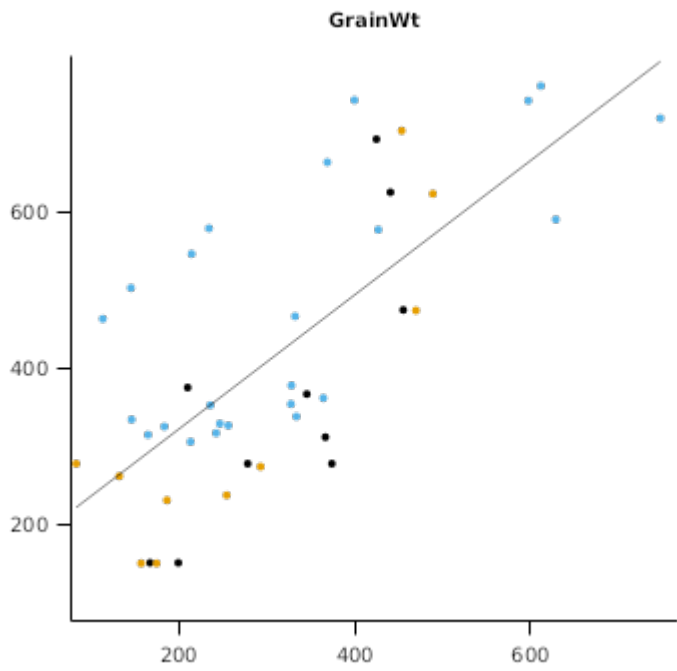
A selection of data from Kenya has been included from the work of [KEATING et al., 1992](#). These include the trial originally labelled JMW2 by the original authors and was labelled Experiment 6 in [KEATING et al., 1992](#). This trial includes maize (cultivar Katumani Composite B) sown at 5 populations (1.1, 2.2, 4.4, 6.6 and 8.8 plants/m<sup>2</sup>) under low (0 kg N/ha applied) and high (120 kgN/ha) fertiliser conditions during the Long Rains of 1988. This same experimental design was conducted at two locations, Katumani (1o 35' S, 37o 14' E) and Kiboko (2o 13' S, 37o 43' E). The trial was rainfed at Katumani but fully irrigated at Kiboko. Yields ranged from 2000 to 5400 kg/ha at Katumani and 1000 to 6000 kg/ha at Kiboko. There were strong population x nitrogen interactions at both sites. A second trial referred to as BMW1 or Experiment 1 in [KEATING et al., 1992](#). A range of populations (2.0-6.5 plants/m<sup>2</sup>) and irrigation treatments (6-176 mm) were planted under high (80 kgN/ha) and low (0kgN/ha) fertiliser conditions at Katumani. Individual replicates are modelled separately due to variability in soil (depth to rock), establishment and irrigation application. Grain yields varied from 1600 to 8000 kg/ha.

Note. The BMW1 trail consists of data from unreplicated plots.

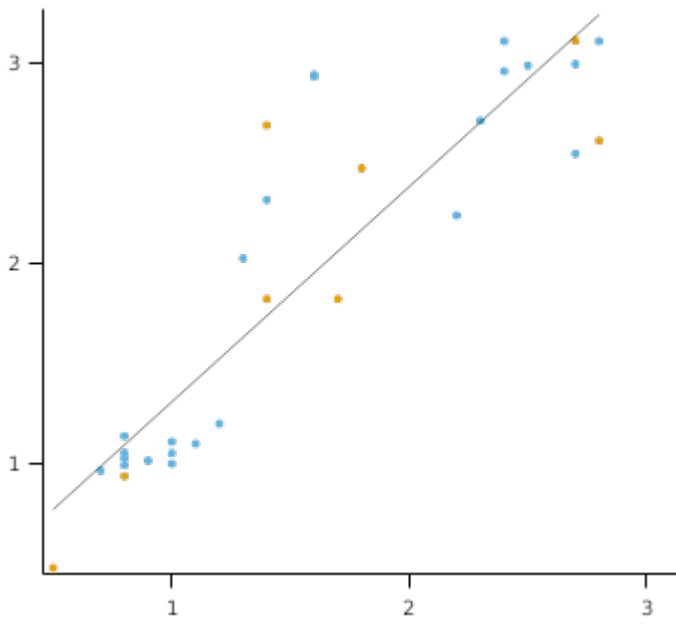
### List of experiments.

Experiment Name	Design (Number of Treatments)
JMW2Katumani	Popn x Fert (10)

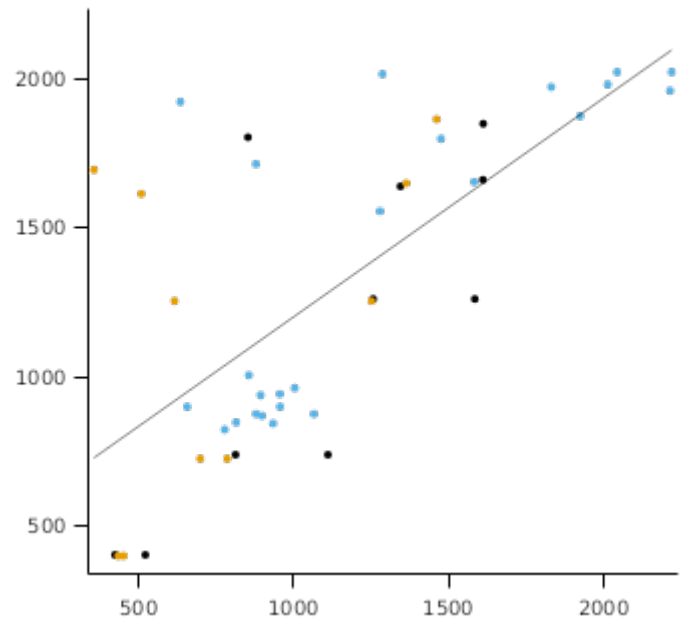
Experiment Name	Design (Number of Treatments)
JMW2Kiboko	Popn x Fert (10)
BMW1	Site (24)



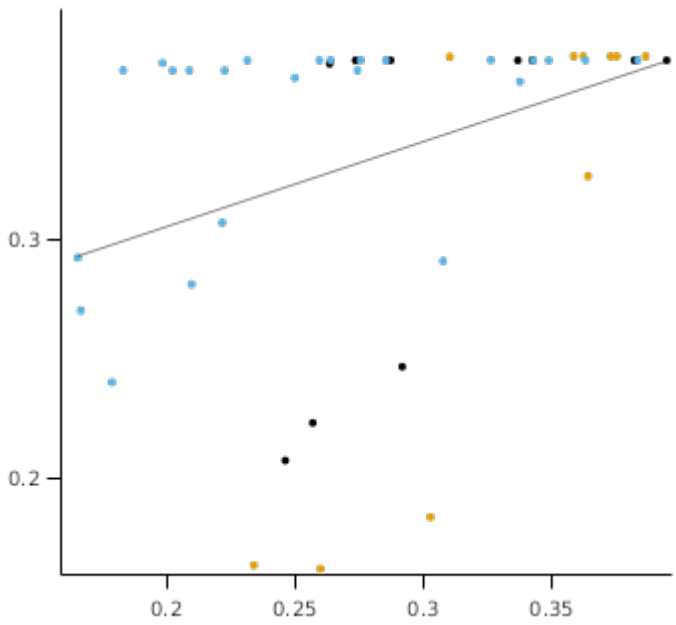
Maximum LAI



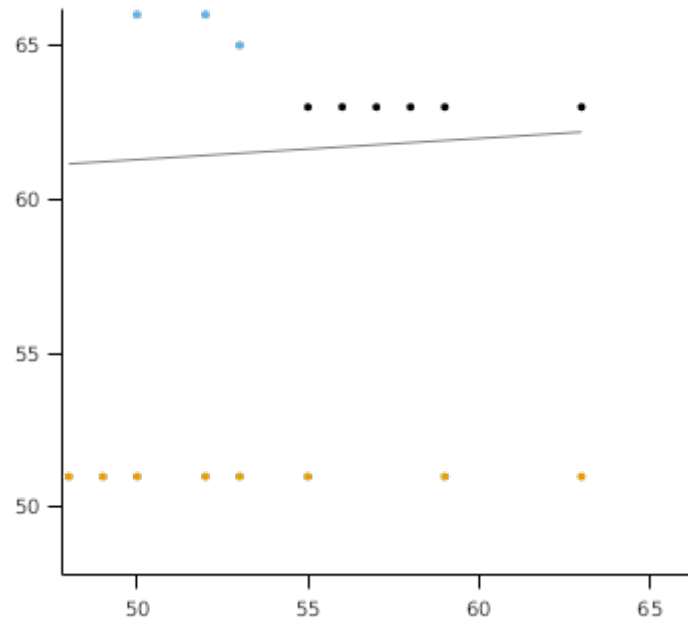
Grain Number



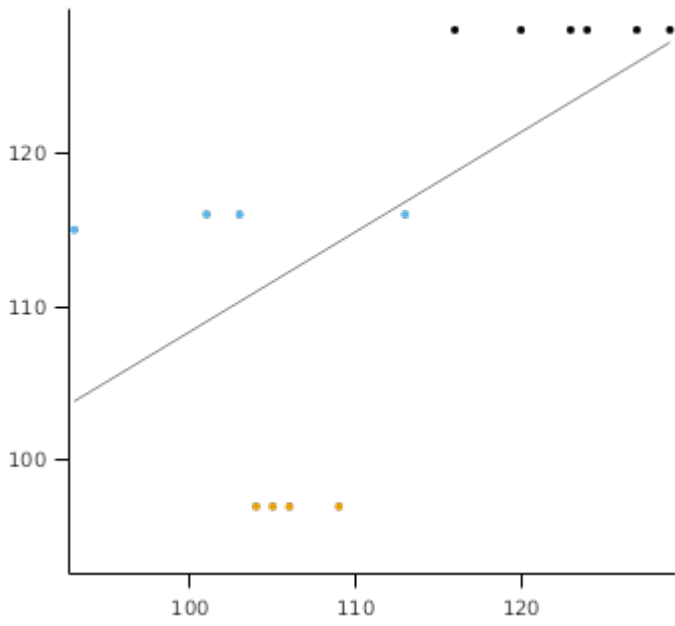
Grain Size



FloweringDAS

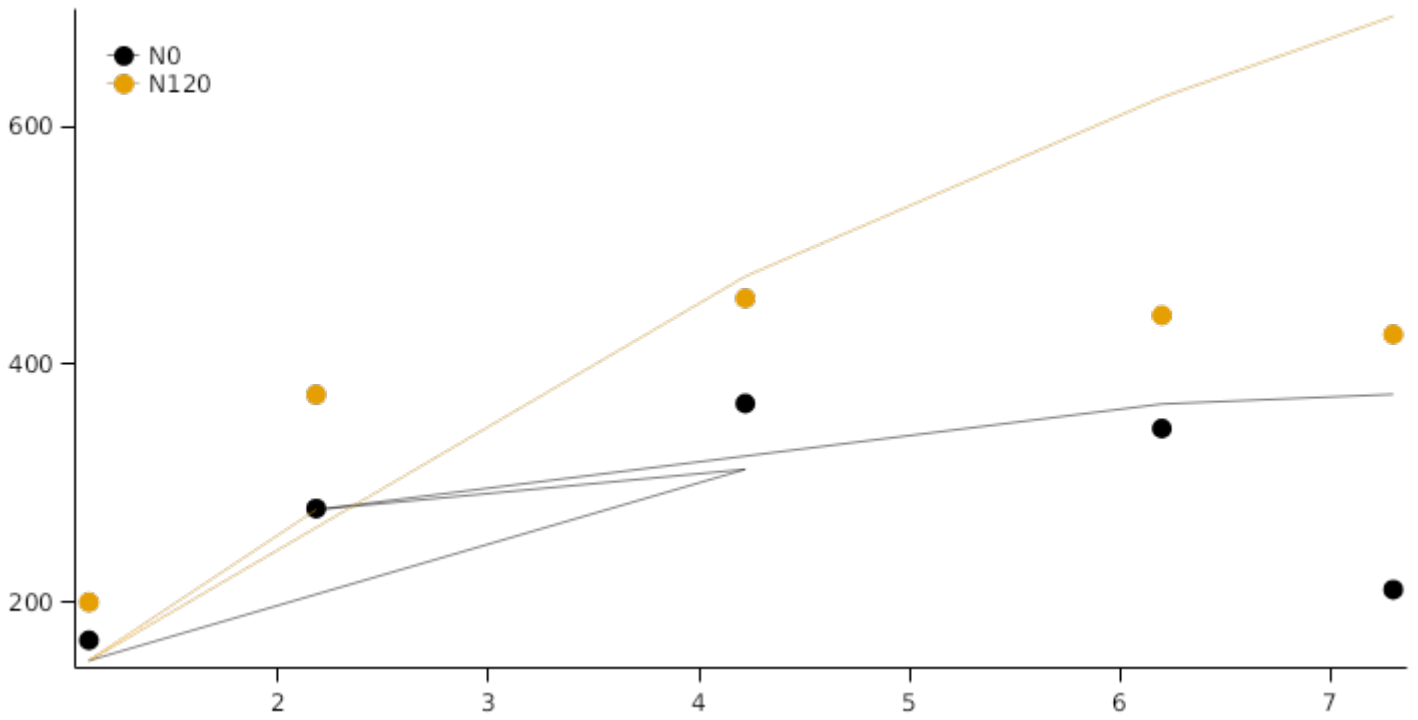


MaturityDAS

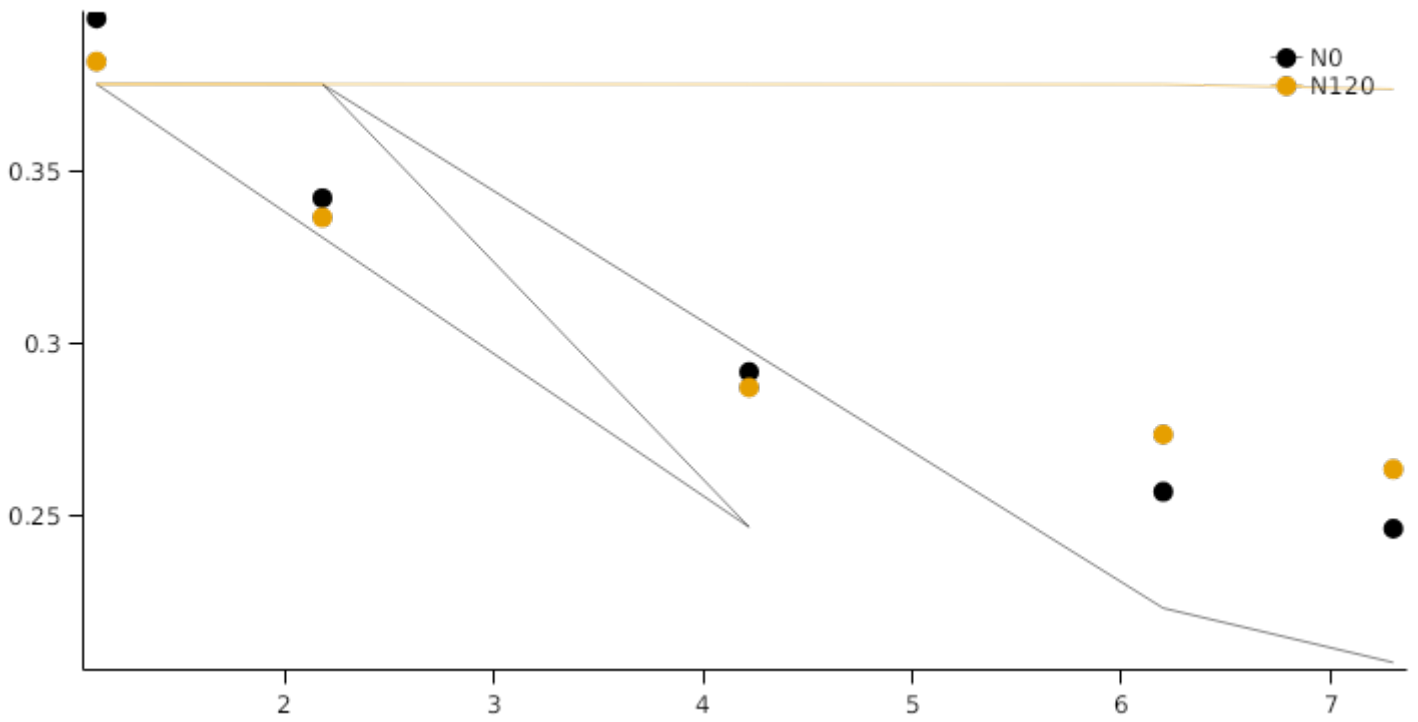


### 2.2.1 JMW2Katumani

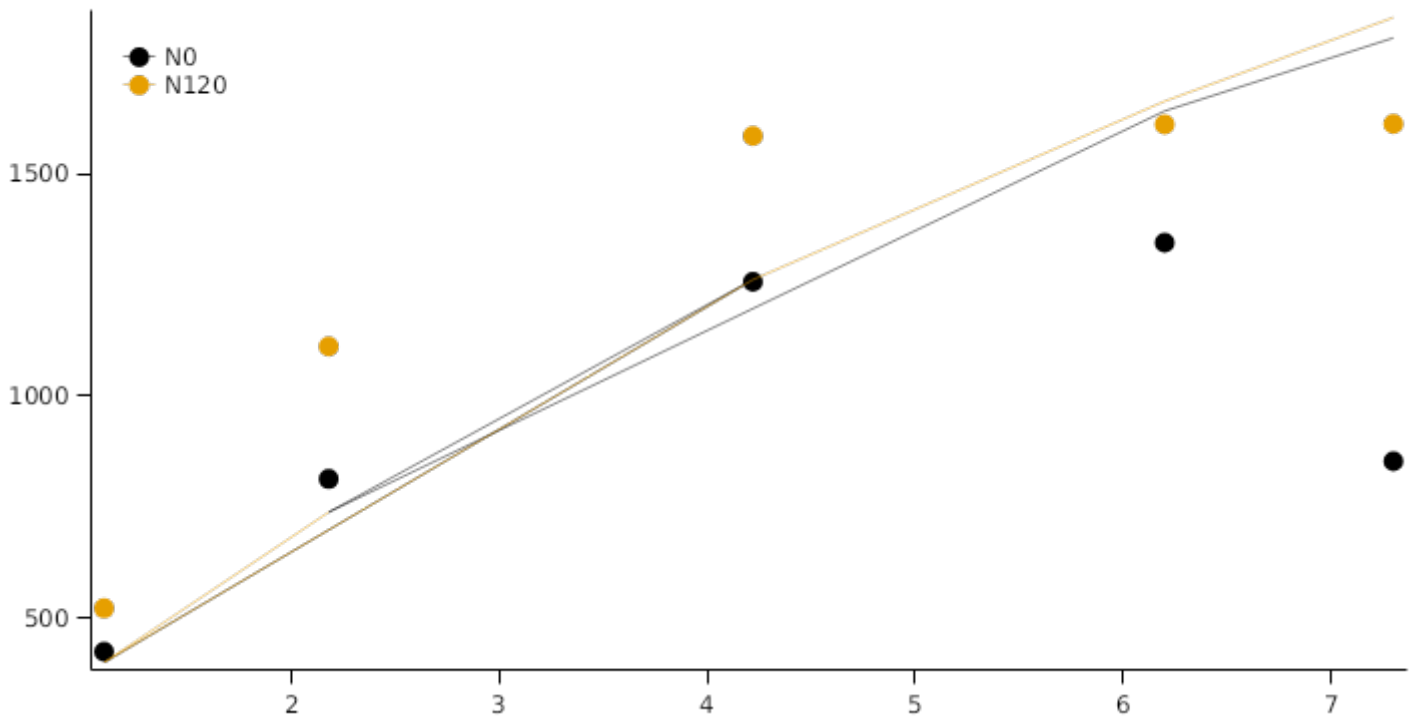
GrainWt Population Response



### GrainSize Population Response

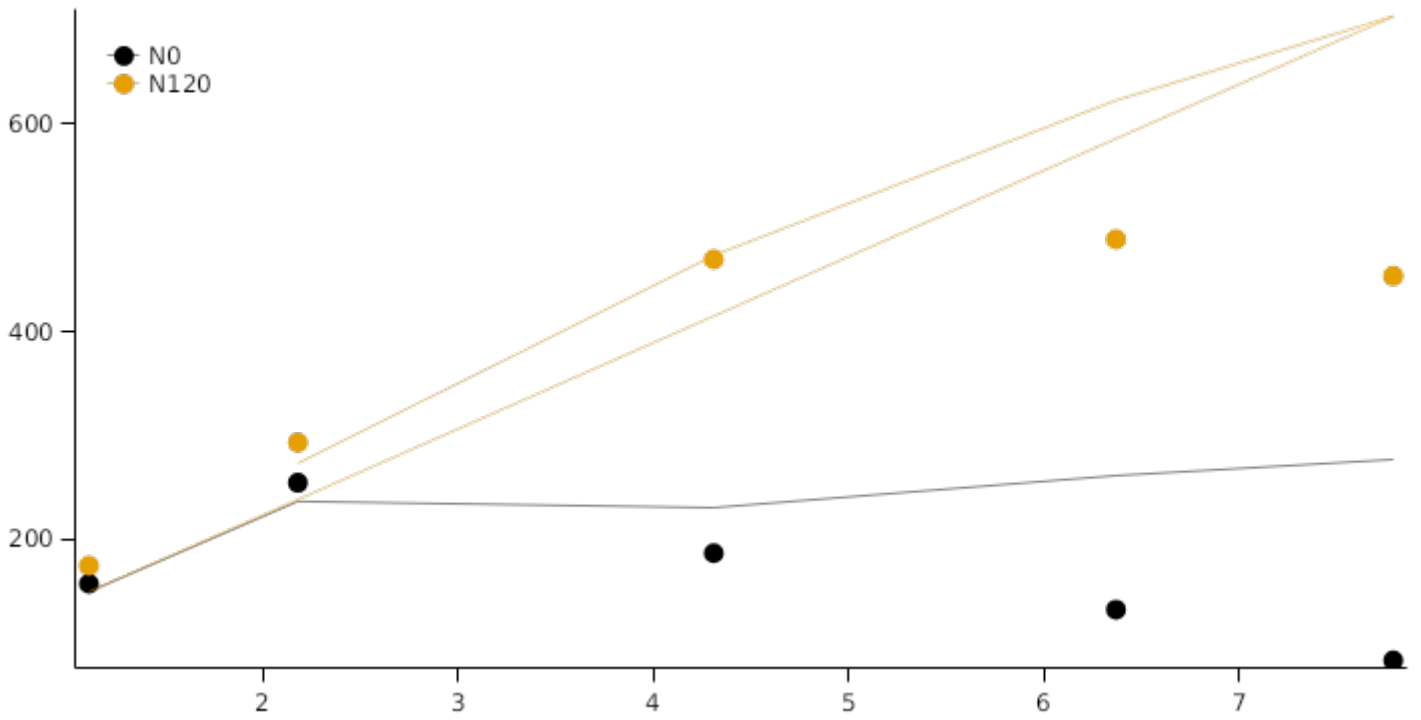


### GrainNo Population Response

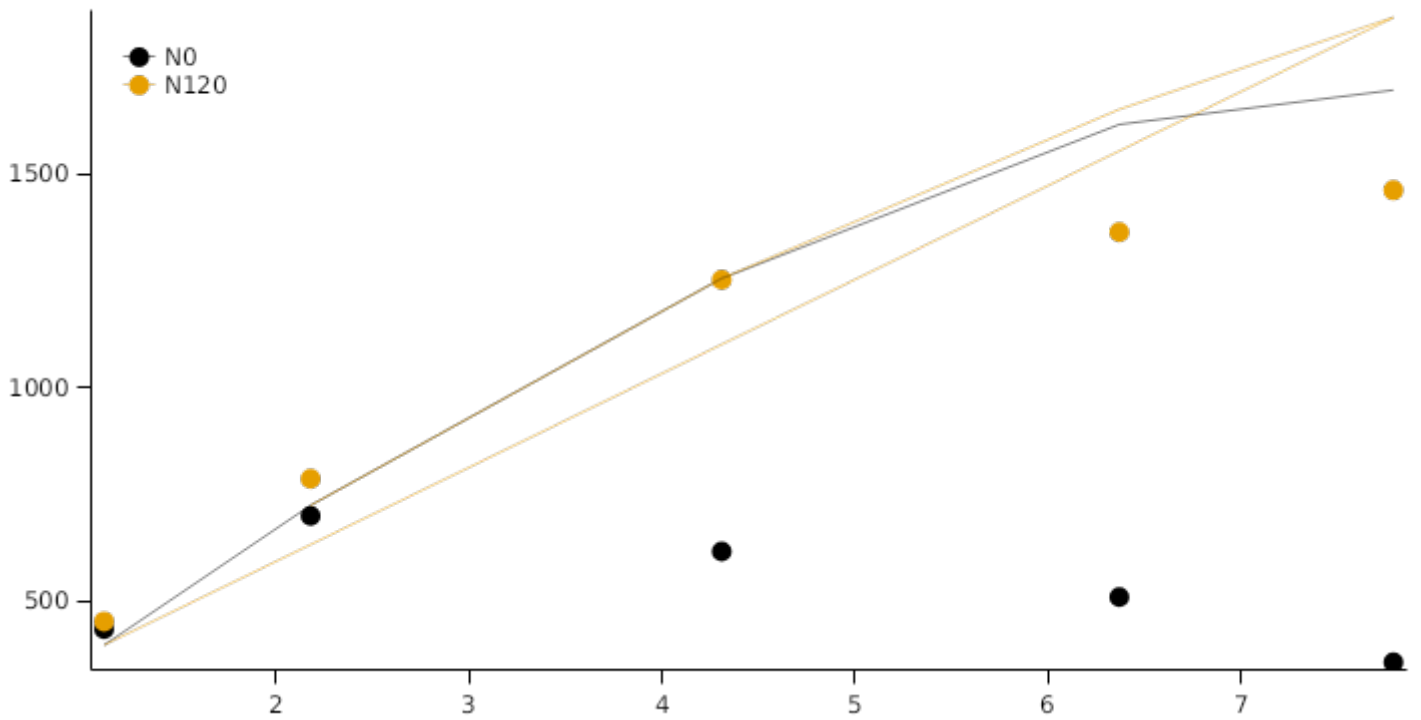


#### 2.2.2 JMW2Kiboko

### GrainWt Population Response

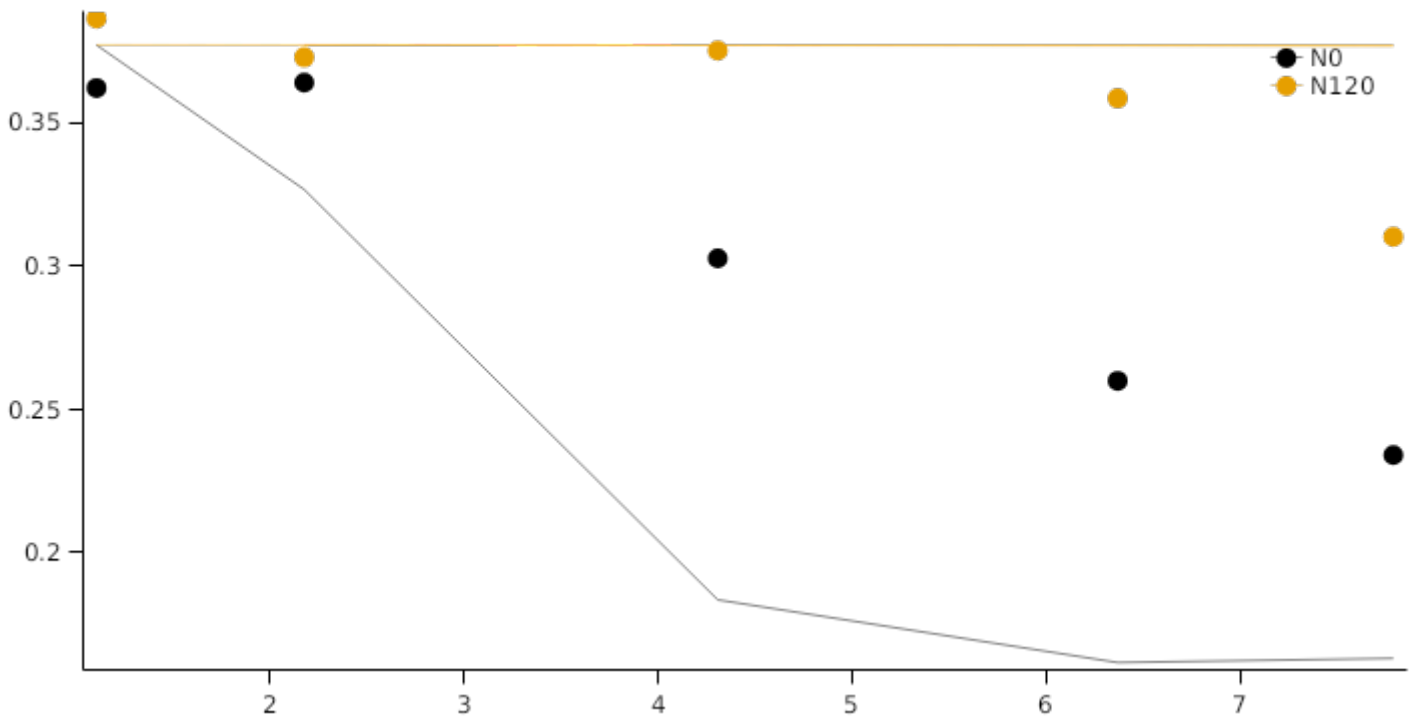


### GrainNo Population Response

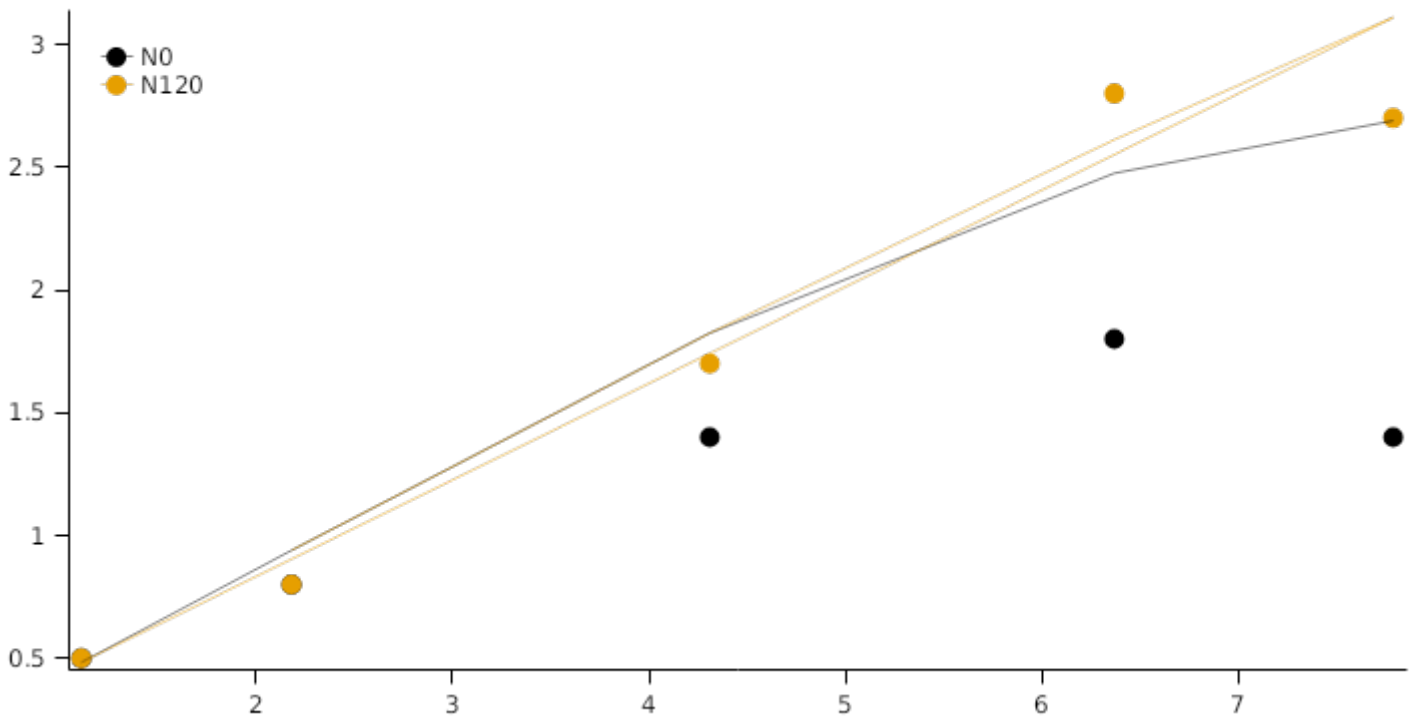




### GrainSize Population Response



### LAImax Population Response



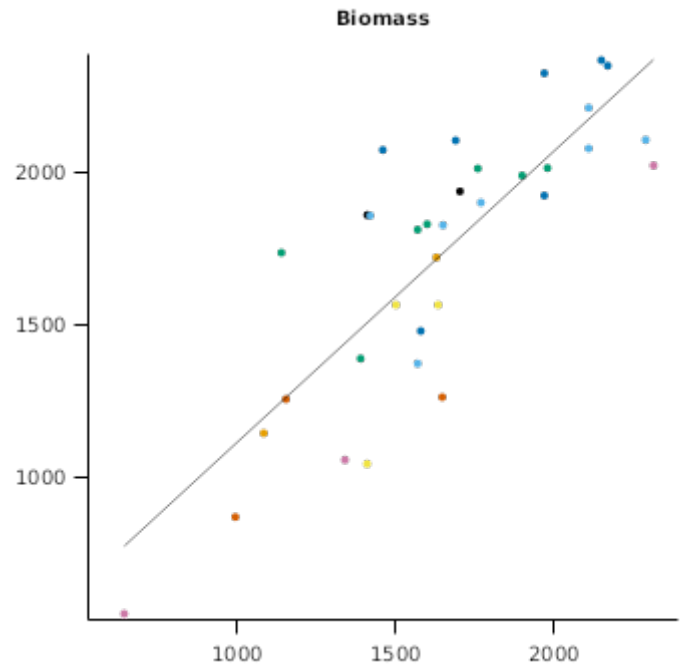
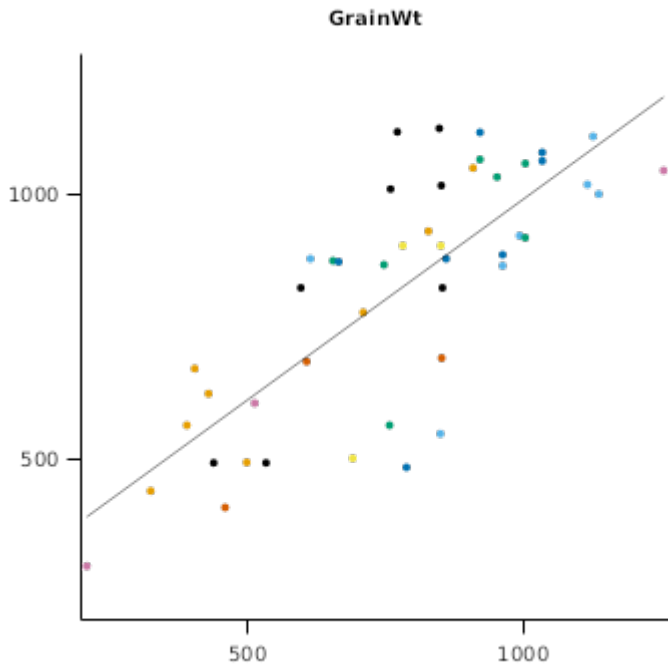
## 2.3 Australia

The data for two trials from near Katherine, Northern Territory, Australia, have been included from [CARBERRY et al., 1989](#) and [CARBERRY et al., 1991](#). These include two planting dates for the summer of 1983/1984 (labelled as K841, K842). At each planting date, cultivar Dekalb XL82 was sown at 4 populations (3, 5, 7, 9 plants/m<sup>2</sup>) under low and high irrigation conditions. The low irrigation treatments only included preliminary applications to ensure crop establishment.

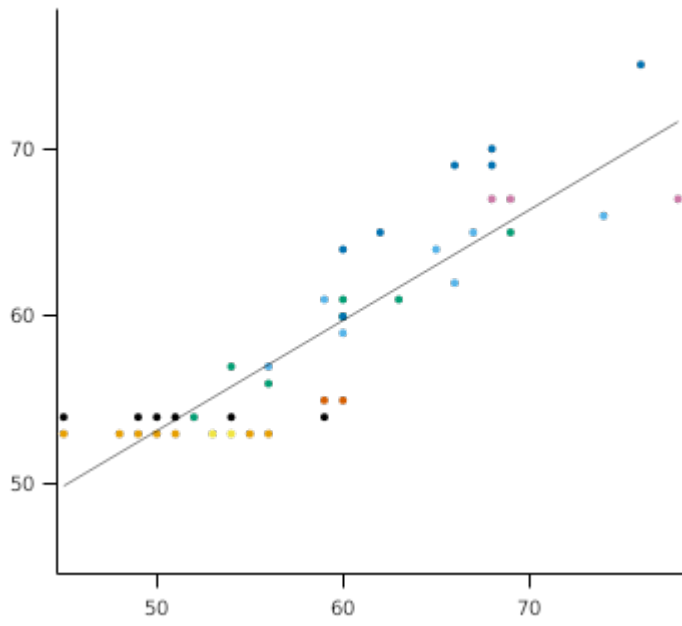
### List of experiments.

Experiment Name	Design (Number of Treatments)
K841	Popn x Irrigation (8)

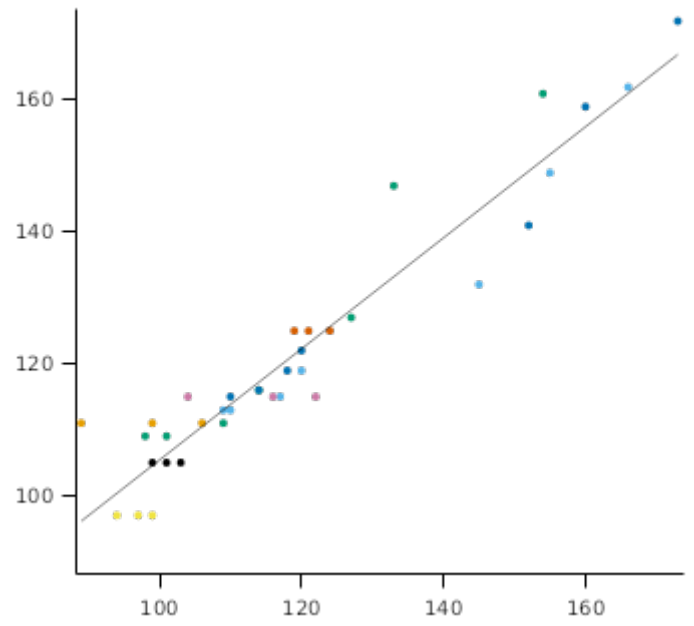
Experiment Name	Design (Number of Treatments)
K842	Popn x Irrigation (8)
DRK1	Sowing (7)
DRK2	Sowing (7)
DRK3	Sowing (7)
Angelo98	Fert (3)
Angelo99	Fert (3)
Angelo00	Fert (3)



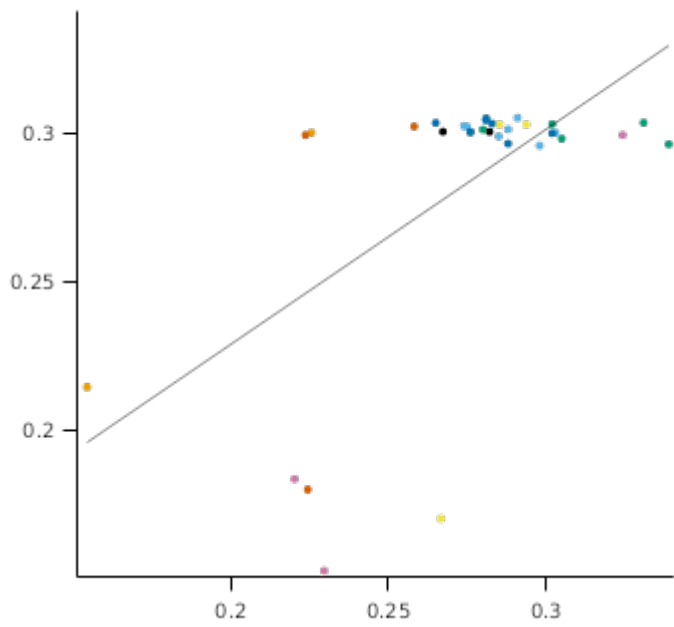
**FloweringDAS**



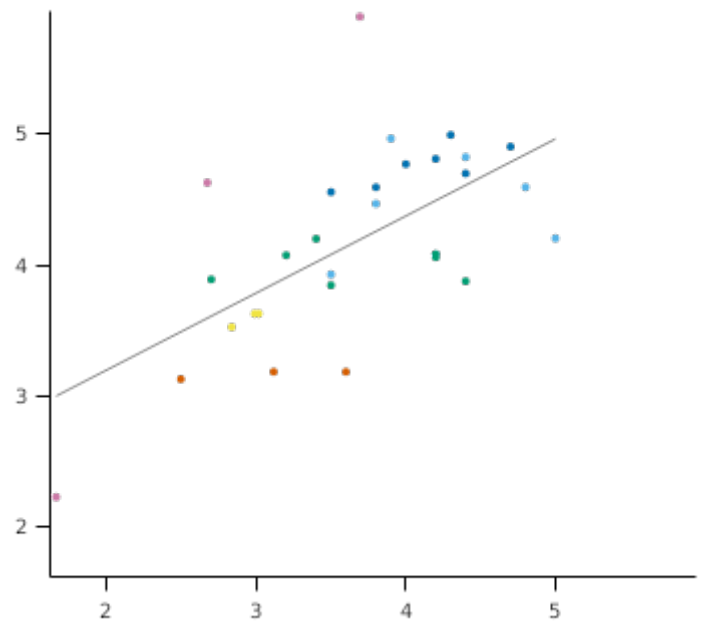
**MaturityDAS**



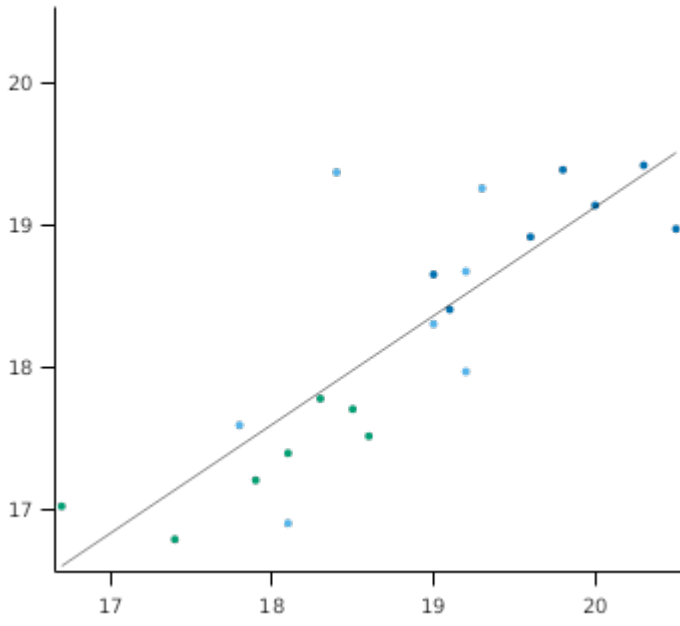
**Grain Size**



**Maximum LAI**

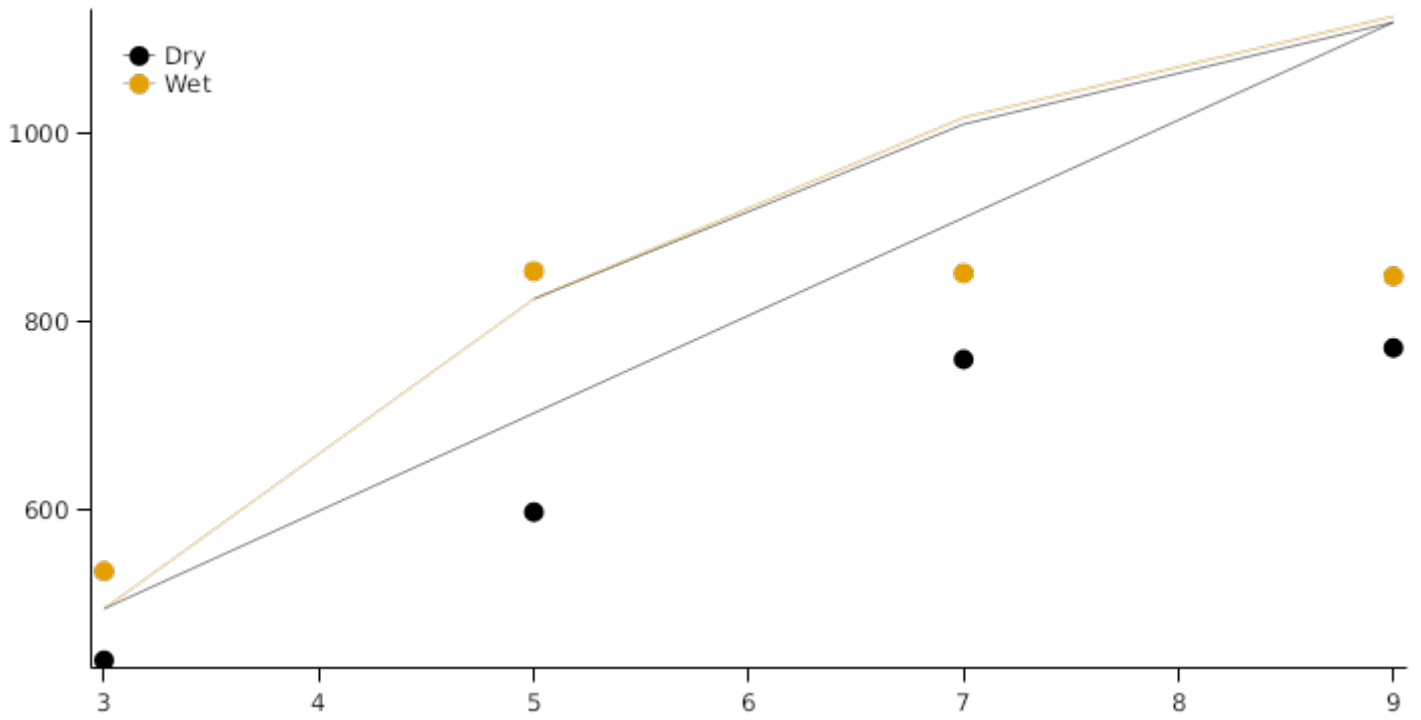


FinalLeafNumber

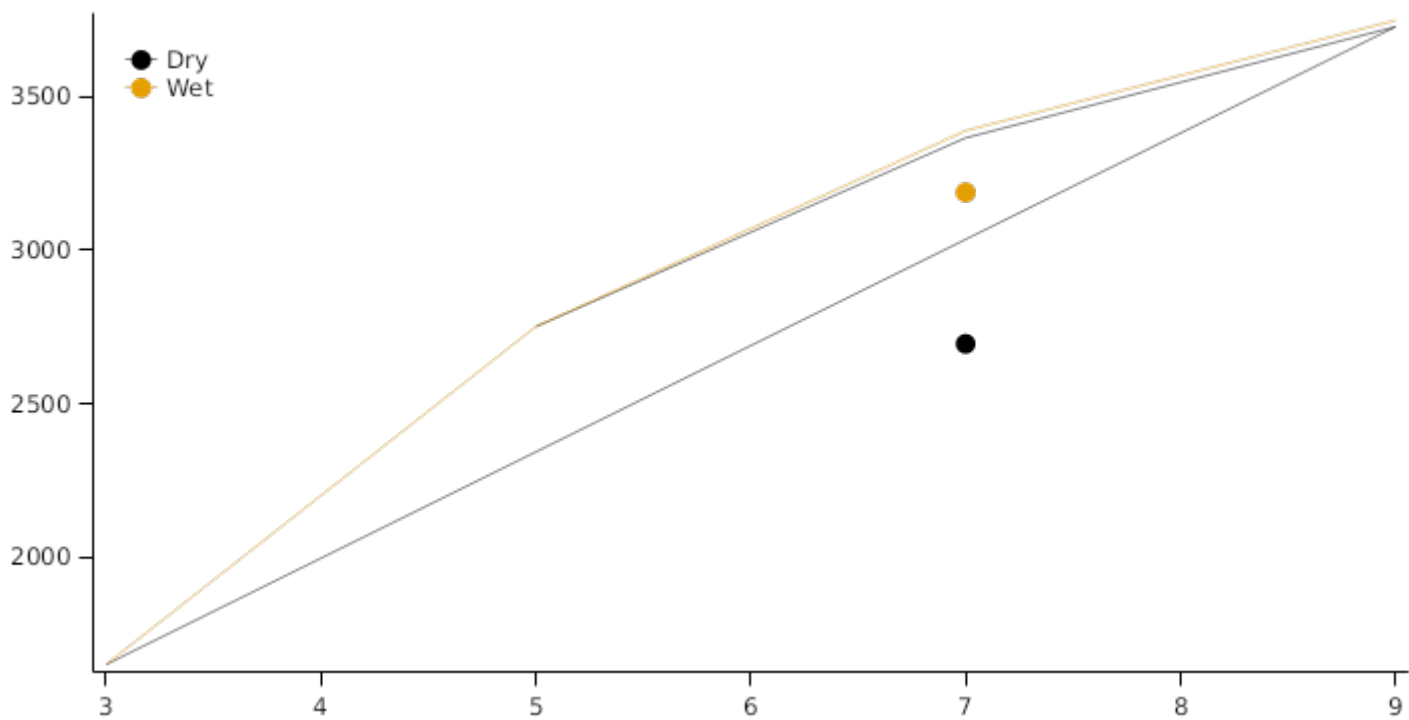


2.3.1 K841

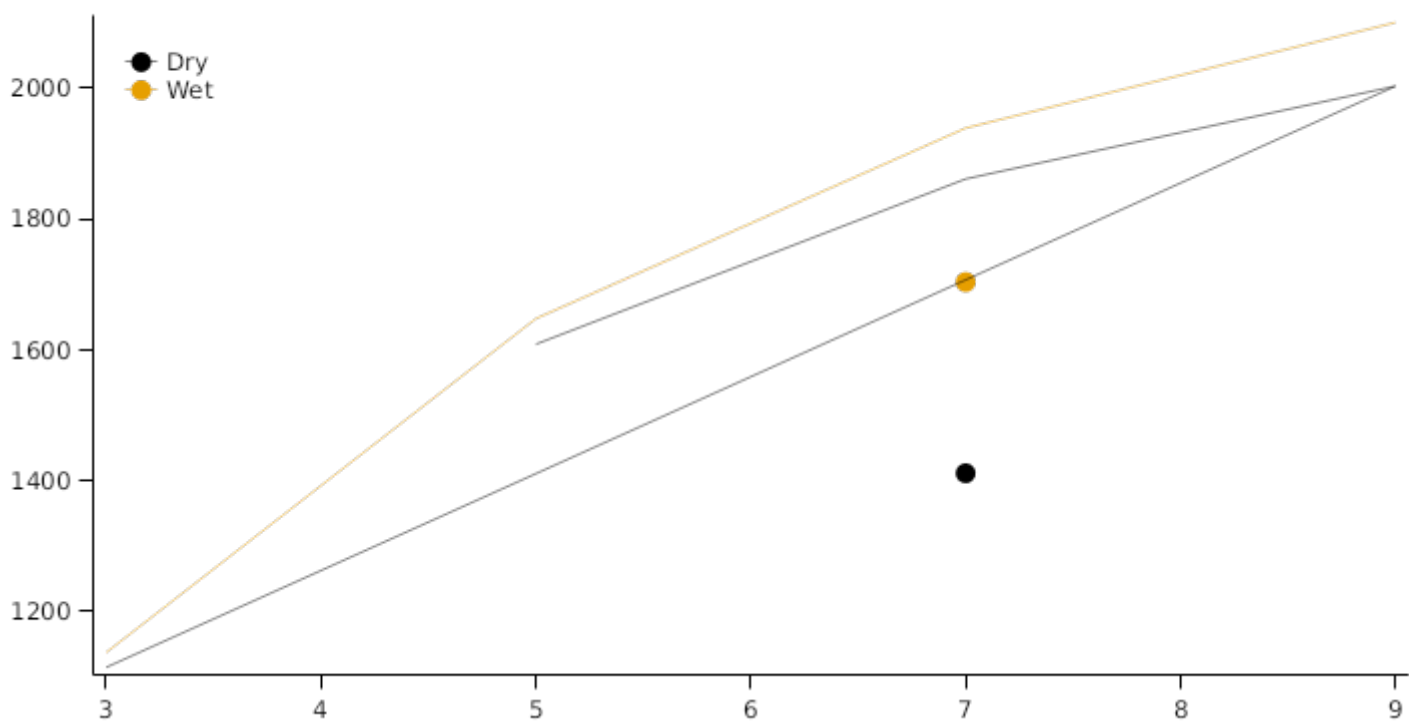
GrainWt Population Response



### GrainNo Population Response

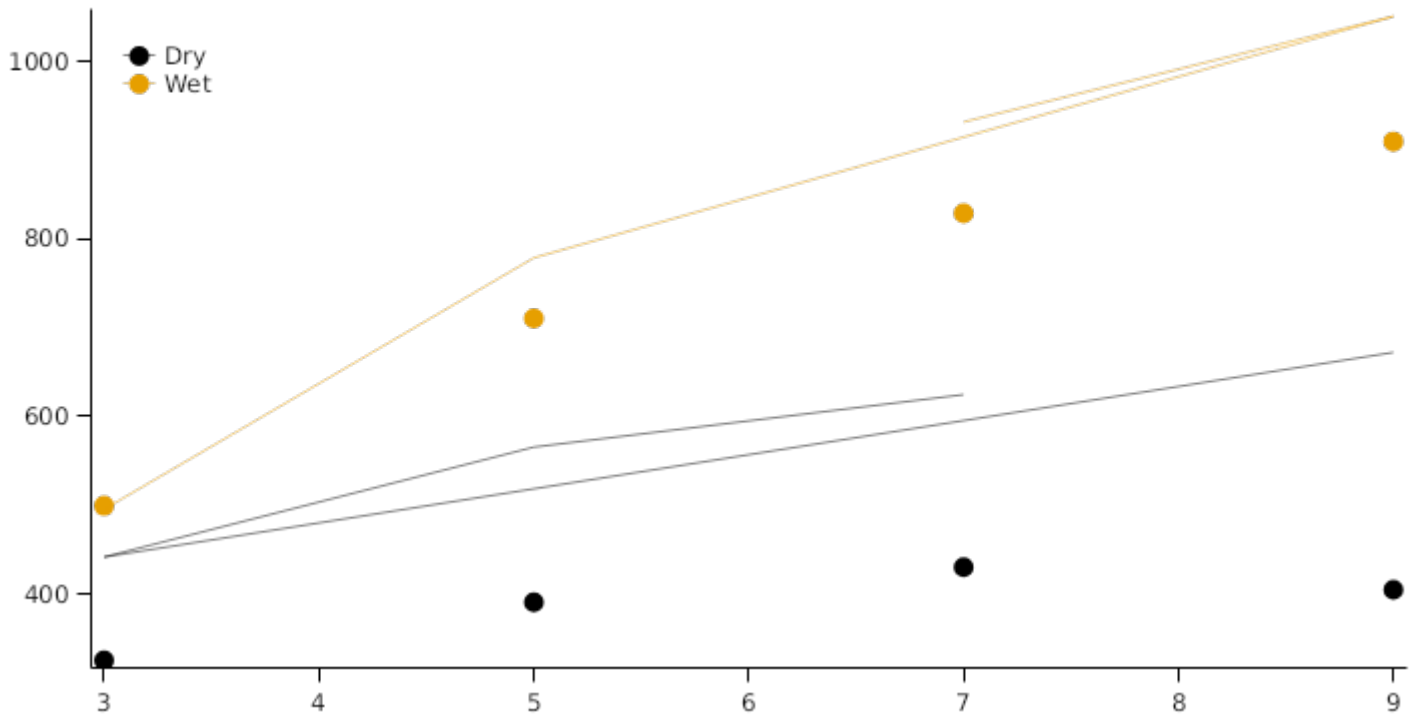


### Biomass Population Response

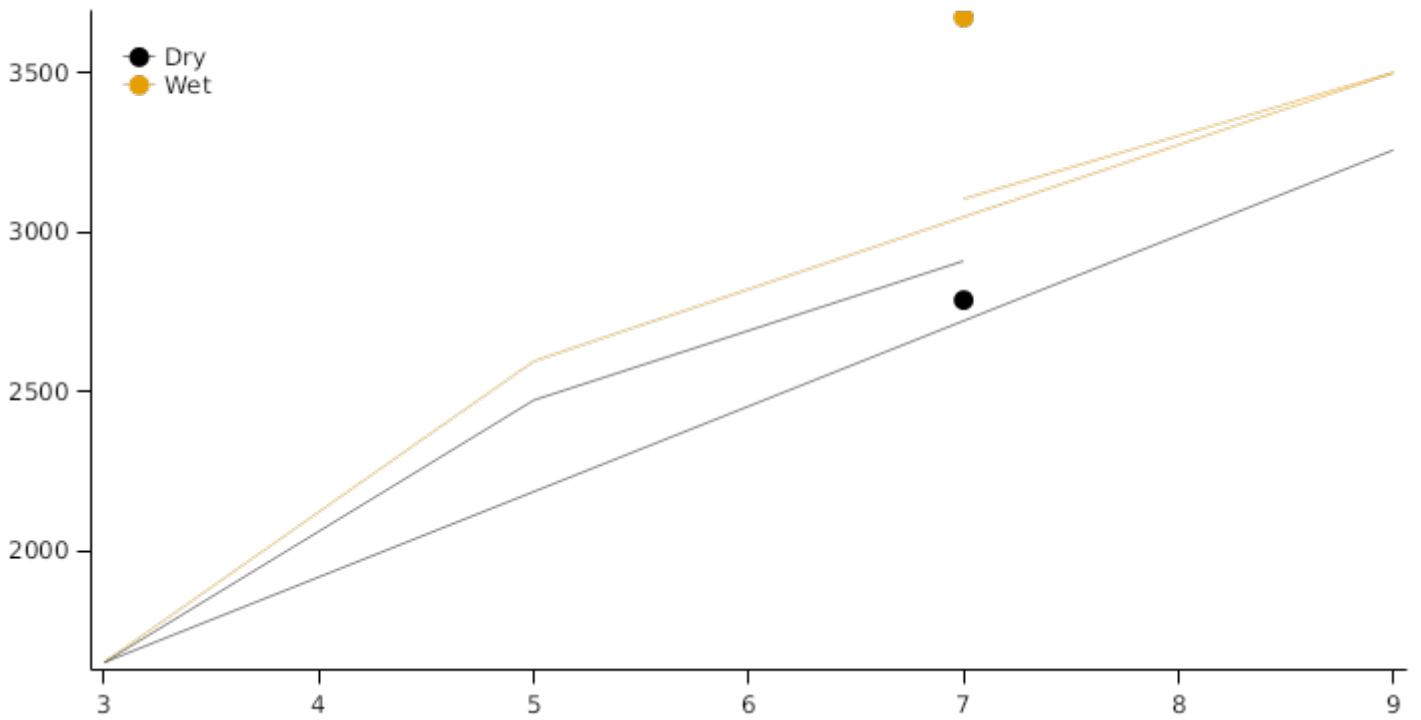


2.3.2 K842

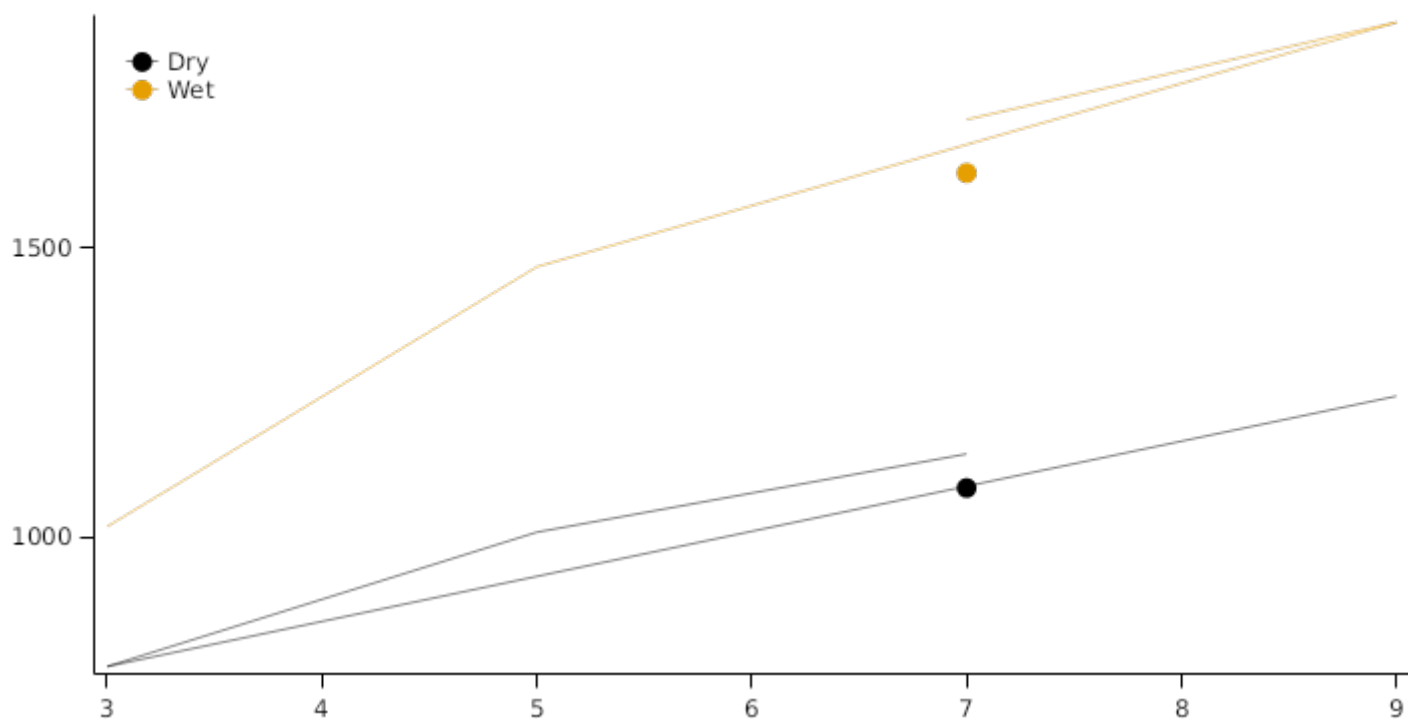
### GrainWt Population Response



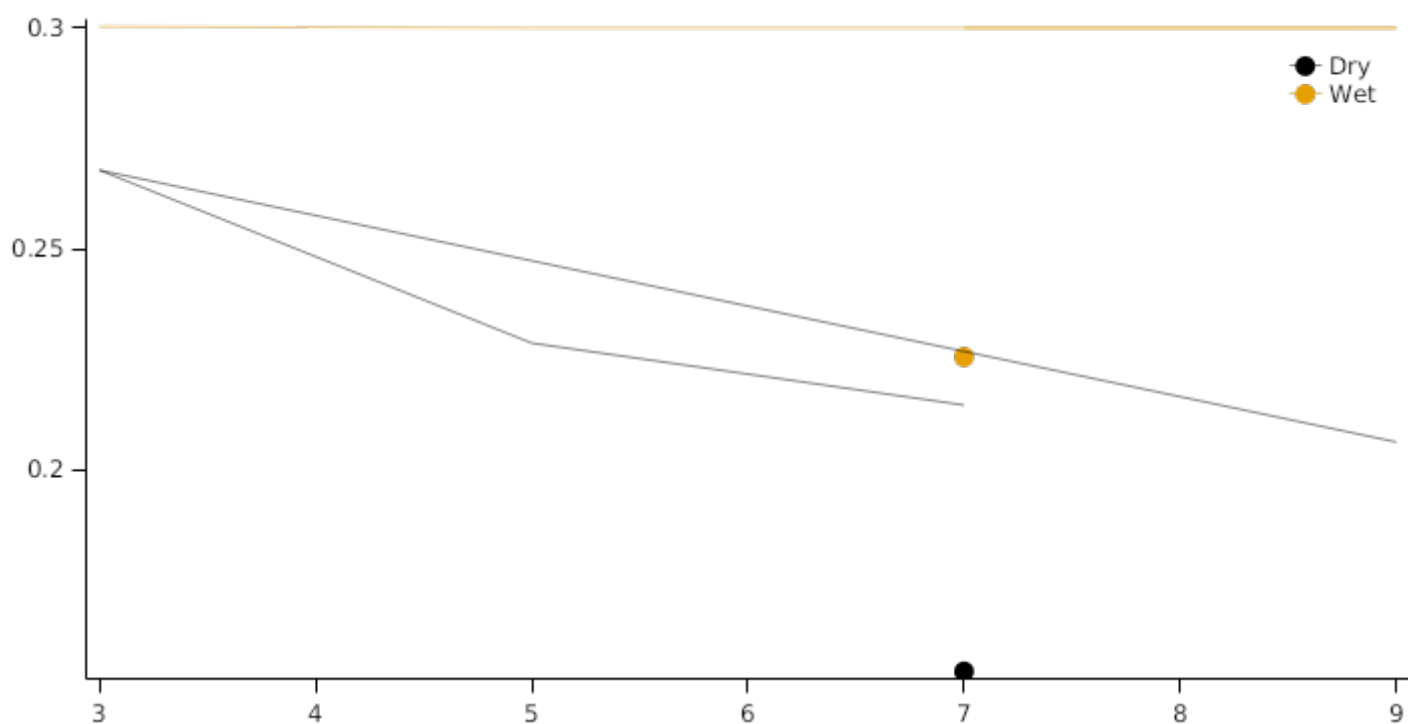
### GrainNo Population Response



## Biomass Population Response



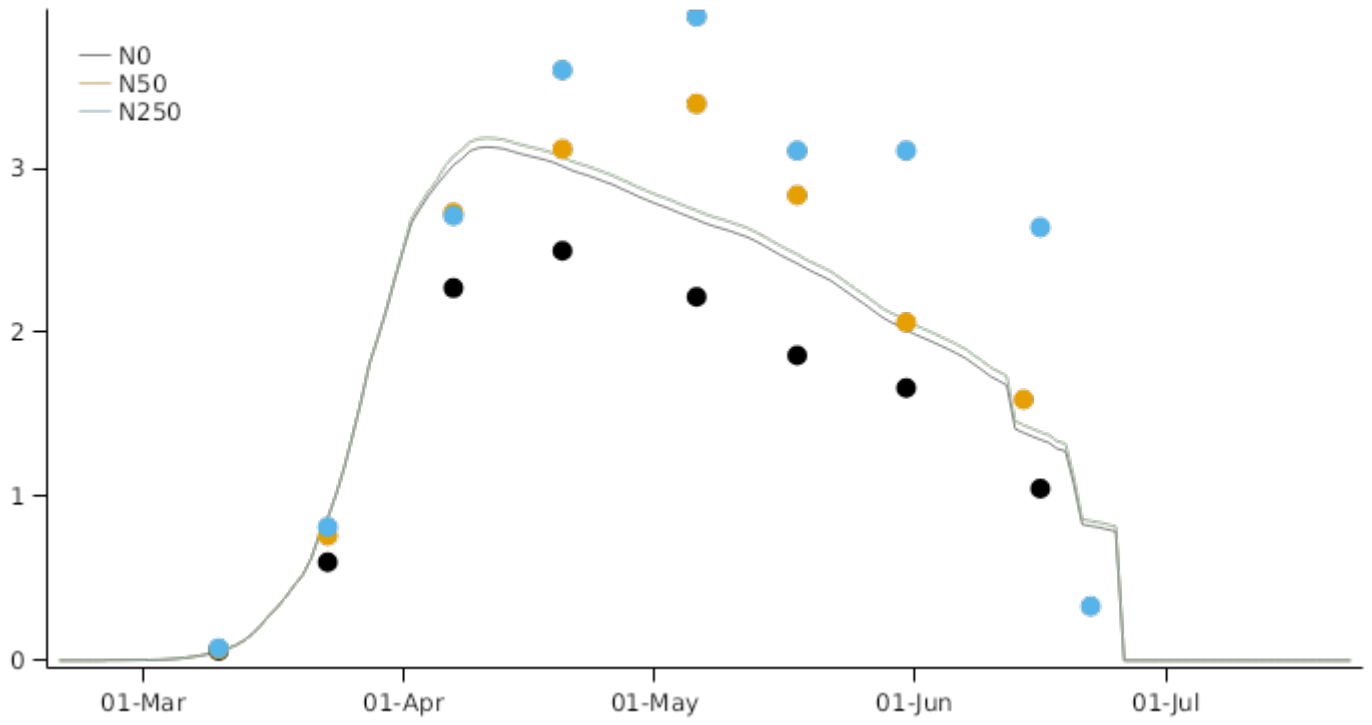
## GrainSize Population Response



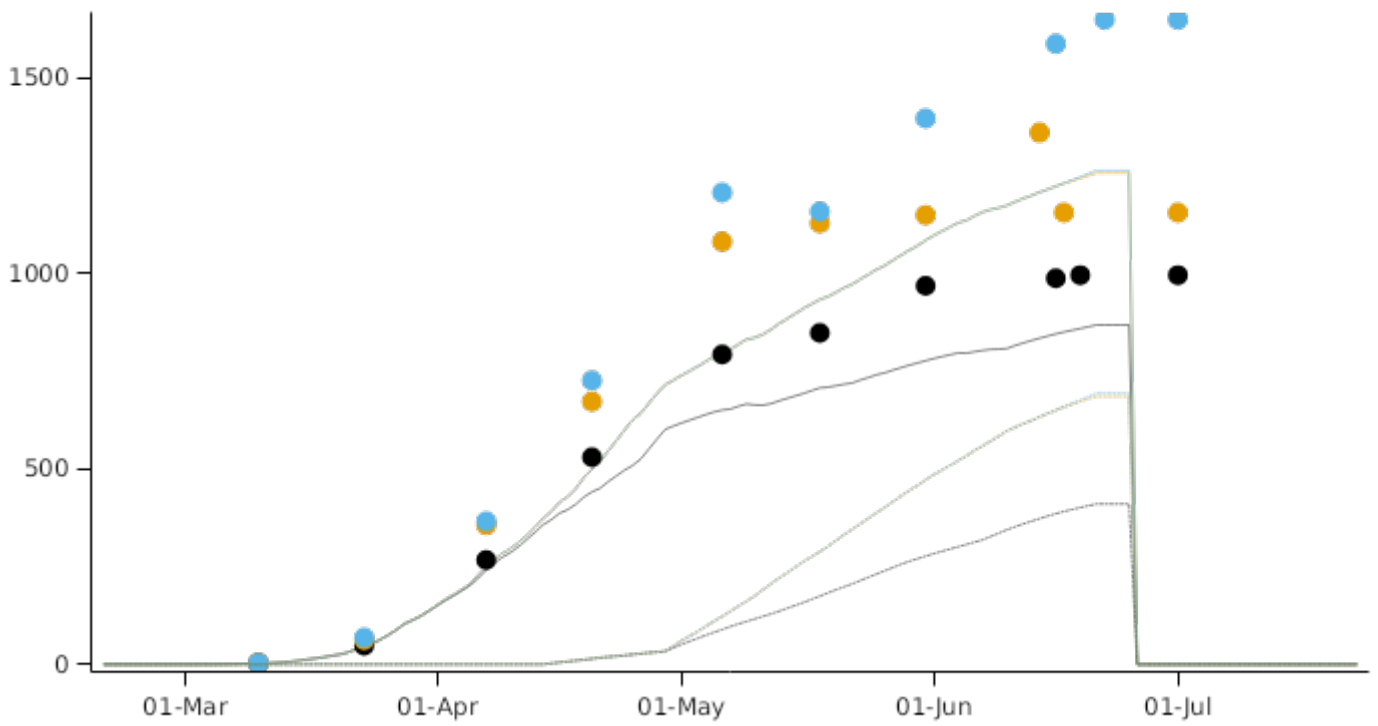
### 2.3.3 Angelo98

This data set has been published by [Massignam et al., 2009](#). Three sowings were conducted at the CSIRO Cooper field station (27°34'S, 152°20'E), Gatton, Australia. The soil was a Lockyer clay loam for sowing 1 and 2 and a Hooper clay loam for sowing 3. Maize (Hycorn53) was sown on three dates to provide a range of growing environments. Each sowing was provided with 3 rates of nitrogen. To ensure moisture was non-limiting, the plots were irrigated weekly by trickle irrigation, except when rainfall occurred. Plots were weeded regularly and insect pests and foliar diseases controlled as required.

### LAI

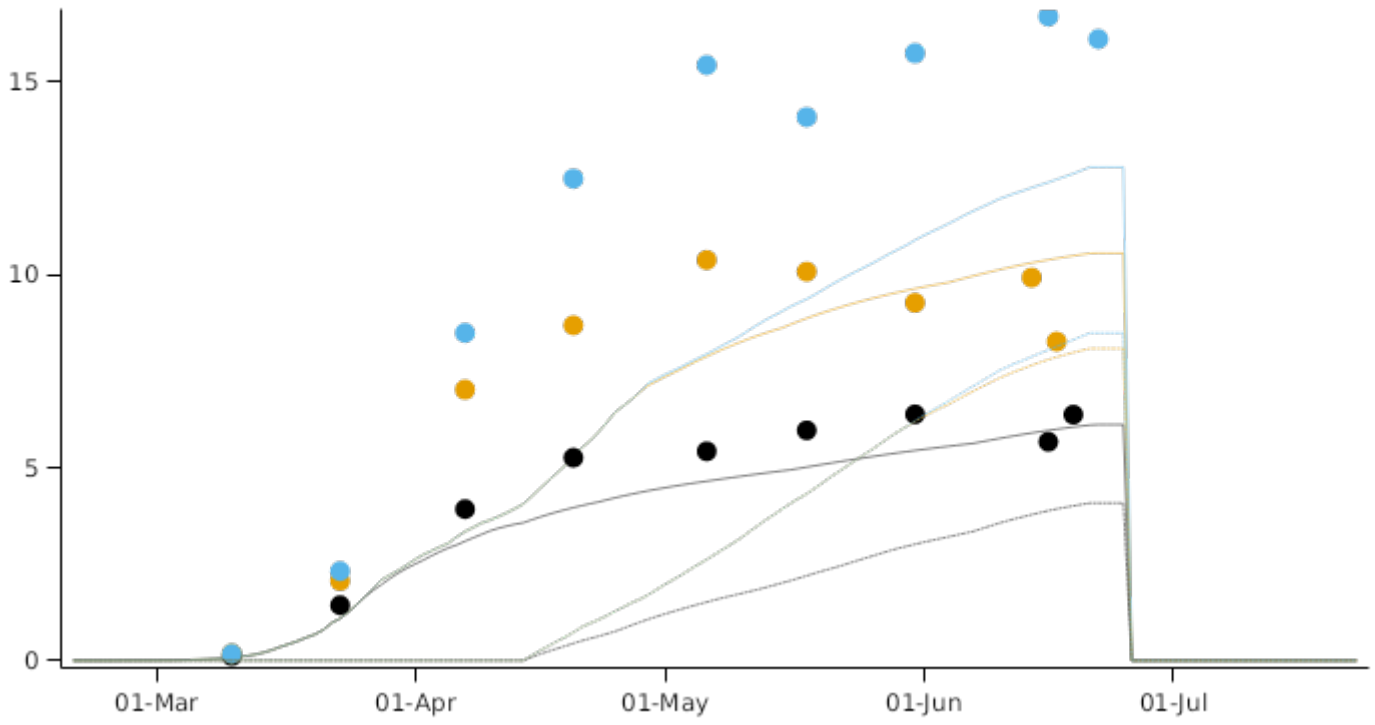


### Biomass

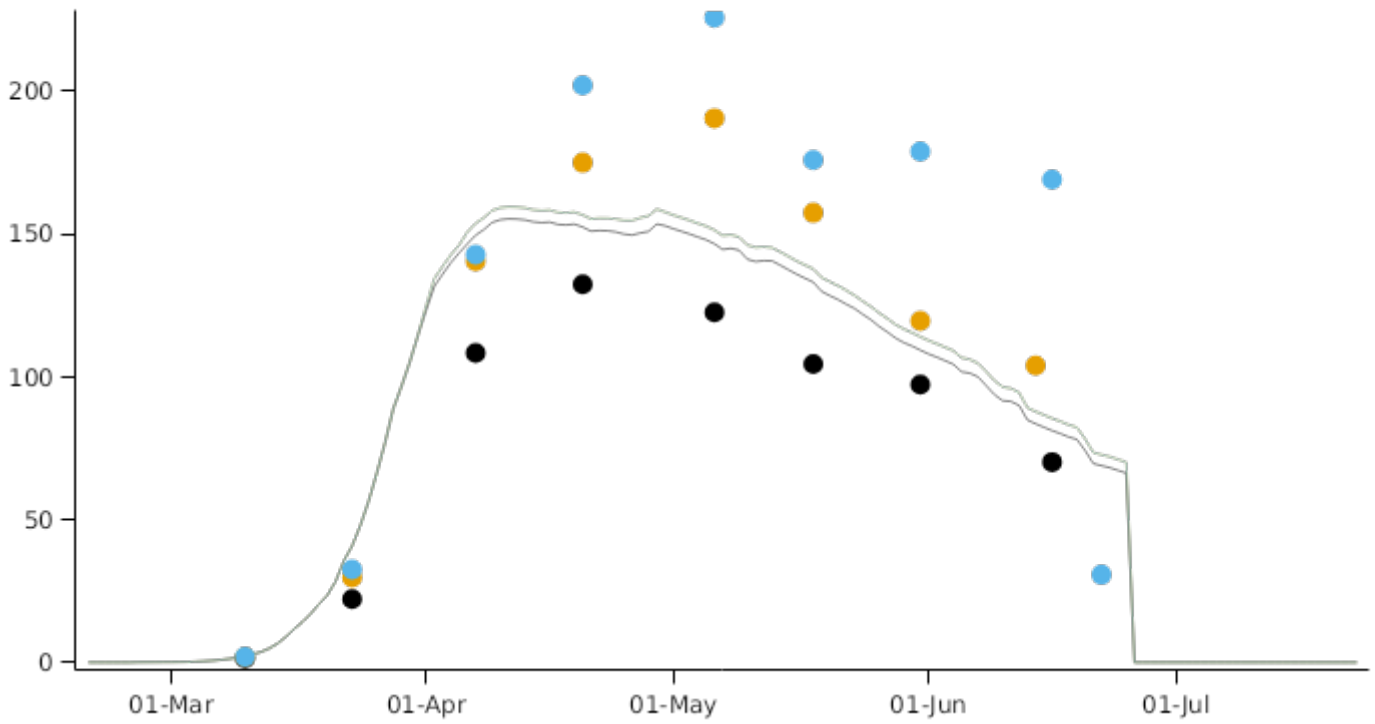




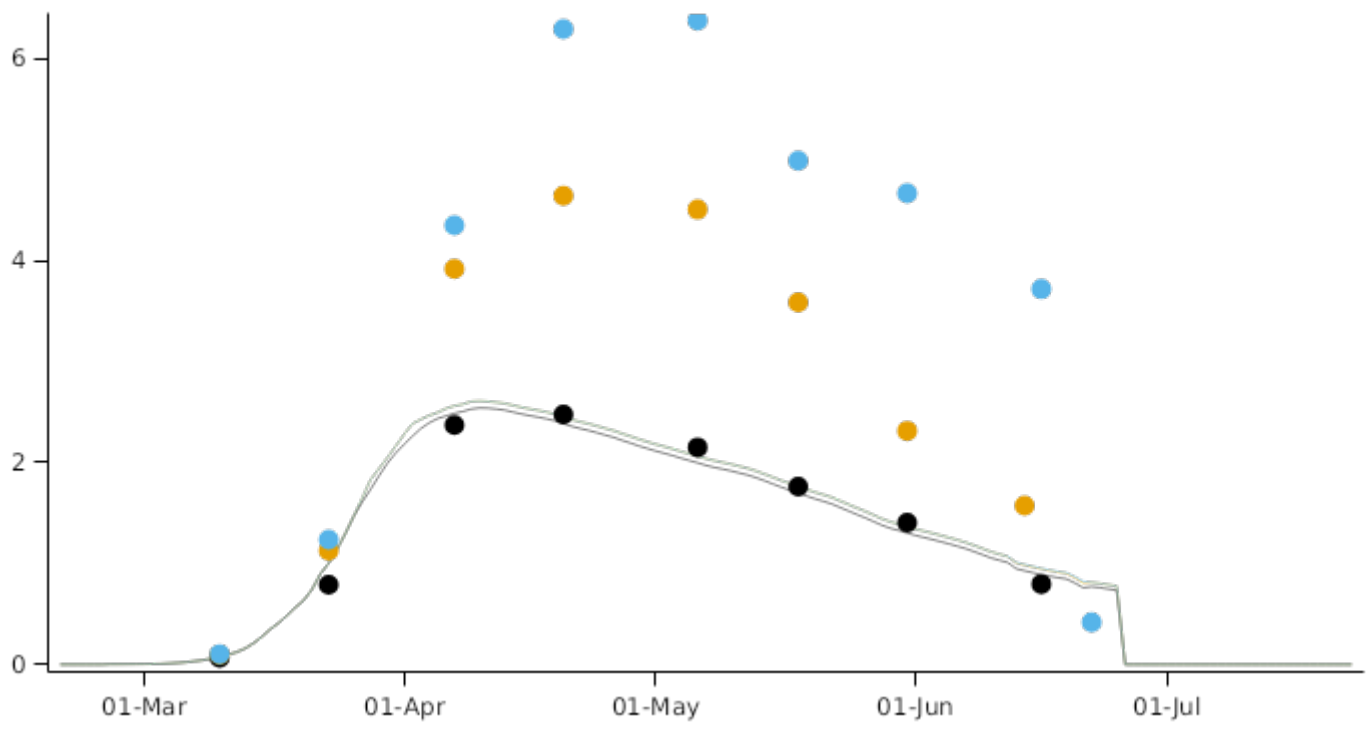
### BiomassN



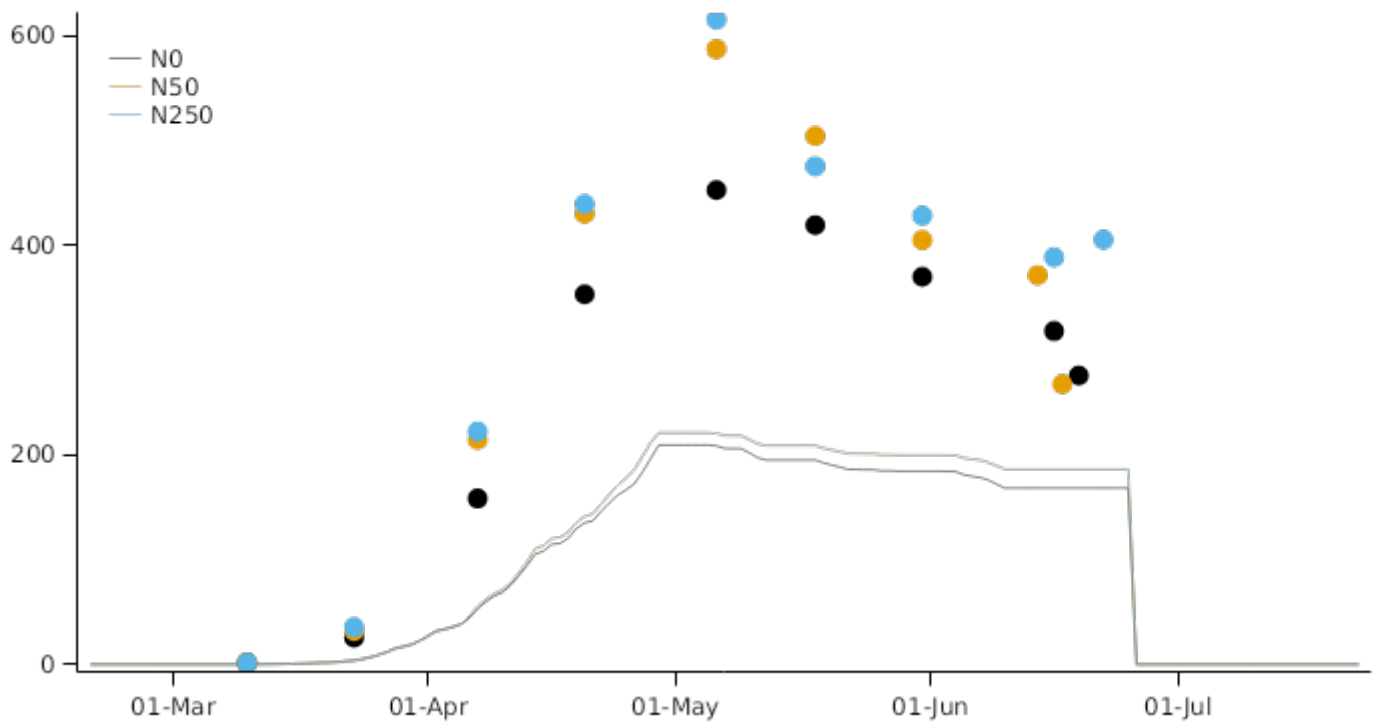
### Leaf



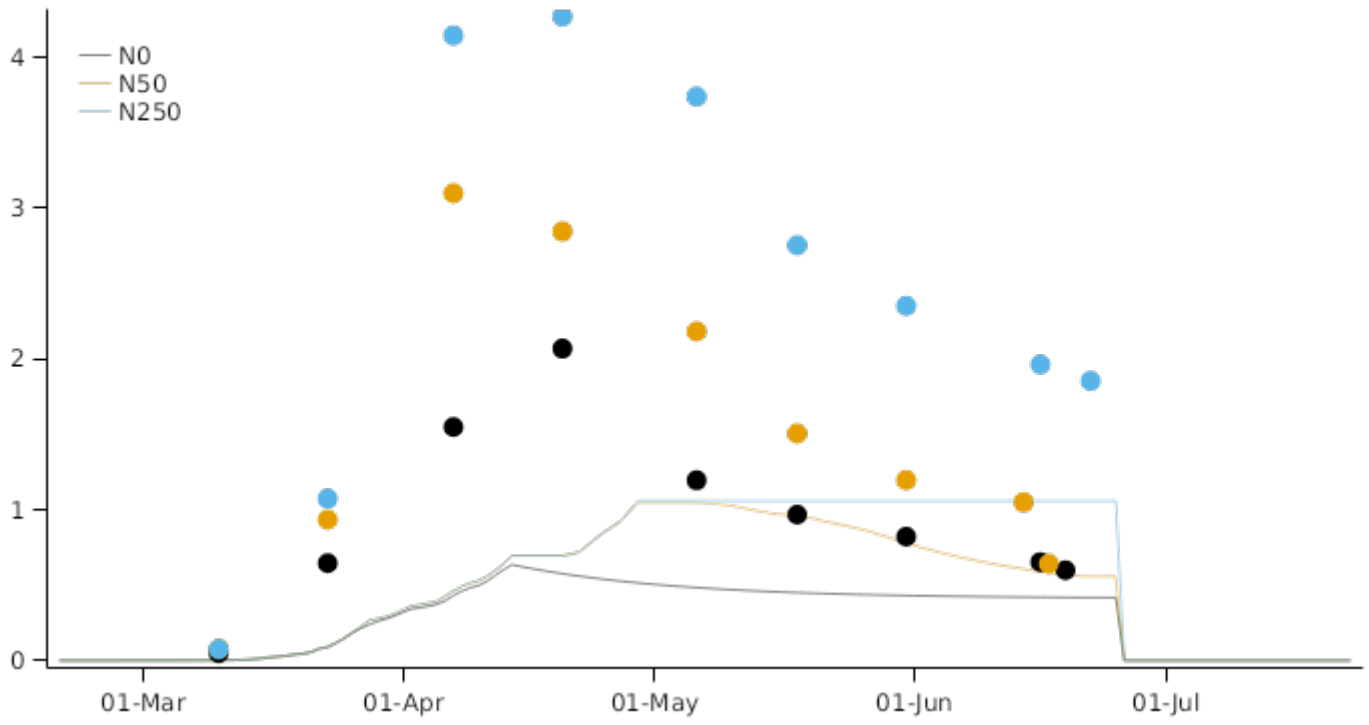
### LeafN



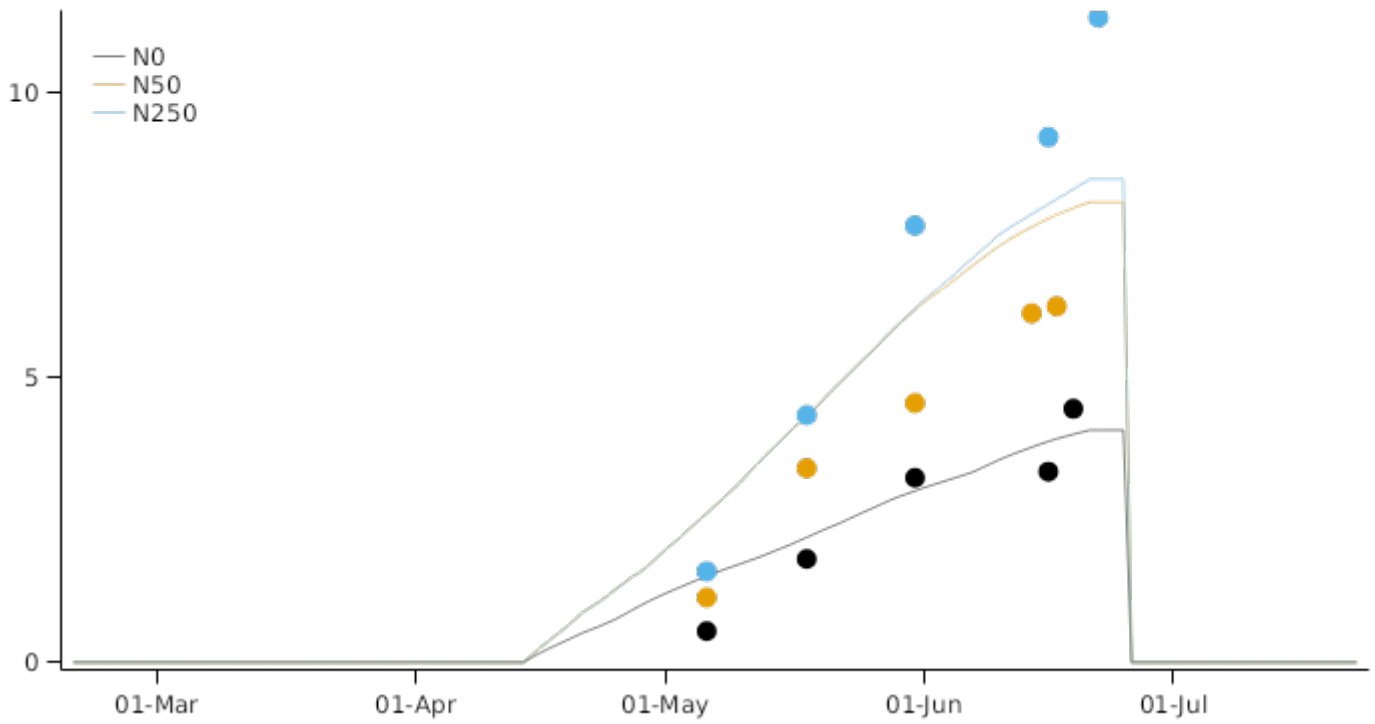
### Stem



### StemN

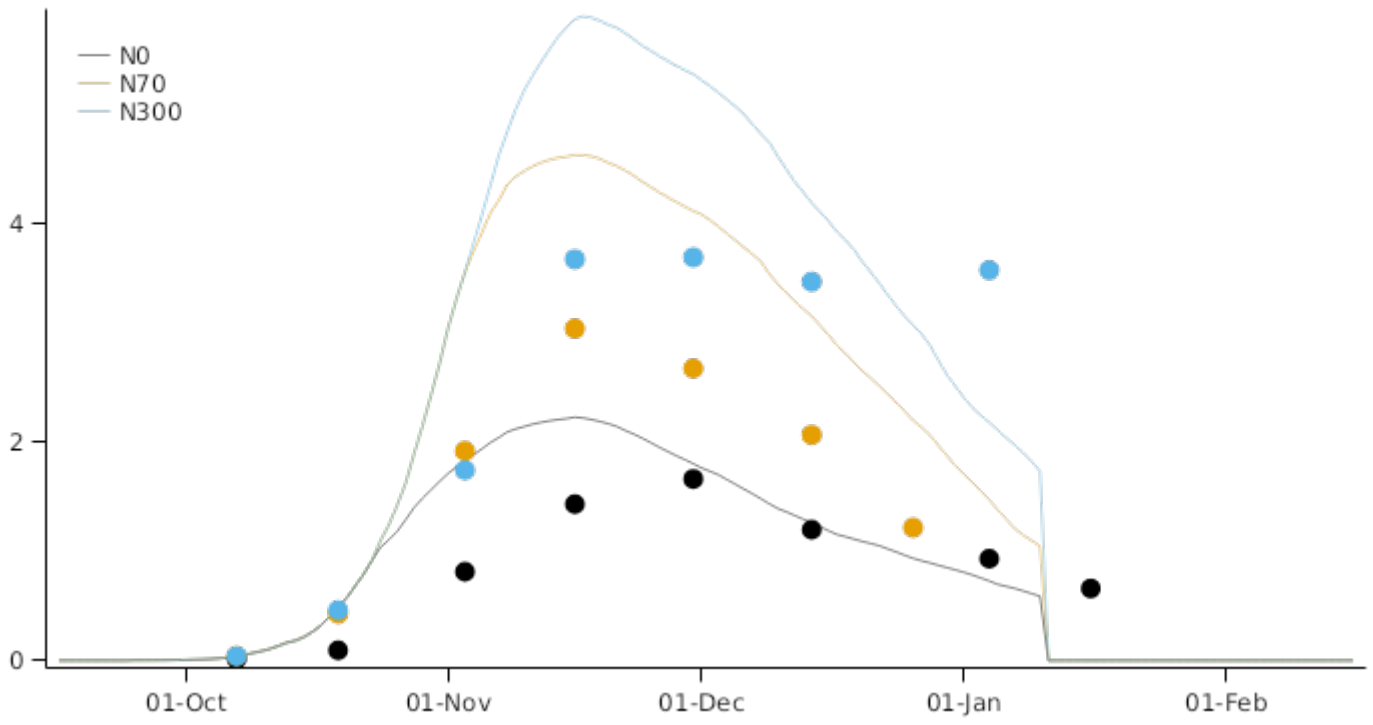


### GrainN

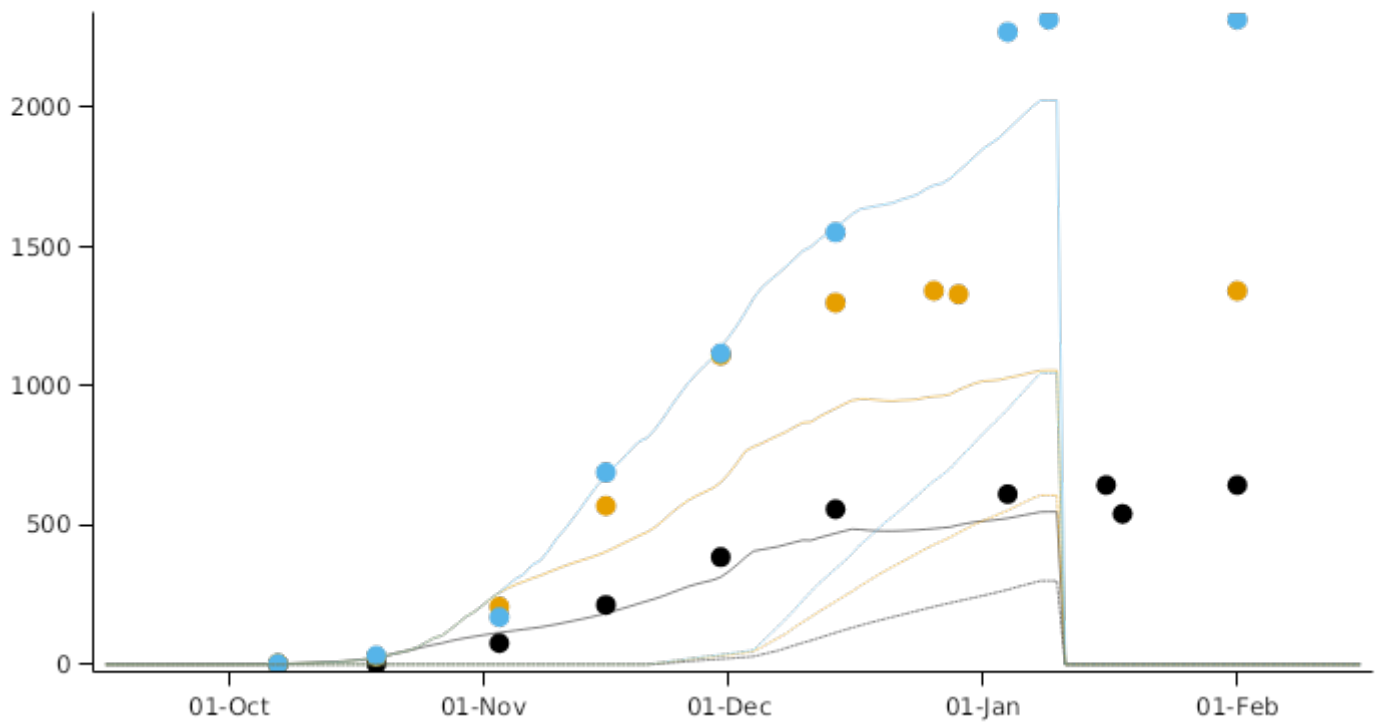


#### 2.3.4 Angelo99

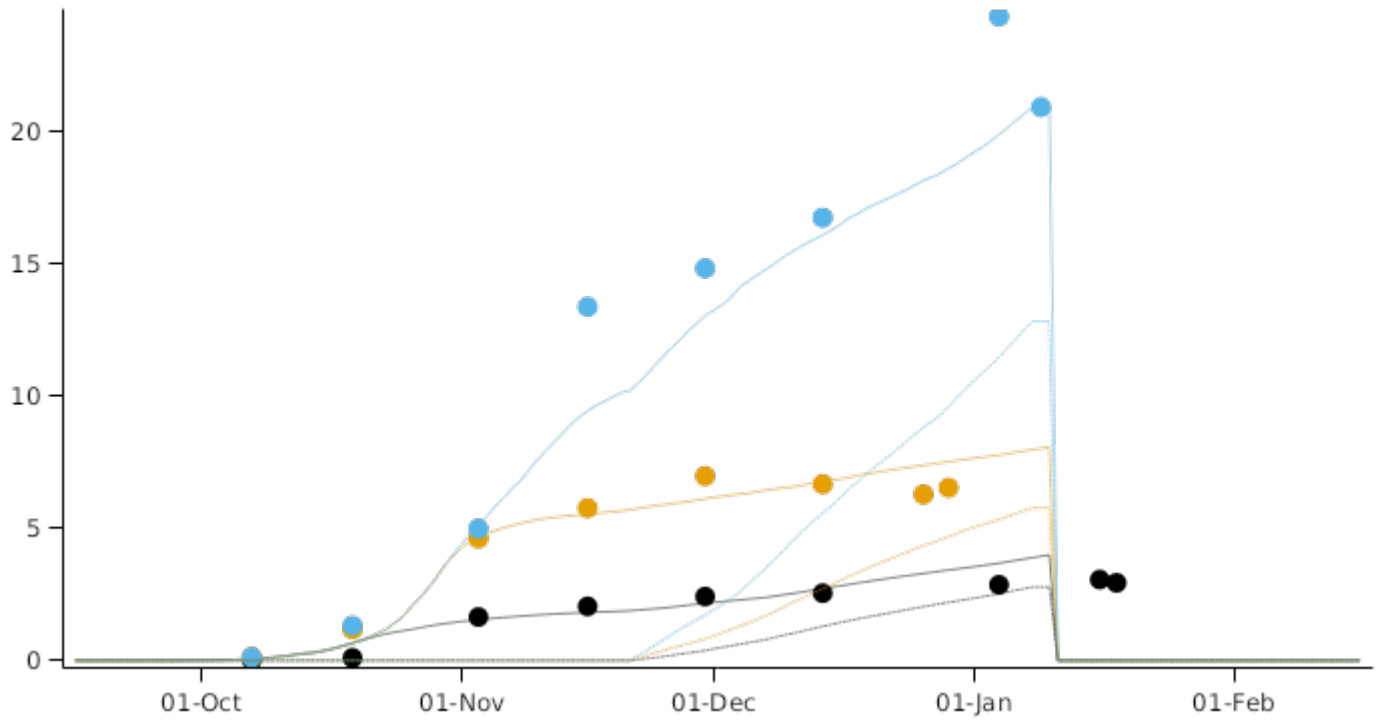
### LAI



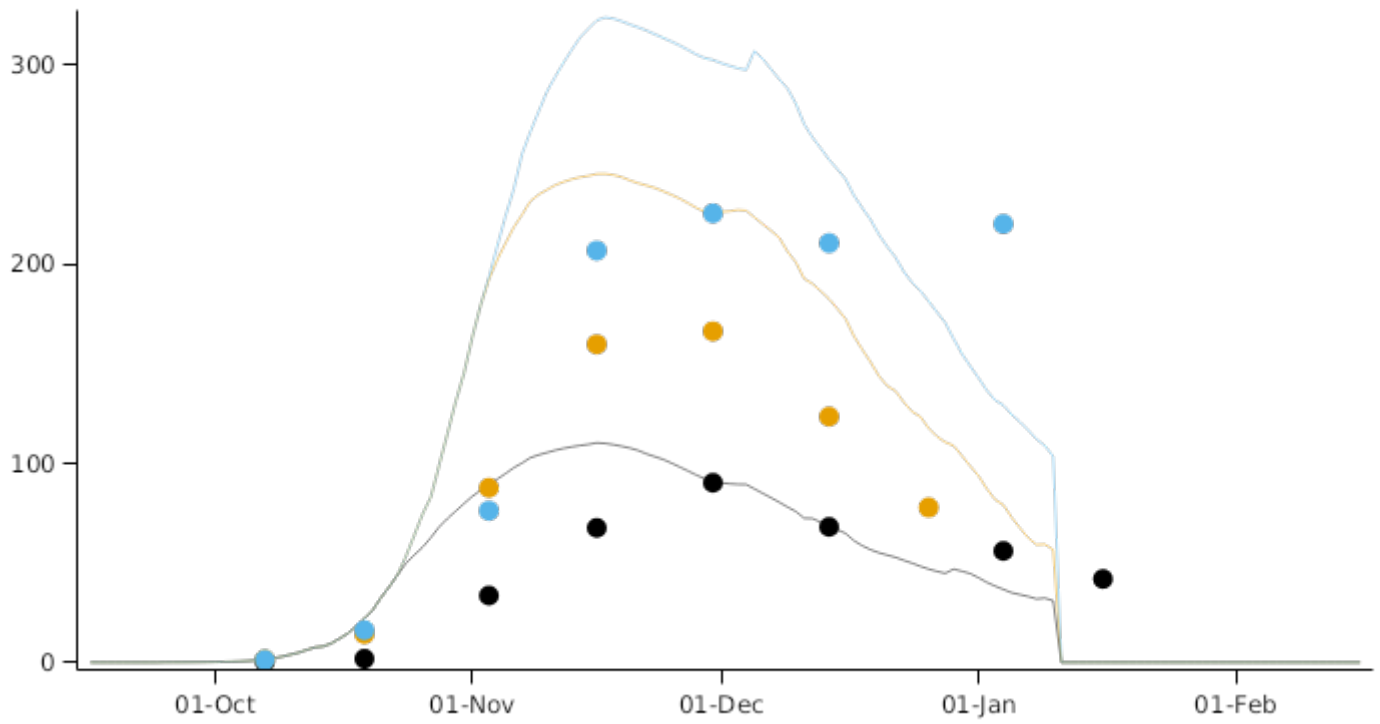
### Biomass



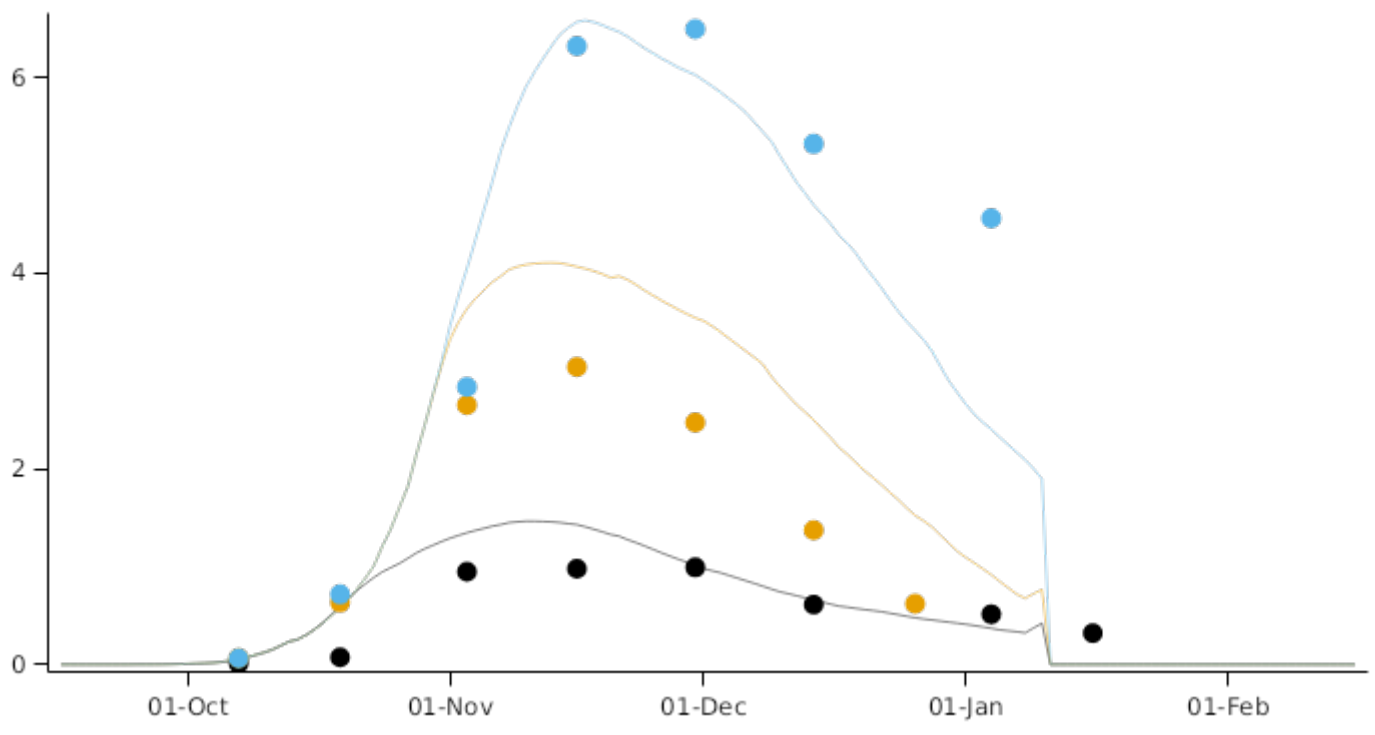
### BiomassN



### Leaf

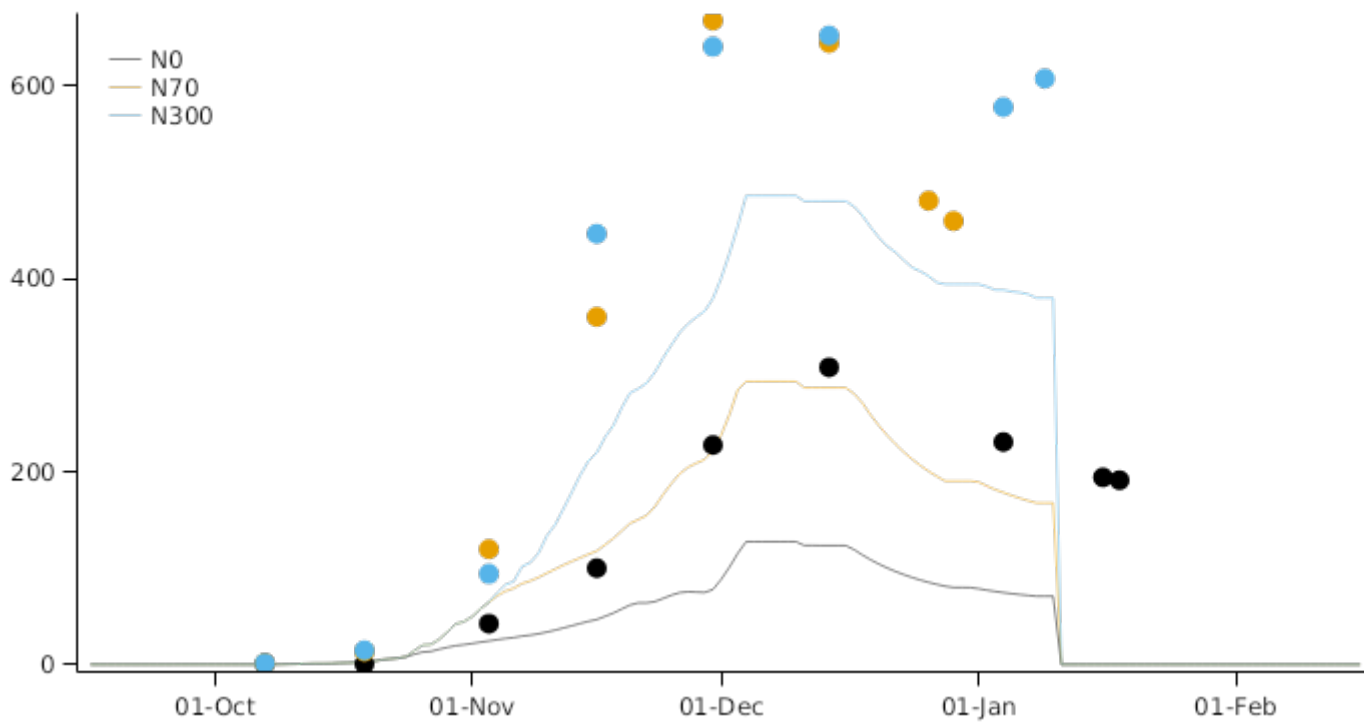


# LeafN

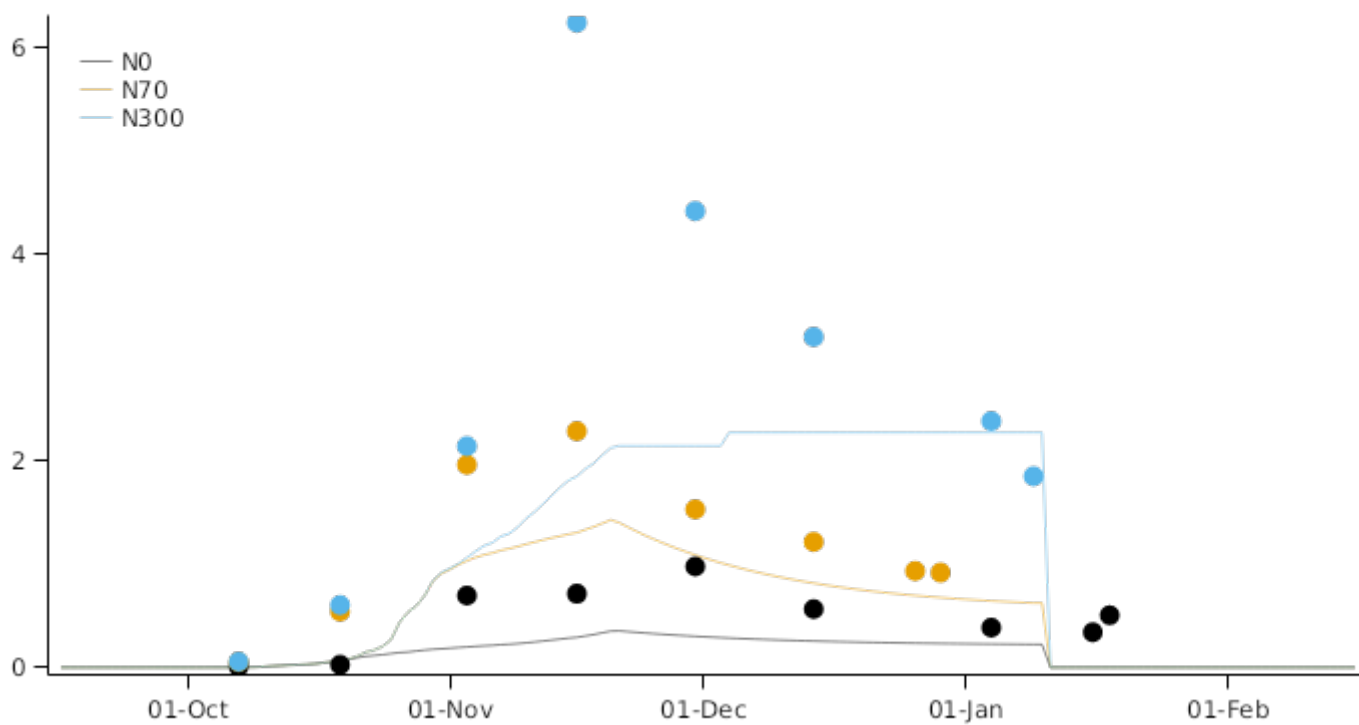


# WaterStress

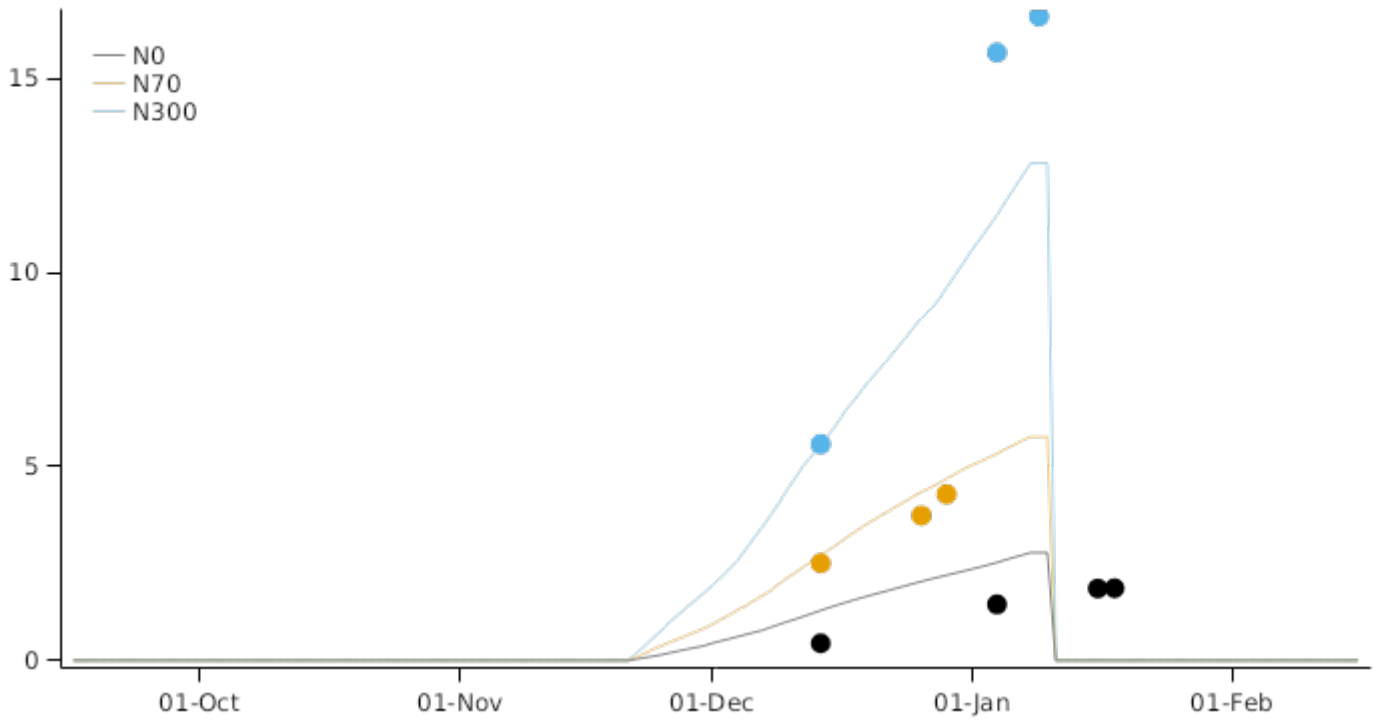
### Stem



### StemN

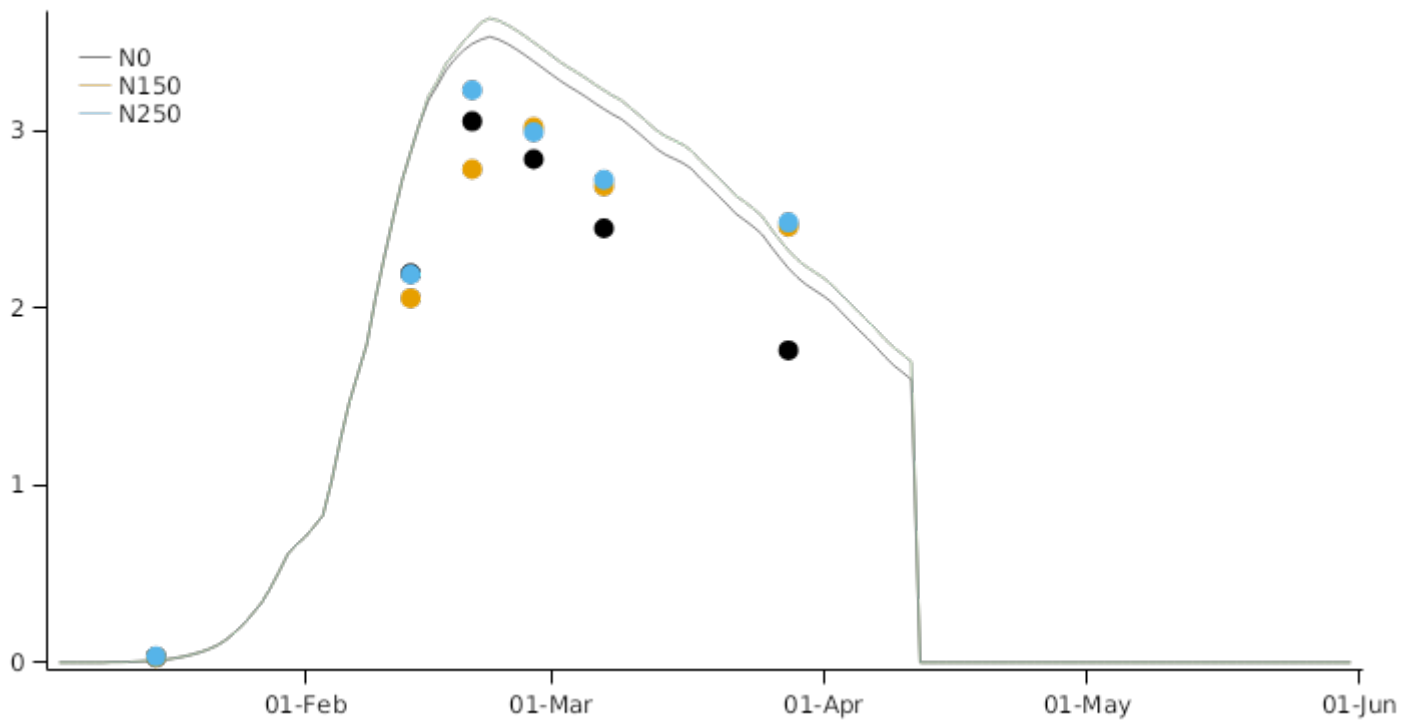


### GrainN



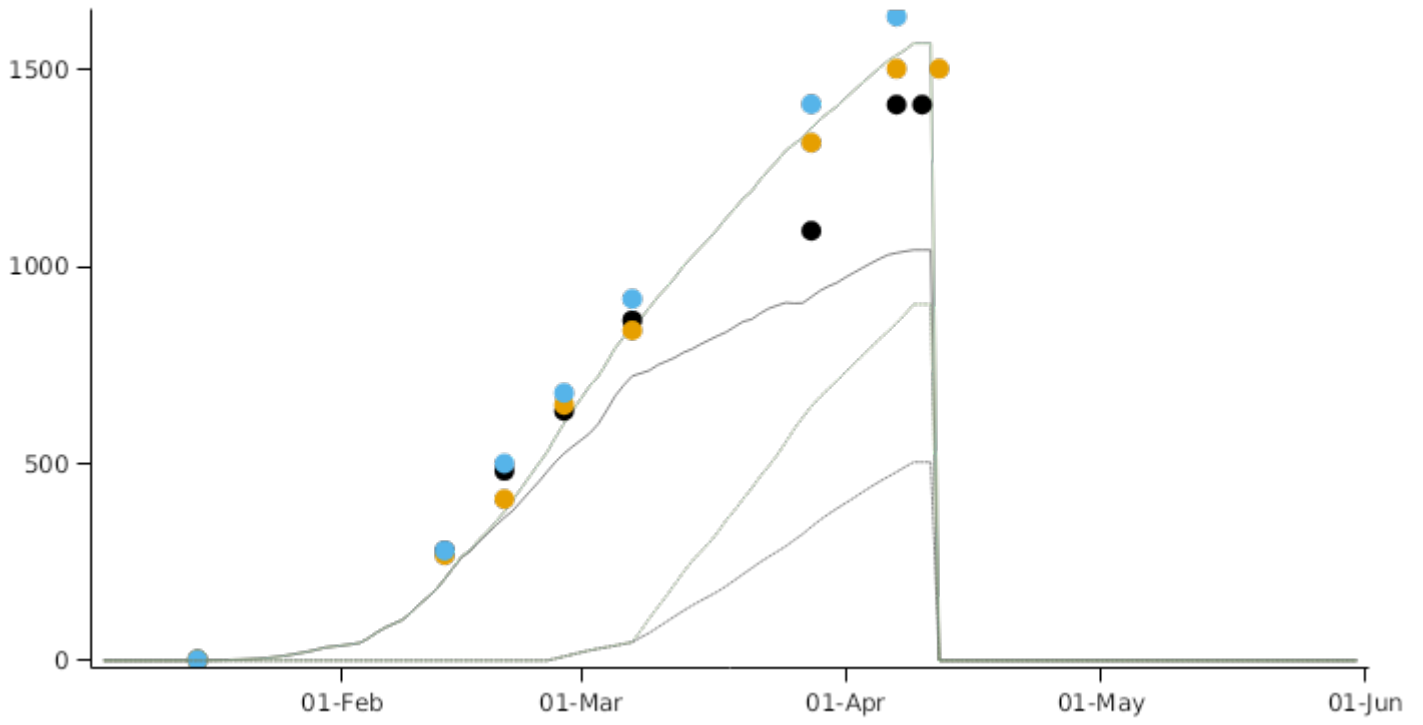
### 2.3.5 Angelo00

### LAI

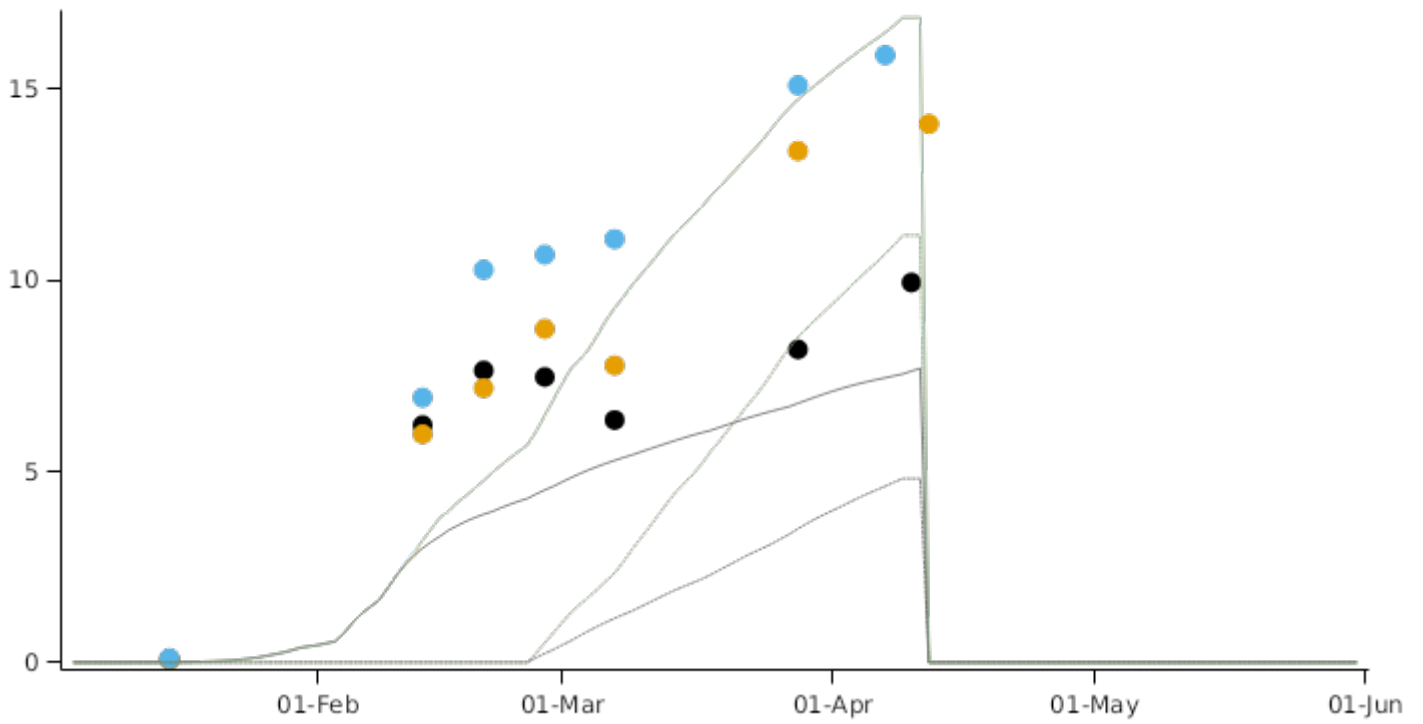




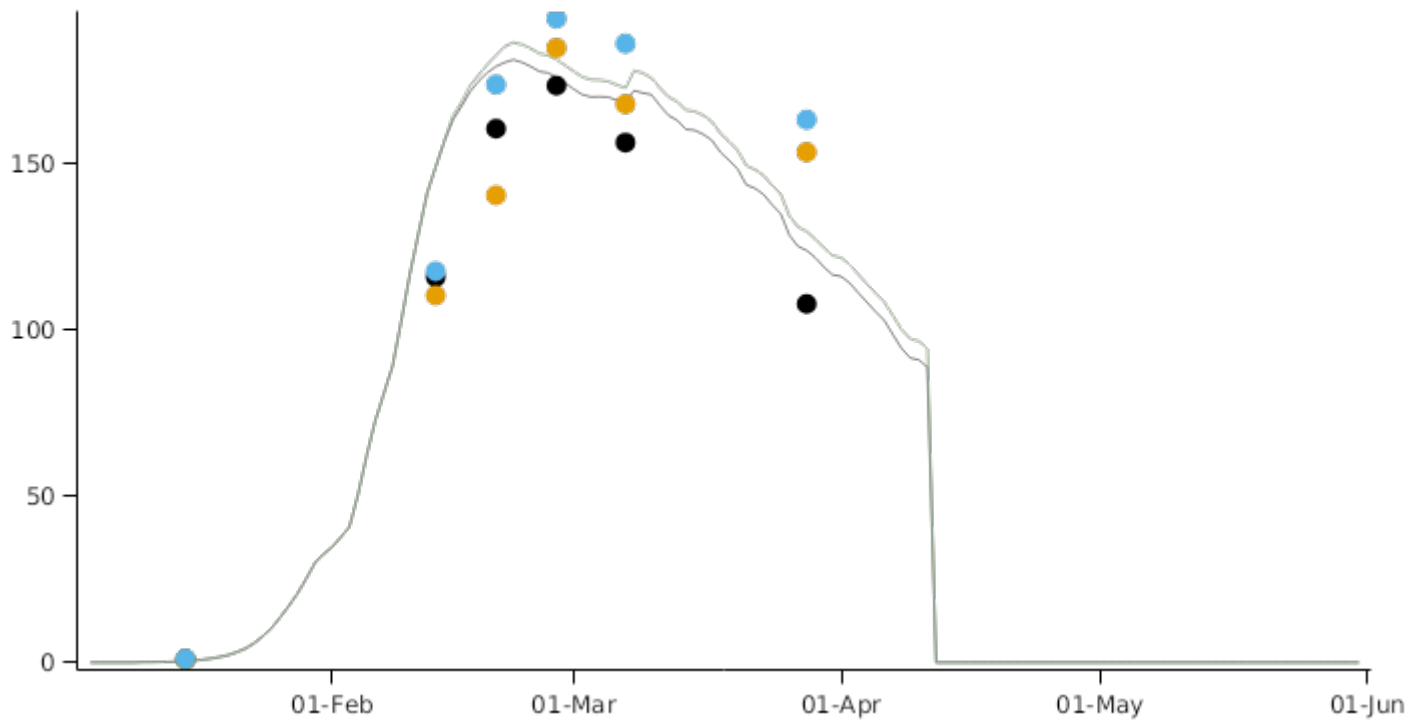
### Biomass



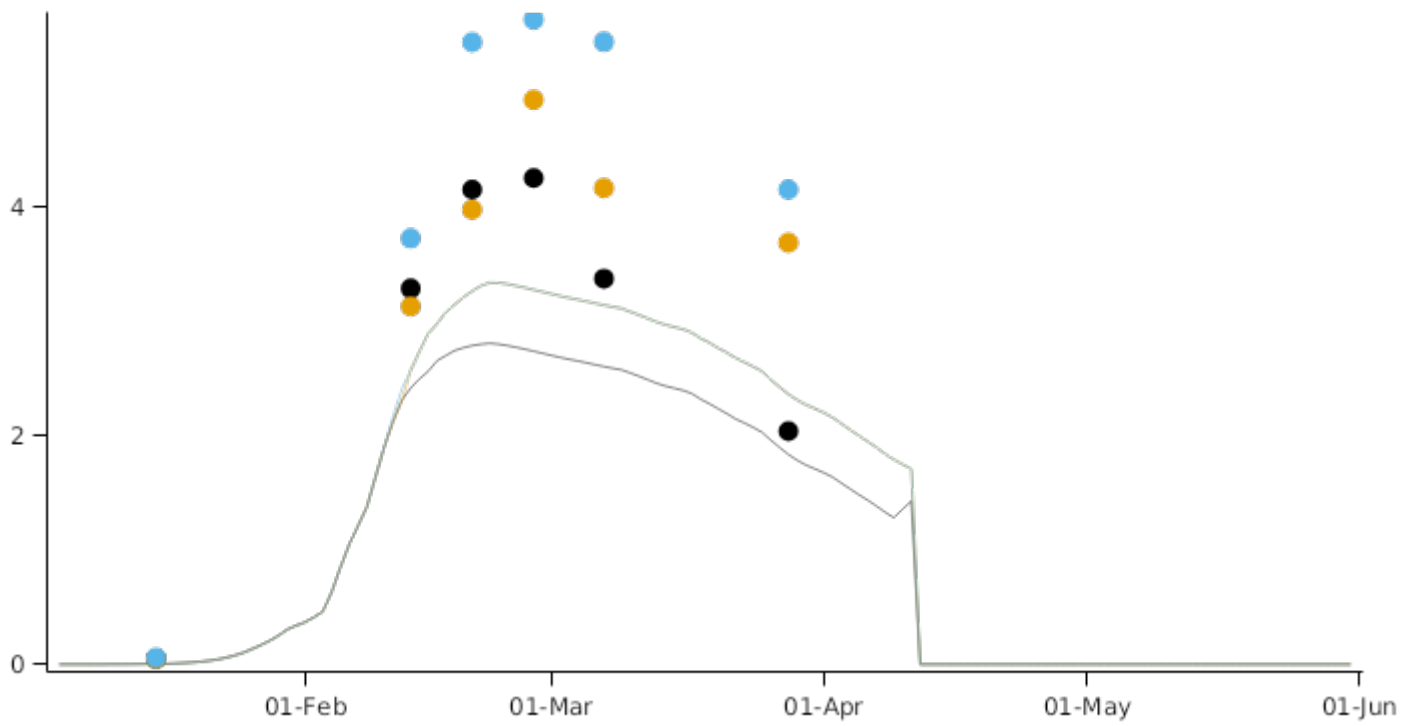
### BiomassN



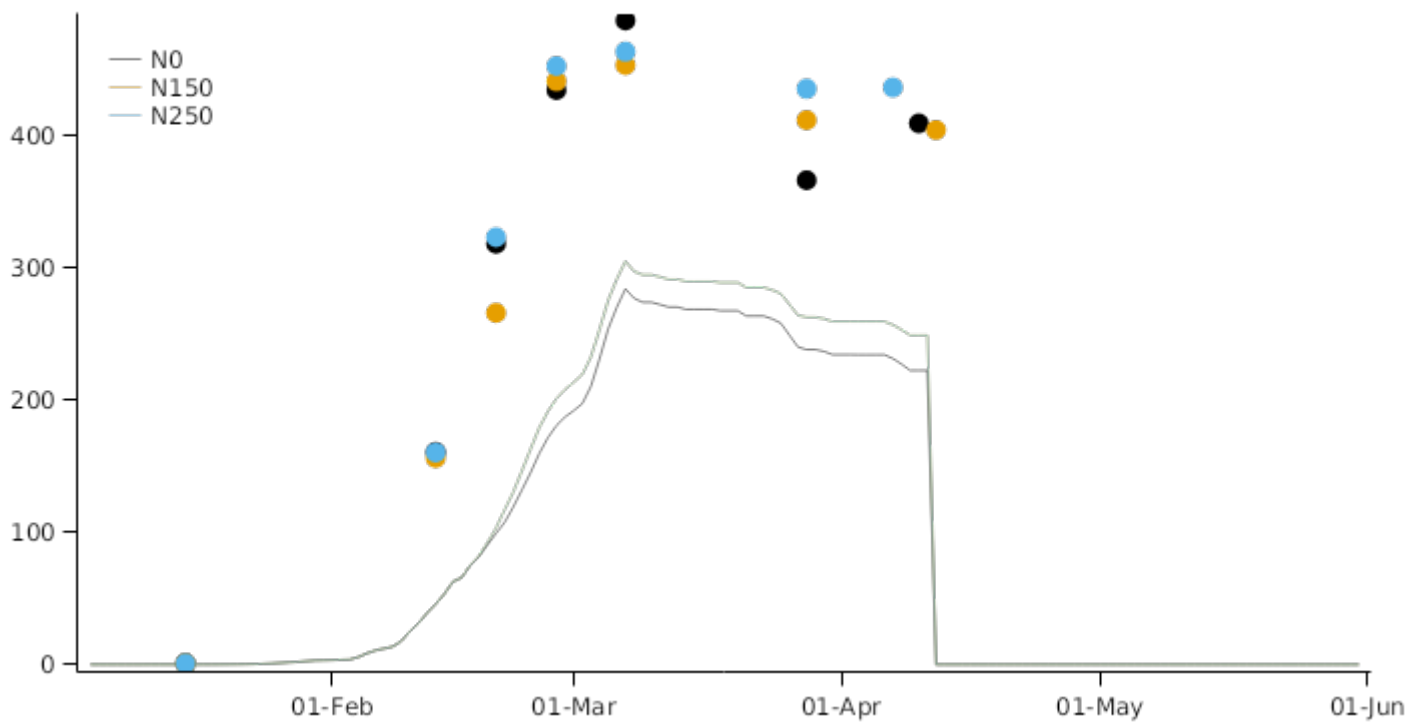
### Leaf



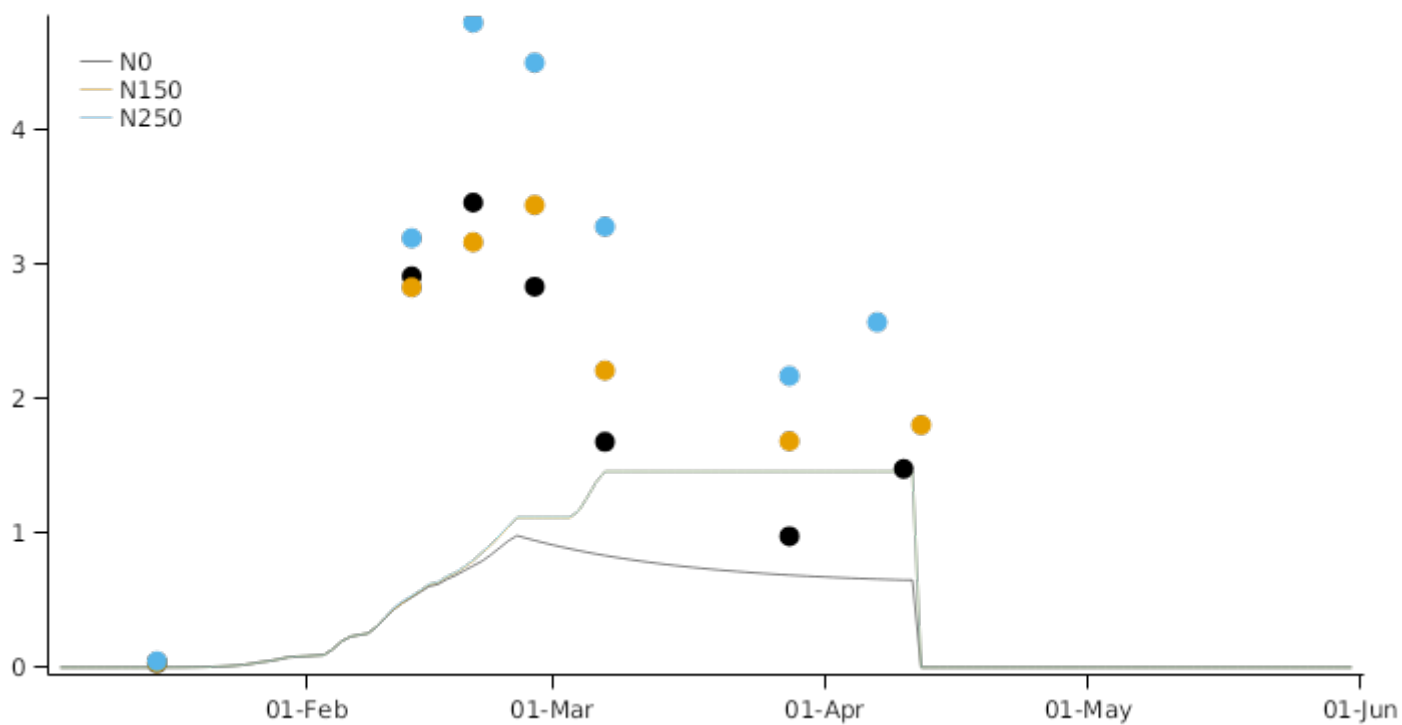
### LeafN



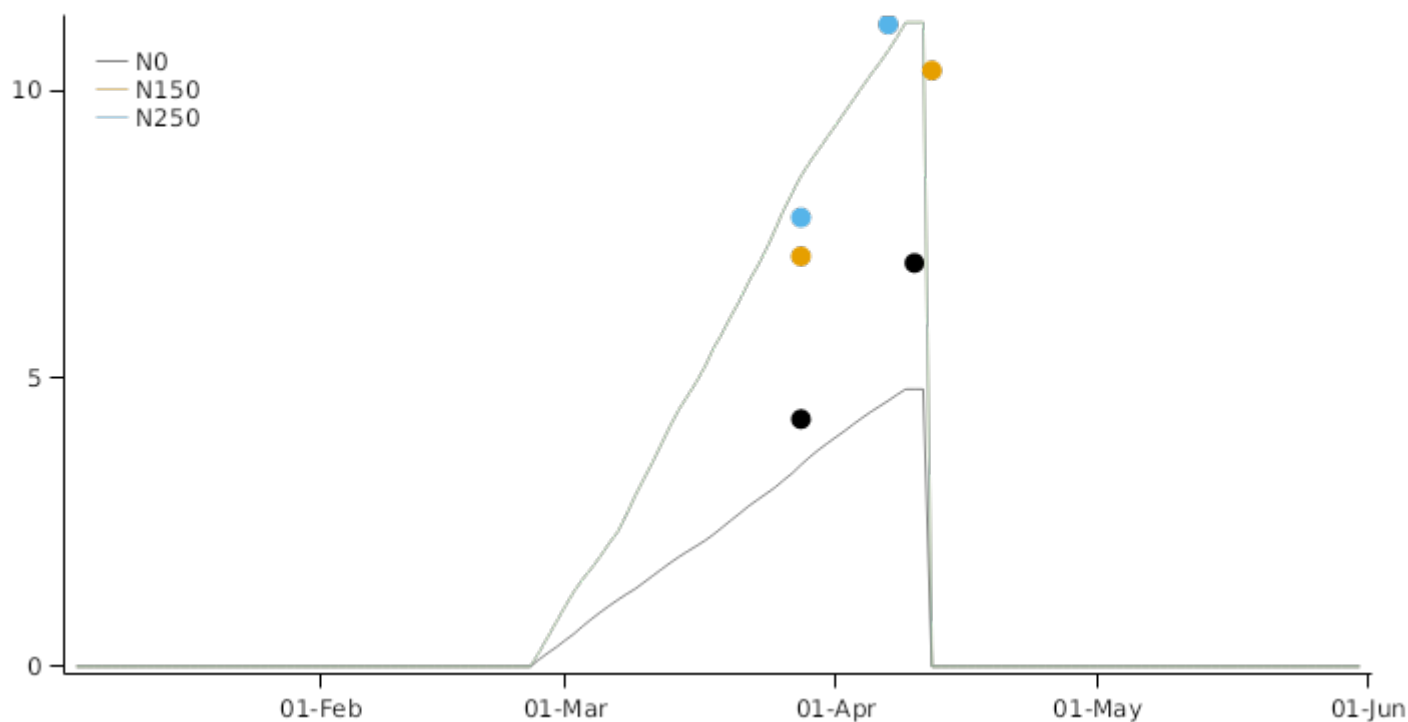
### Stem



### StemN



## GrainN



## 2.4 USA

Testing of APSIM Maize, Soil Water, Soil Nitrogen, Manure, and Soil Temperature Modules in the Midwestern United States was undertaken by [Archontoulis\_etal\_2014] using APSIM version 7.4. A subset of these data has been chosen for testing of the maize model. Datasets involving application of compost have not been included so as to concentrate on tests of maize model performance, rather than overall soil organic matter dynamics.

### List of experiments.

Experiment Name	Design (Number of Treatments)
Ames2011	NRate (3)

## 2.5 New Zealand

Testing of APSIM Maize under New Zealand conditions to determine the performance of the model under temperate conditions with cool springs and mild summers. A range of trials have been conducted and are detailed below

### List of experiments.

Experiment Name	Design (Number of Treatments)
Lincoln2012	Nit x Irr (6)
Lincoln2010	Nitrogen (2)
Lincoln2008	Sow x Cover (8)
HawksBay2010	Sow x Cv (9)

### 2.5.1 Lincoln2012

#### 2.5.1.1 Lincoln2012 (Rain-Shelter Trail)

Testing of APSIM Maize under New Zealand conditions was undertaken using the data of [Teixeira et al., 2014](#). This dataset includes the impact of three N (0 to 250 kg/ha N) and two water regimes (dryland and fully irrigated) using a rain-shelter structure. Observations include biomass growth and nitrogen content of individual organs, soil water contents, leaf area index, phenology and yield components. Total biomass ranged from 8000 kg/ha for dryland nil N crops to up to 28000kg/ha for fully irrigated and N fertilised crops. Dryland crops recovered 25 percent less N from applied fertilizer than irrigated crops.

## 2.5.2 Lincoln2010

### 2.5.2.1 Lincoln2010 (Leaf properties trial)

This trial was conducted to provide data for parameterising the Leaf Area model and is described in detail by [Teixeira et al., 2011](#). Detailed measurements were conducted on leaf appearance and expansion with some measurements of biomass accumulation. Treatments of plus and minus Nitrogen were applied but there was sufficient N in the soil than N responses were very small.

## 3 Sensibility Tests

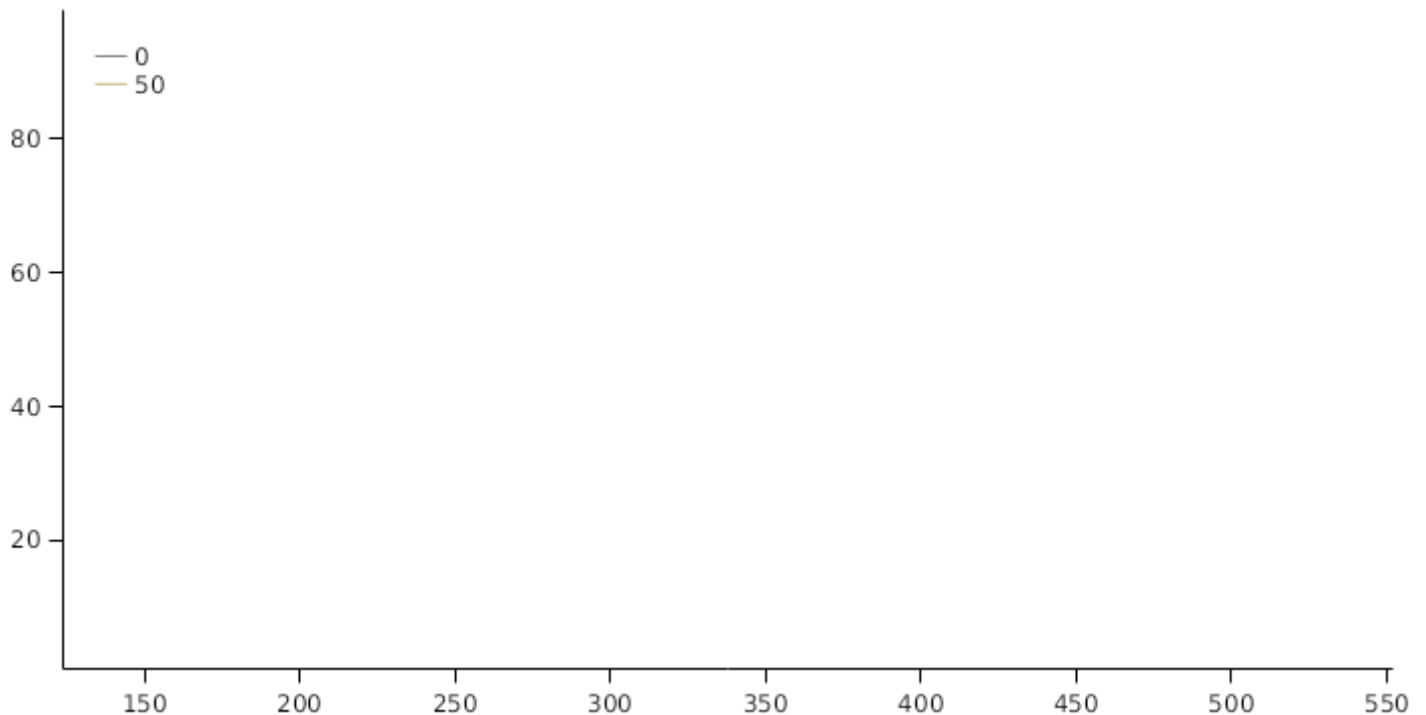
List of experiments.

Experiment Name	Design (Number of Treatments)
Bugesera	NRate (2)

### 3.1 Bugesera

Maize is grown in the Bugesera region of southern Rwanda in Central Africa. The region has a bimodal rainfall distribution which allows two plantings per year. For low input situations the maize yields should vary between 1 and 3 t/ha per crop. Under fertilised conditions the yield should increase up to 5 t/ha per crop.

### Yield CDF



## 4 References

- [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.](#)
- CARBERRY, PS, ABRECHT, DG, 1991. TAILORING CROP MODELS TO THE SEMIARID TROPICS. CLIMATIC RISK IN CROP PRODUCTION : MODELS AND MANAGEMENT FOR THE SEMIARID TROPICS AND SUBTROPICS, Eds: MUCHOW, RC and BELLAMY, JA, 157-182.
- CARBERRY, PS, MUCHOW, RC, MCCOWN, RL, 1989. TESTING THE CERES-MAIZE SIMULATION-MODEL IN A SEMI-ARID TROPICAL. *FIELD CROPS RESEARCH* 20 (4), 297-315.
- [Hammer, Graeme L., Dong, Zhanshan, McLean, Greg, Doherty, Al, Messina, Carlos, Schussler, Jeff, Zinselmeier, Chris, Paszkiewicz, Steve, Cooper, Mark, 2009. Can Changes in Canopy and/or Root System Architecture Explain Historical Maize Yield Trends in the U.S. Corn Belt?. \*Crop Sci\* 49 \(1\), 299-312.](#)
- [Jones, C.A., Kiniry, J.R., Dyke, P.T., 1986. CERES-Maize: a simulation model of maize growth and development.](#)

- KEATING, BA, WAFULA, BM, WATIKI, JM, 1992. DEVELOPMENT OF A MODELING CAPABILITY FOR MAIZE IN SEMIARID EASTERN KENYA. ACIAR PROCEEDINGS SERIES, 26-33.
- Lawless, Conor, Semenov, MA, Jamieson, PD, 2005. A wheat canopy model linking leaf area and phenology. *European Journal of Agronomy* 22 (1), 19-32.
- Massignam, AM, Chapman, SC, Hammer, GL, Fukai, S, 2009. Physiological determinants of maize and sunflower grain yield as affected by nitrogen supply. *Field crops research* 113 (3), 256-267.
- 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.
- 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.
- Teixeira, E. I., George, M., Brown, H. E., Fletcher, A. L., 2011. A framework for quantifying maize leaf expansion and senescence at the individual leaf level. *Agronomy New Zealand* 41, 59-65.
- Teixeira, Edmar I., George, Michael, Herreman, Thibault, Brown, Hamish, Fletcher, Andrew, Chakwizira, Emmanuel, de Ruiter, John, Maley, Shane, Noble, Alasdair, 2014. The impact of water and nitrogen limitation on maize biomass and resource-use efficiencies for radiation, water and nitrogen. *Field Crops Research* 168, 109-118.