



1 The APSIM Potato Model

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

1.1 Building the model.

The APSIM potato model has been described in part by [Brown et al., 2011](#) and 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 consists of:

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

1.2 Peculiarities of potato

Potato is a challenging crop to model because it is vegetatively propagated. For seed propagated crops, plants are always equally dormant at planting and their development begins from there. With potato the physiological status of the seed tubers vary depending on:

- * the conditions the mother plant encountered late in its life (Stress, Desiccation etc)
- * the conditions that the tuber was stored in (Cool store, ground store, dry store etc)
- * treatment of the seed prior to planting (chitting, cutting etc).

All of these factors influence the developmental behaviour of the crop that is produced including:

- * The timing of emergence
- * The number of stems produced per plant and subsequent tuber number and canopy development
- * The timing of tuber initiation and subsequent growth and partitioning
- * The duration of haulm growth and subsequent tuber filling duration

There are an enormous number of potato cultivars and they all differ in their responses of post planting behaviour due to pre planting seed conditions, there is not a good quantitative understanding of how environment and management of seed potato prior to planting influences post planting behaviour and it is beyond the scope of the current APSIM infrastructure to provide environmental conditions during storage of seed. We have tried but with a list of excuses this long we have not yet succeeded in building the effects of seed physiology into the potato model in anything other than a very elementary way. However, it is possible to set the model up to capture seed effects for specific simulations.

1.3 Including a potato crop in an APSIM simulation

There is a Potato example simulation available by clicking the "Open an Example" icon available on the first tab displayed when APSIM Next Gen is opened. This provides a useful demonstration of how to simulate a potato crop and provides some useful graphs for viewing models predictions.

To include a potato crop in a simulation the "Potato" model needs to be added to the paddock, field or zone in which it is to be grown. This can be done by right clicking on the "Paddock", selecting "Add" from the drop down menu and then selecting "Potato" from the list that comes up. A sowing action also needs to be set to start the crop off and a harvest event to remove the potato crop so the next crop can be simulated. The "PotatoPlantAndHarvest" manager module that is included in the "Potato" example simulation has script written to instigate these events and can be copied from the example simulation into other simulations.

1.4 Setting sowing depth (emergence date)

The potato model predicts emergence date as a function of sowing depth and all subsequent phenology is dependent on this timing. Because this date is effected by seed physiology and can't yet be modeled we recommend that planting depth

is not set at the actual planting depth, but varied to ensure the model predicts a realistic emergence date. Planting depth is set in the 'PotatoPlantAndHarvest' module

1.5 Setting stem number (tuber number and early canopy growth)

The 'PotatoPlantAndHarvest' script sets the "Number of stems per seed piece". This varies enormously due to seed physiology and variety and should be set to a number that is realistic for the cultivar being simulated and the pre-planting treatment of the seed tubers. This parameter influences tuber number and the early development of the canopy so should be set carefully. **##Setting final leaf number (Crop duration)** The duration of a potato crop is determined by the number of leaves that the mainstem produces and how long it takes for these to appear and senesce. This responds to photoperiod in many varieties and is influenced by the pre-planting treatment of seed tubers. As such this parameter is given a default value for each cultivar but can also be set in the "PotatoPlantAndHarvest" script if more is know about the expected determinany of the potatoes under the circumstances being simulated. Please read [Brown et al., 2011](#) for a clear definition of final leaf number.

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
Phenology	Models.PMF.Phen.Phenology
Arbitrator	Models.PMF.OrganArbitrator
Structure	Models.PMF.Struct.Structure
Tuber	Models.PMF.Organs.GenericOrgan
Root	Models.PMF.Organs.Root
Leaf	Models.PMF.Organs.Leaf
Stem	Models.PMF.Organs.GenericOrgan
MortalityRate	Models.Functions.Constant

1.6 Phenology

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

1.6.1 ThermalTime

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

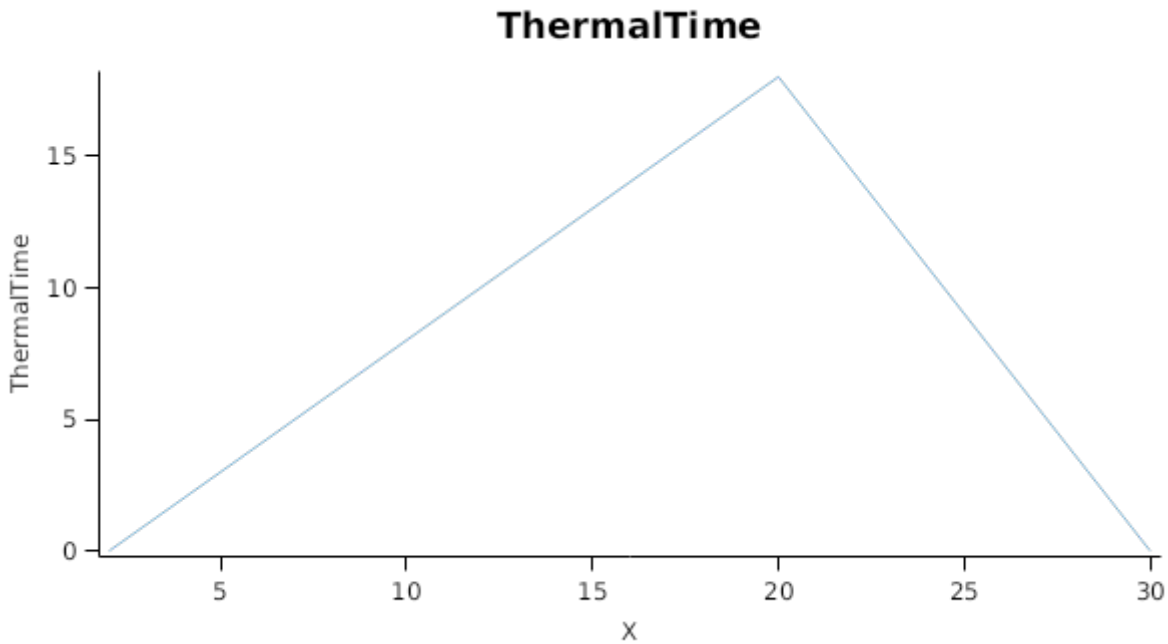
Thermal time drives progress through all phenological phases, including those which are driven by leaf apparence and senescence. The base temperature of 2 degrees was determined by [Brown et al., 2011](#) and [Oliveira, 2015](#) and the upper threshold of 30 degrees is from [Van Keulen et al., 1995](#). The optimum of 20 degrees was determined in an unpublished model fitting exercise (Brown pers com).

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 ThermalTime

X	ThermalTime
2.0	0.0
20.0	18.0

X	ThermalTime
30.0	0.0



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

Phase Number	Phase Name	Initial Stage	Final Stage
1	Dormant	Planting	Peeping
2	Sprouting	Peeping	Emergence
3	Vegetative	Emergence	TuberInitiation
4	EarlyTuber	TuberInitiation	FinalLeaf
5	LateTuber	FinalLeaf	FullSenescence
6	Senesced	FullSenescence	Maturity
7	Maturity	Maturity	Eternity

1.6.2 Dormant

This phase goes from planting to peeping.

This phase is currently set with a very short duration and is a place holder until we have a suitable method for modelling tuber dormancy

The *Target* for completion is calculated as:

$$\text{Target} = 1 \text{ (oD)}$$

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

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

1.6.3 Sprouting

This phase goes from peeping 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:

$$\text{ShootRate} = 1.35 \text{ (deg day/mm)},$$

ShootLag = 72 (deg day),

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

The sprouting phase controls when the crop emerges. In potato this is influenced by the physiological state of the seed tuber when it is planted. This is difficult to model as the seed tubers physiological status begins when it is still on the mother plant and is influenced by the treatment of the mother plant, the storage conditions of the tuber, conditions when the seed is being prepared for planting and the variety of potato. There is currently no good way of modelling this as it would require environmental information from beyond the field where the crop is to be grown which is currently beyond the scope of APSIM. In the absence of something more elegant, we recommend that the user adjusts sowing depth to achieve a realistic emergence date and let the model deal with phenological development there after.

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

ThermalTime = [*Phenology*].*ThermalTime*

1.6.4 Vegetative

This phase goes from emergence to tuberinitiation.

During the *Vegetative Phase* the crop is only growing stem, root and leaf. Tuber initiation usually occurs soon after *Emergence* and the duration of this phase can be influenced by the physiological status of the seed tuber at planting. As discussed above, we do not have a suitable method for modelling this yet so we have treated the time from emergence to tuber initiation as a thermal time constant.

The *Target* for completion is calculated as:

Target = 300 (oD)

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

Progression = [*Phenology*].*ThermalTime*

1.6.5 EarlyTuber

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

The *FinalLeafNumber* responds to photoperiod in many varieties and is influenced by the pre-planting treatment of seed tubers. Therefore *FinalLeafNumber* is given a default value for each cultivar but can also be set in the "PotatoPlantAndHarvest" script if more is know about the expected determinany of the potatoes under the circumstances being simulated.

FinalLeafNumber = [*Structure*].*FinalLeafNumber*

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

FullyExpandedLeafNo = [*Leaf*].*ExpandedCohortNo*

InitialisedLeafNumber = [*Leaf*].*InitialisedCohortNo*

1.6.6 LateTuber

The *LateTuber* phase goes from the *FinalLeaf* stage to the *FullSenescence* stage, which occurs when all leaves have fully senesced.

1.6.7 Senesced

This phase goes from fullsenescence to maturity.

During this phase the potato crop has no live shoots above ground but the tubers may be left in the ground to allow their skins to harden so they are not damaged by mechanical digging.

The *Target* for completion is calculated as:

Target = 200 (oD)

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

Progression = [*Phenology*].*ThermalTime*

1.6.8 Maturity

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

1.7 Arbitrator

1.7.1 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.

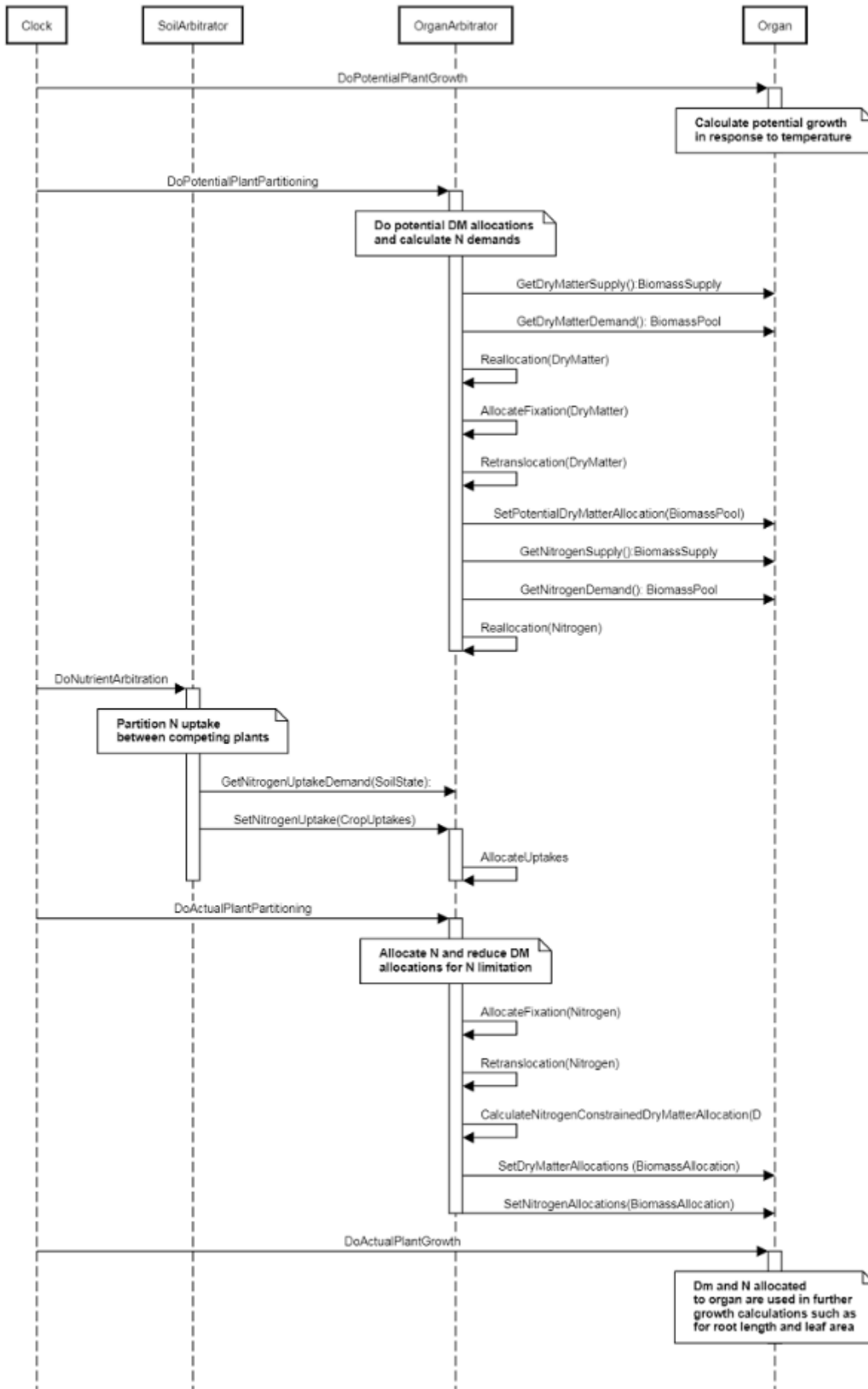


Figure 1: Schematic showing the procedure for arbitration of biomass partitioning. Pink boxes represent events that occur every day and their numbering shows the order of calculations. Blue boxes represent the methods that are called when these events occur. Orange boxes contain properties that make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

1.8 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.

1.8.1 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 Potato is calculated as:

$$\text{MainStemPopn} = \text{Plant.Population} \times \text{PrimaryBudNumber}$$

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

1.8.2 Main-Stem leaf appearance

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

$$\text{LeafTipsAppeared} += \text{DeltaTips}$$

Where *DeltaTips* is calculated as:

$$\text{DeltaTips} = \text{ThermalTime} / \text{Phyllochron}$$

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

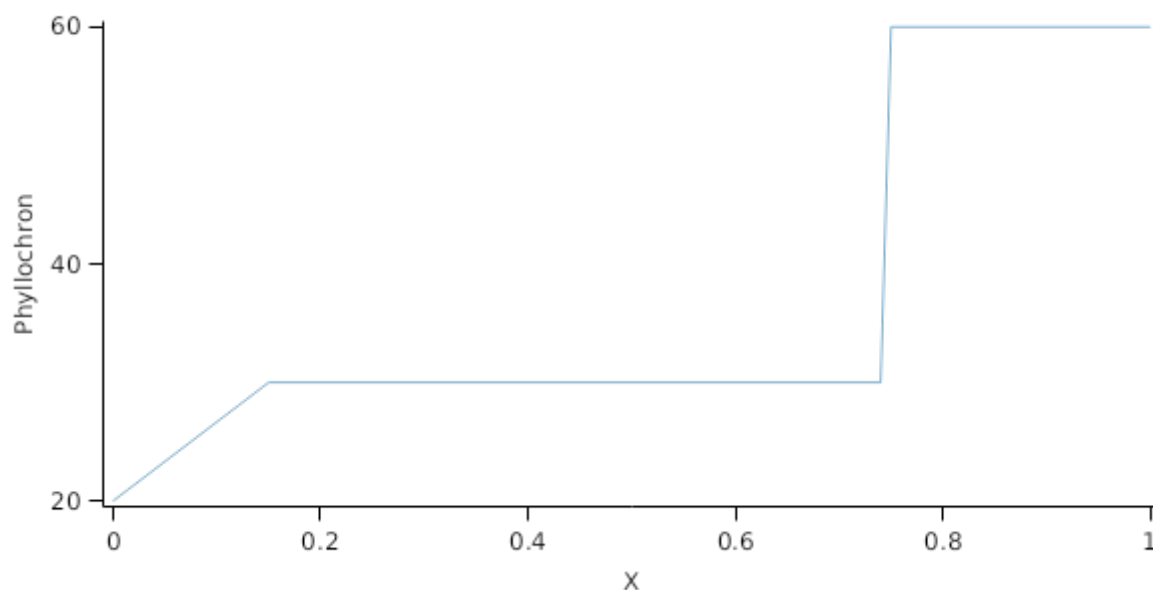
Phyllochron is calculated using specific values or functions for various growth phases. The function will use a value of zero for phases not specified below.

Phyllochron has a value between Emergence and FinalLeaf calculated as:

Phyllochron is calculated using linear interpolation

X	Phyllochron
0.0	20.0
0.1	30.0
0.7	30.0
0.8	60.0
1.0	60.0

Phyllochron



ThermalTime is given by

$ThermalTime = [Phenology].ThermalTime$

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

A default $FinalLeafNumber$ is given for Potato but most cultivars overwrite this with different values and it may also be overwritten in the sowing rule as described in the section "Including a potato crop in an APSIM simulation" above.

$FinalLeafNumber = 34$

1.8.3 Branching and Branch Mortality

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

$TotalStemPopn = MainStemPopn + NewBranches - NewlyDeadBranches$

Where:

$NewBranches = MainStemPopn \times BranchingRate$

$BranchingRate$ is given by:

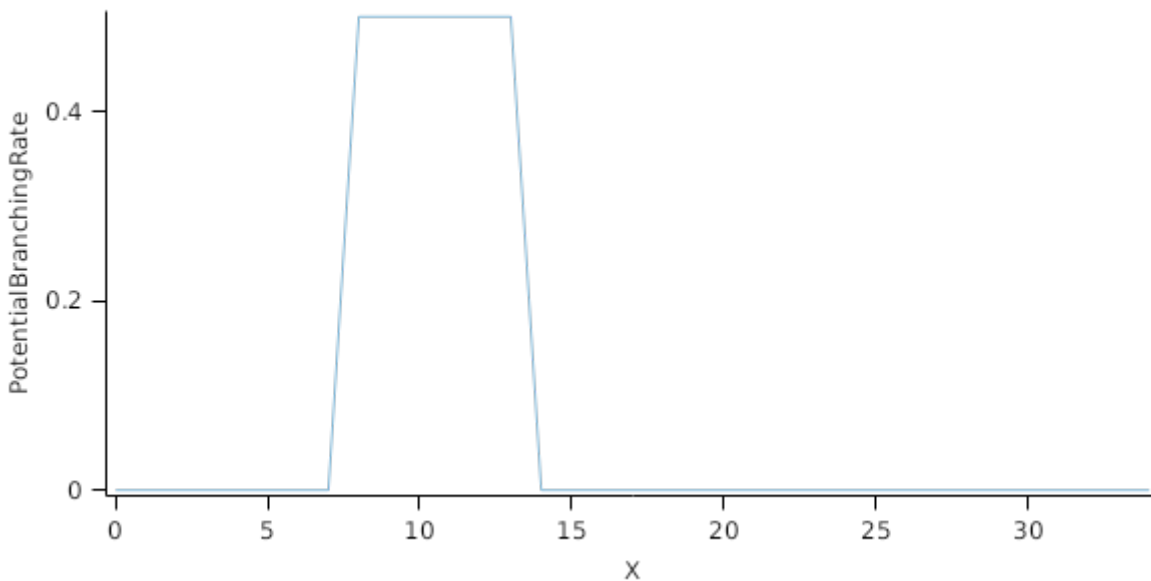
Branching rate is calculated as the product of a potential branching rate, which specifies the branching pattern of widely spaced plants, and a crowding function which decreases branching rate as green cover increases.

$BranchingRate = PotentialBranchingRate \times CrowdingFunction$

$PotentialBranchingRate$ is calculated using linear interpolation

X	PotentialBranchingRate
0.0	0.0
7.0	0.0
8.0	0.5
13.0	0.5
14.0	0.0
34.0	0.0

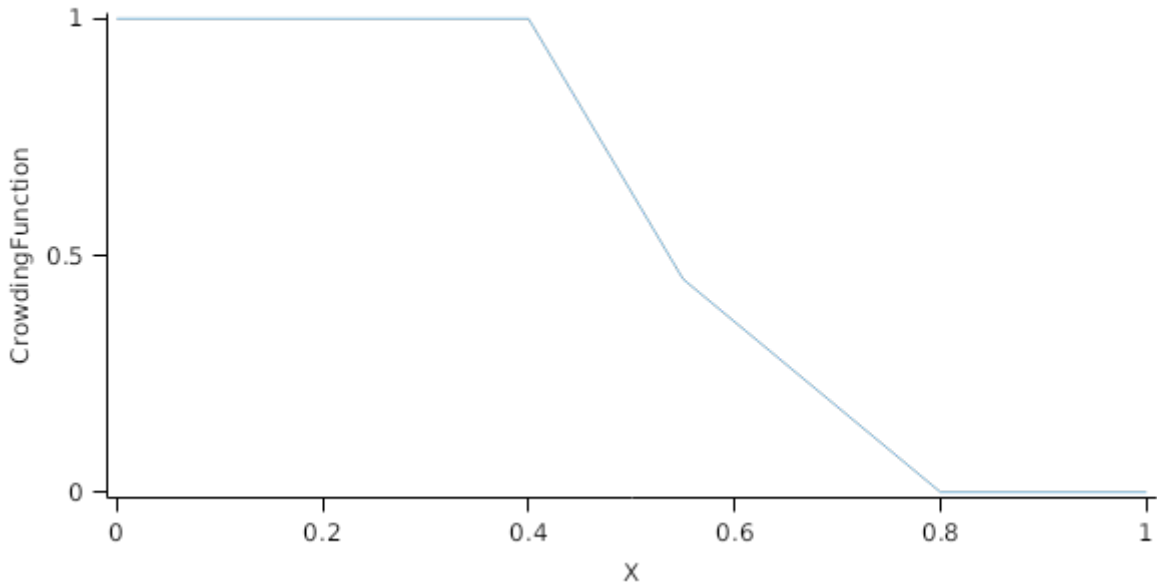
PotentialBranchingRate



$CrowdingFunction$ is calculated using linear interpolation

X	CrowdingFunction
0.0	1.0
0.4	1.0
0.6	0.5
0.8	0.0
1.0	0.0

CrowdingFunction



NewlyDeadBranches is calculated as:

$$\text{NewlyDeadBranches} = (\text{TotalStemPopn} - \text{MainStemPopn}) \times \text{BranchMortality}$$

where *BranchMortality* is given by:

$$\text{BranchMortality} = 0$$

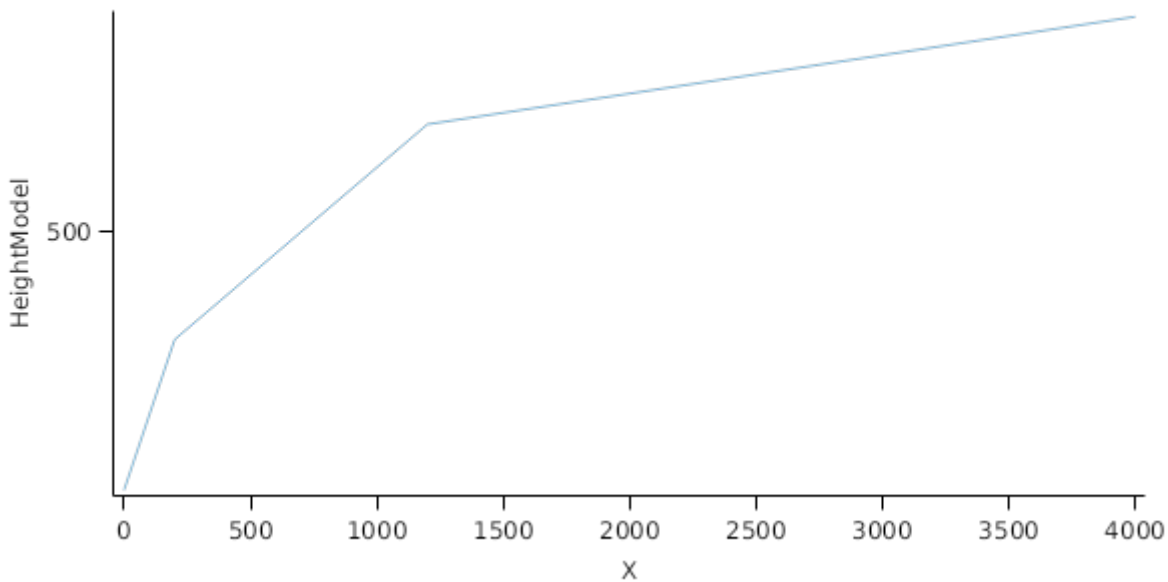
1.8.4 Height

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

HeightModel is calculated using linear interpolation

X	HeightModel
0.0	20.0
200.0	300.0
1200.0	700.0
4000.0	900.0

HeightModel



1.9 Tuber

1.9.1 Tuber

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.

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

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

TuberFilling has a value between TuberInitiation and Maturity calculated as:

$$\text{DailyDemand} = \text{OrganNumber} \times \text{PotentialGrowthIncrement}$$

$$\text{OrganNumber} = \text{TuberNumberPerStem} \times [\text{Structure}].\text{MainStemPopn}$$

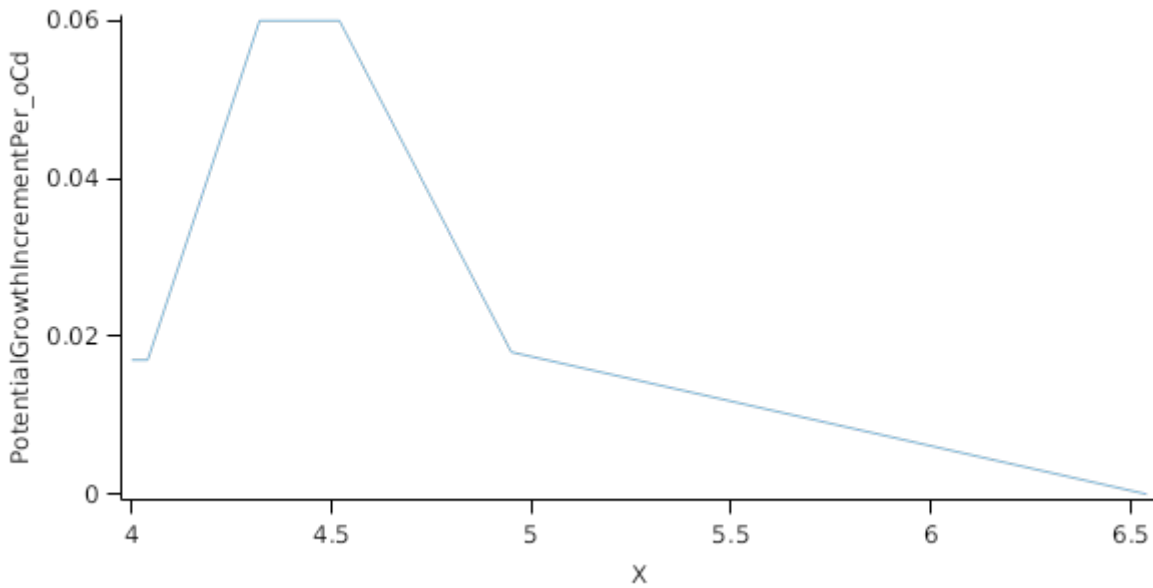
$$\text{TuberNumberPerStem} = 5$$

$$\text{PotentialGrowthIncrement} = \text{PotentialGrowthIncrementPer_oCd} \times \text{ThermalTime}$$

PotentialGrowthIncrementPer_oCd is calculated using linear interpolation

X	PotentialGrowthIncrementPer_oCd
4.0	0.0
4.0	0.0
4.3	0.1
4.5	0.1
5.0	0.0
6.5	0.0

PotentialGrowthIncrementPer_oCd



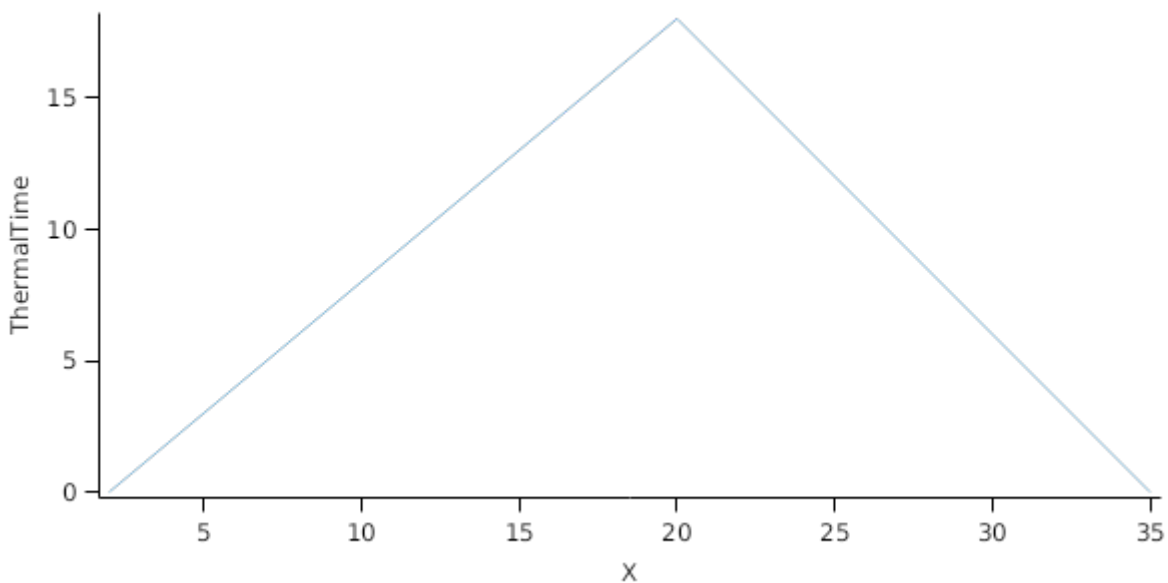
ThermalTime is the average of sub-daily values from a XYPairs.

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 ThermalTime

X	ThermalTime
2.0	0.0
20.0	18.0
35.0	0.0

ThermalTime



StructuralFraction = 1 (g/g)

Metabolic = 0

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

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

$QStructuralPriority = 1$

$QMetabolicPriority = 1$

$QStoragePriority = 1$

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.

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

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

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

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

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

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

$NitrogenDemandSwitch = [Tuber].nitrogenDemandSwitch$

$MaxNconc = [Tuber].maximumNconc$

$QStructuralPriority = 1$

$QMetabolicPriority = 1$

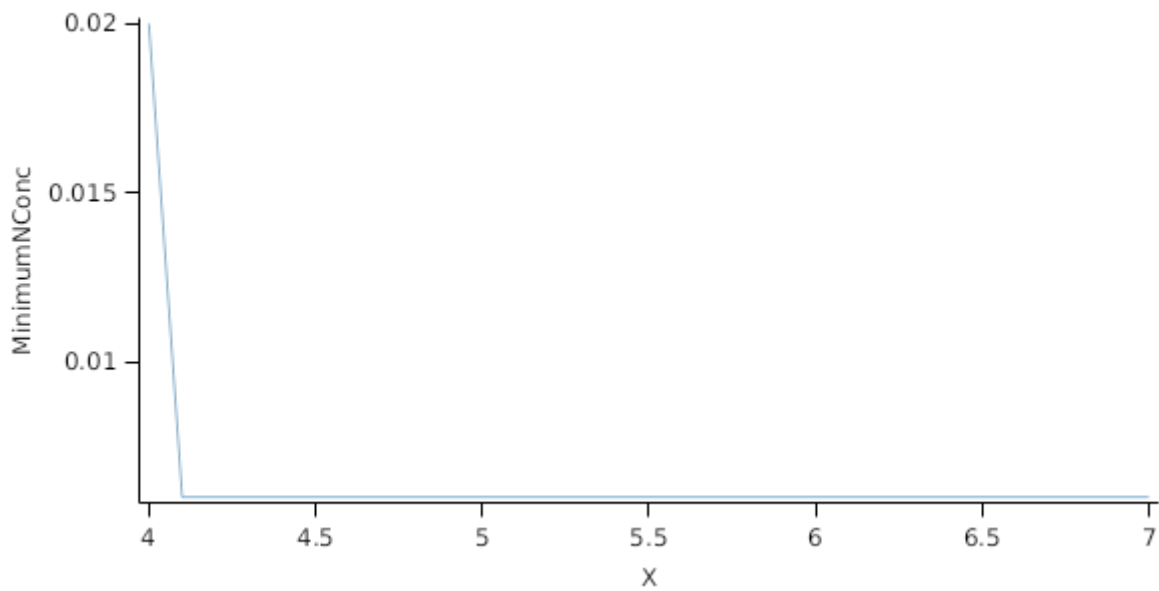
$QStoragePriority = 1$

1.9.4 N Concentration Thresholds

MinimumNConc is calculated using linear interpolation

X	MinimumNConc
4.0	0.0
4.1	0.0
7.0	0.0

MinimumNConc

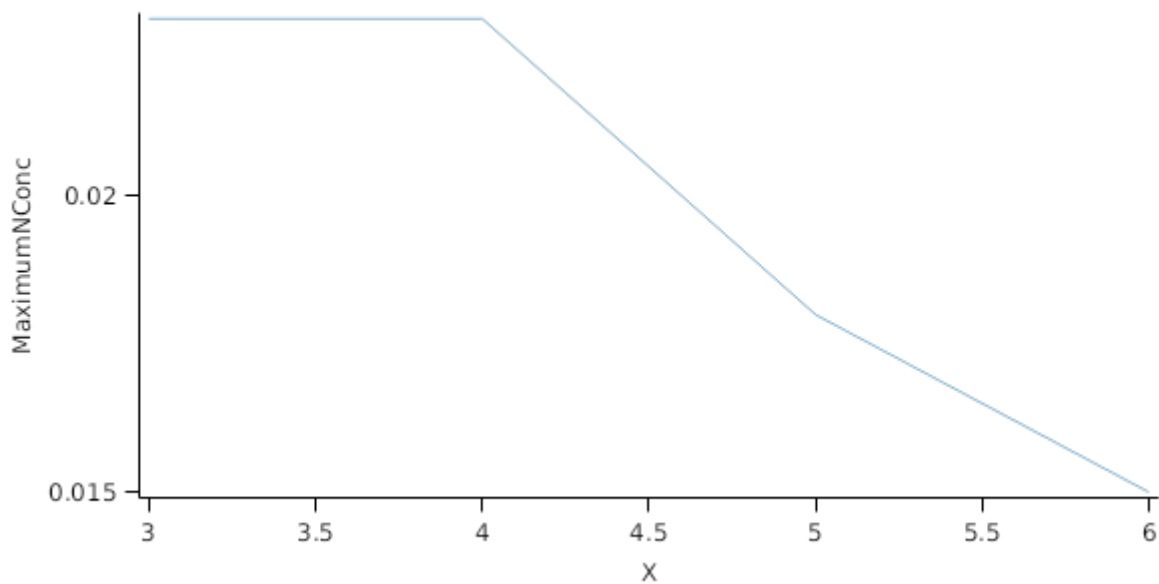


$CriticalNConc = [Tuber].MinimumNConc$

MaximumNConc is calculated using linear interpolation

X	MaximumNConc
3.0	0.0
4.0	0.0
5.0	0.0
6.0	0.0

MaximumNConc



1.9.5 Dry Matter Supply

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

Tuber does not retranslocate non-structural DM.

1.9.6 Nitrogen Supply

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

Tuber does not retranslocate non-structural N.

1.9.7 Senescence and Detachment

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

Tuber 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 Root

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

1.10.1 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.

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

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

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

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

$$DMAllocated_i = \text{TotalDMAllocated} \times RAw_i / \text{TotalRAw}$$

1.10.2 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.

1.10.3 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.

1.10.4 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 (kNO3 or kNH4), the concentration of N form (ppm), and a soil moisture factor (NUptakeSWFactor) which typically decreases as the soil dries. $NO_3 \text{ uptake} = NO_{3_i} \times kNO_3 \times NO_{3_{ppm, i}} \times NUptakeSWFactor$ $NH_4 \text{ uptake} = NH_{4_i} \times kNH_4 \times NH_{4_{ppm, i}} \times NUptakeSWFactor$ As can be seen from the above equations, the values of kNO3 and kNH4 equate to the potential fraction of each mineral N pool which can be taken up per day for wet soil when that pool has a concentration of 1 ppm. Nitrogen uptake demand is limited to the maximum daily potential uptake (MaxDailyNUptake) and the plant's N demand. The former provides a means to constrain N uptake to a maximum value observed in the field for the crop as a whole. The demand for soil N is then passed to the soil arbitrator which determines how much of the N uptake demand each plant instance will be allowed to take up.

1.10.5 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. $SW \text{ uptake} = (SW_i - LL_i) \times KL_i \times KLModifier$

1.10.6 Constants

MaximumNConc = 0.01

MaxDailyNUptake = 6

MinimumNConc = 0.01

MaximumRootDepth = 1000000

KNO3 = 0.03

KNH4 = 0.03

SpecificRootLength = 40 (m/g)

KL is not modified by any plant or soil attribute and so a constant value of 1 is used for this KL modifier.

KLModifier = 1

DMConversionEfficiency = 1

MaintenanceRespirationFunction = 1

RemobilisationCost = 0

CarbonConcentration = 0.4

RootDepthStressFactor = 1

1.10.7 RootShape

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

1.10.8 NitrogenDemandSwitch

NitrogenDemandSwitch has a value between Planting and FinalLeaf calculated as:

Constant = 1

1.10.9 RootFrontVelocity

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

PhaseLookupValue has a value between Peeping and FinalLeaf calculated as:

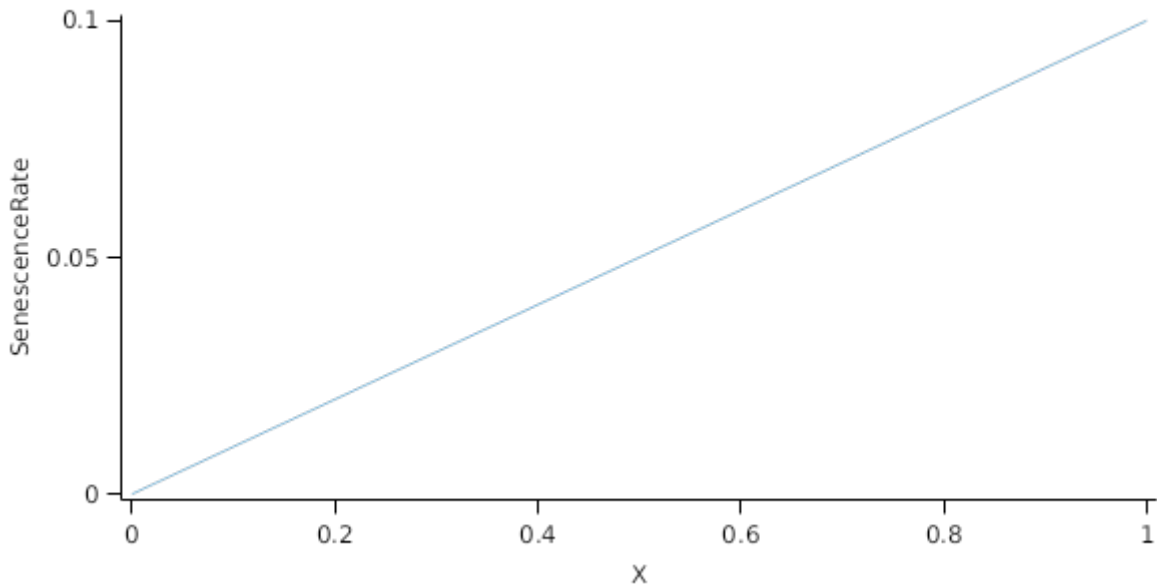
Constant = 10

1.10.10 SenescenceRate

SenescenceRate is calculated using linear interpolation

X	SenescenceRate
0.0	0.0
1.0	0.1

SenescenceRate

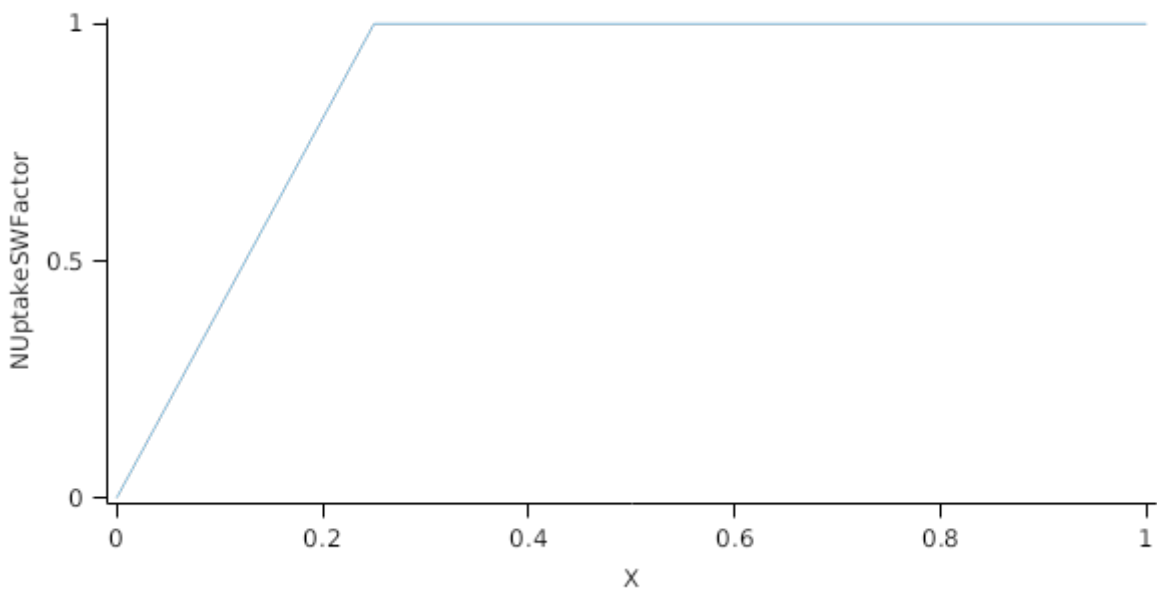


1.10.11 NUptakeSWFactor

NUptakeSWFactor is calculated using linear interpolation

X	NUptakeSWFactor
0.0	0.0
0.2	1.0
1.0	1.0

NUptakeSWFactor



1.10.12 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
Thin	0	0	5	0

1.10.13 DM Demands

1.10.13.1 DM Demands

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

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

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

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

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

PhaseLookupValue has a value between Emergence and TuberInitiation calculated as:

$$\text{Constant} = 0.3$$

PhaseLookupValue1 has a value between TuberInitiation and FinalLeaf calculated as:

$$\text{Constant} = 0.05$$

PhaseLookupValue2 has a value between FinalLeaf and FullSenescence calculated as:

$$\text{Constant} = 0$$

$$\text{StructuralFraction} = 1$$

$$\text{Metabolic} = 0$$

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

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

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

1.10.14 N Demands

1.10.14.1 N Demands

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

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

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

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

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

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

The demand for storage N is further reduced by a factor specified by the `[\text{Root}].NitrogenDemandSwitch`.

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

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

$$\text{QStructuralPriority} = 1$$

$$\text{QMetabolicPriority} = 1$$

$$\text{QStoragePriority} = 1$$

1.10.15 CriticalNConc

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

1.10.16 InitialWt

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

$$\text{Structural} = 5 \text{ (g/plant)}$$

$$\text{Metabolic} = 0$$

$$\text{Storage} = 0$$

1.11 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 m² 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.

1.11.1 Dry Matter Fixation

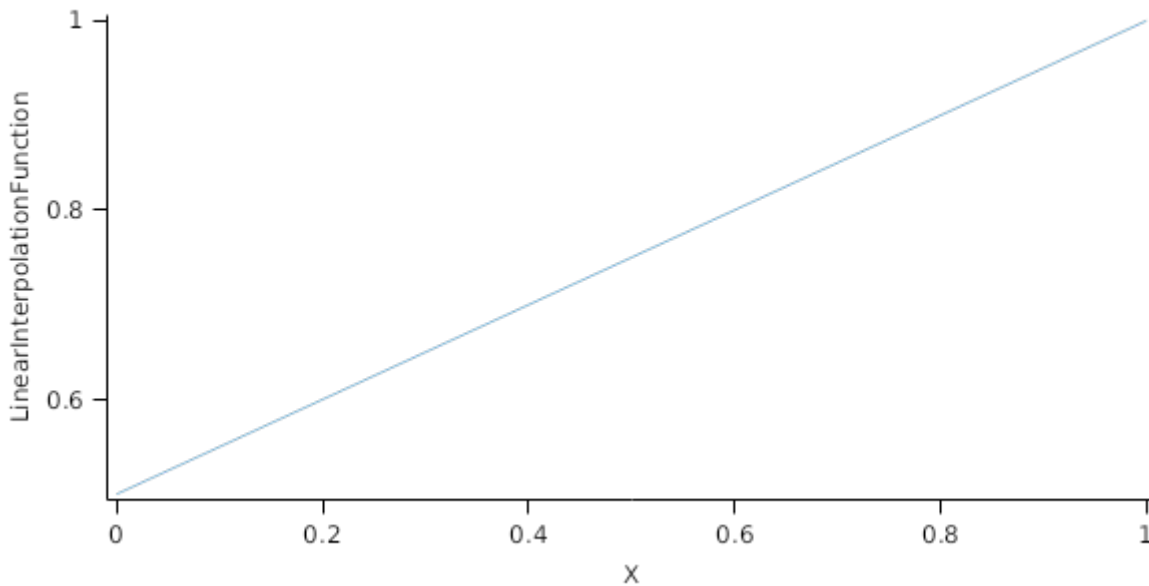
The most important DM supply from leaf is the photosynthetic fixation supply. Radiation interception is calculated from LAI using an extinction coefficient of:

$$\text{ExtinctionCoeff} = 0.8 \times \text{LinearInterpolationFunction}$$

LinearInterpolationFunction is calculated using linear interpolation

X	LinearInterpolationFunction
0.0	0.5
1.0	1.0

Linear Interpolation Function



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 CO₂ 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 = 1.2

This model calculates the CO₂ 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

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

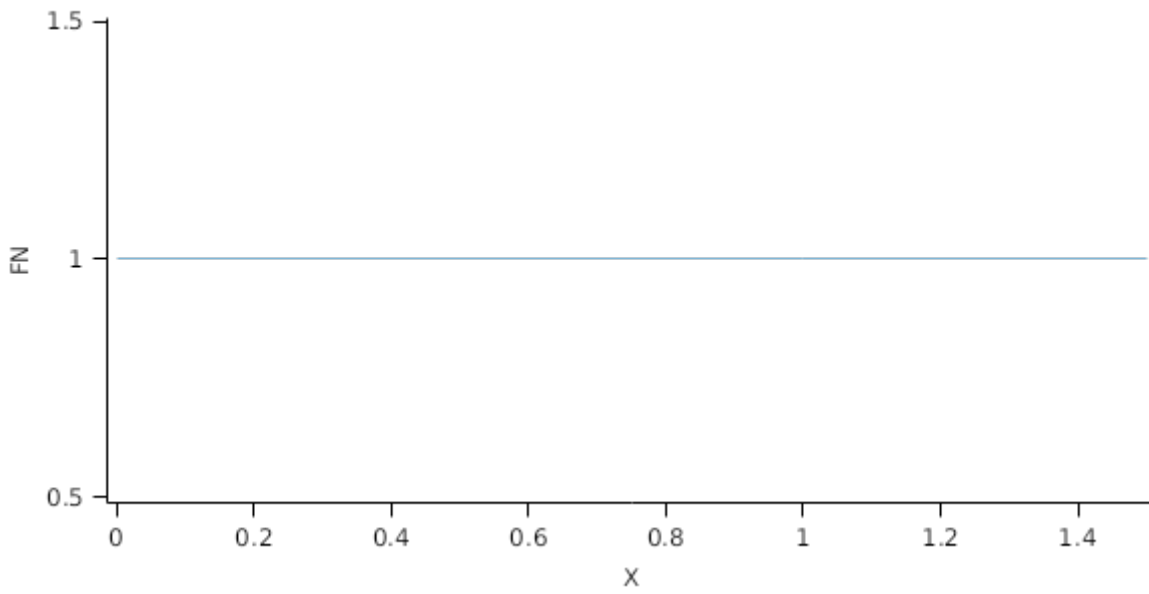
For C4 plants,

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

FN is calculated using linear interpolation

X	FN
0.0	1.0
1.0	1.0
1.5	1.0

FN

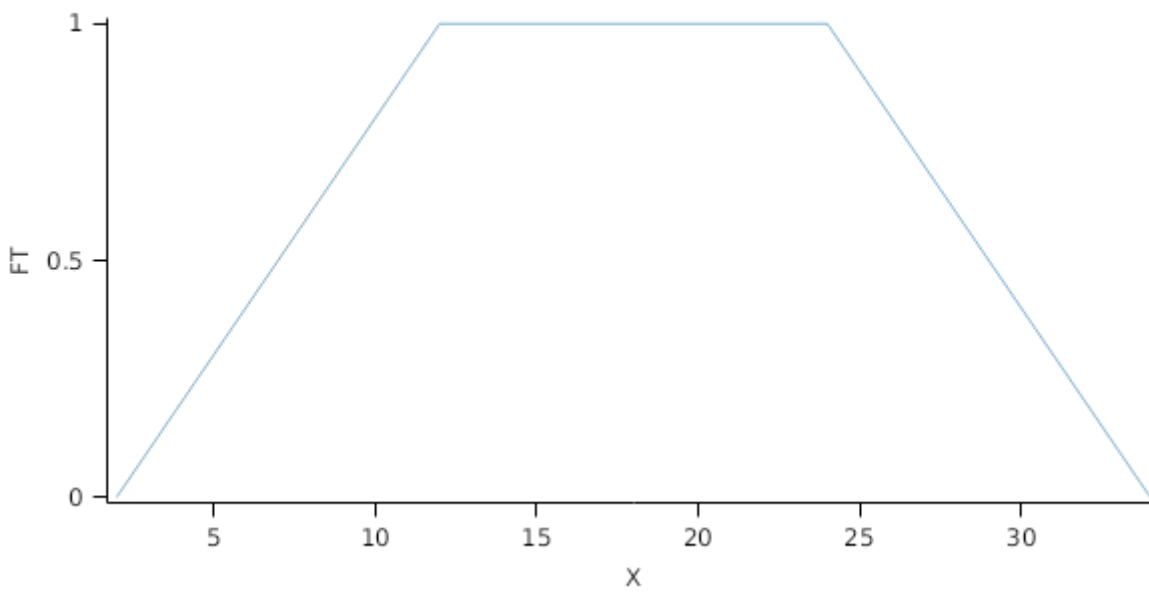


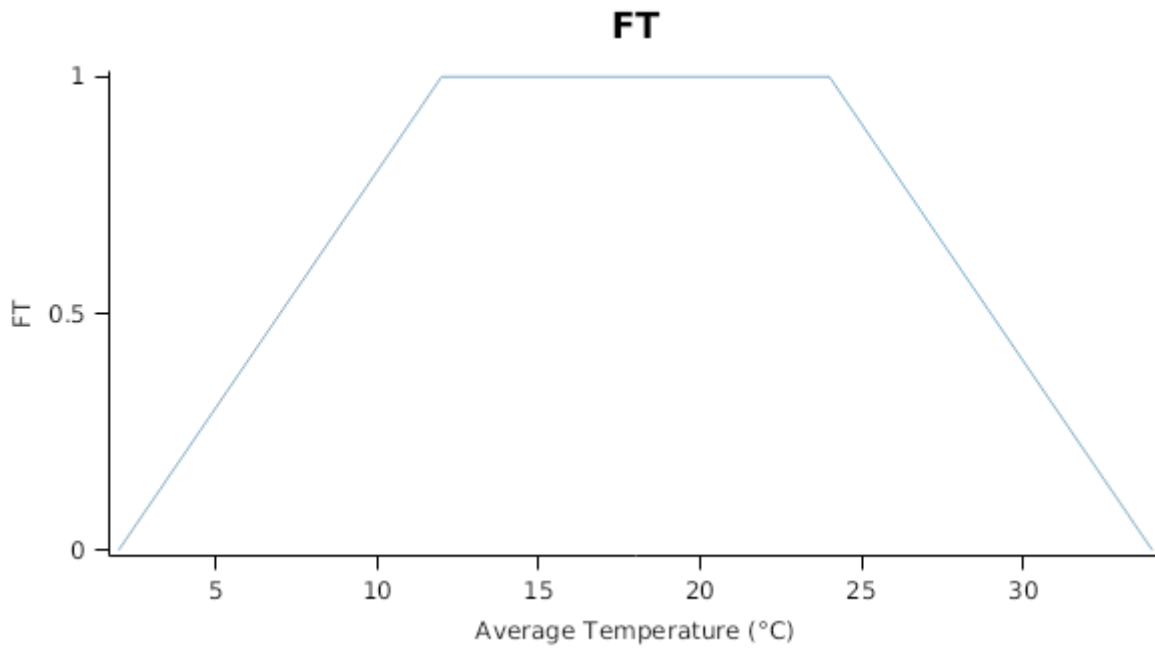
FT is calculated as a function of daily min and max temperatures, these are weighted toward max temperature according to the specified MaximumTemperatureWeighting factor. A value equal to 1.0 means it will use max temperature, a value of 0.5 means average temperature.

MaximumTemperatureWeighting = 0.5

X	FT
2.0	0.0
12.0	1.0
24.0	1.0
34.0	0.0

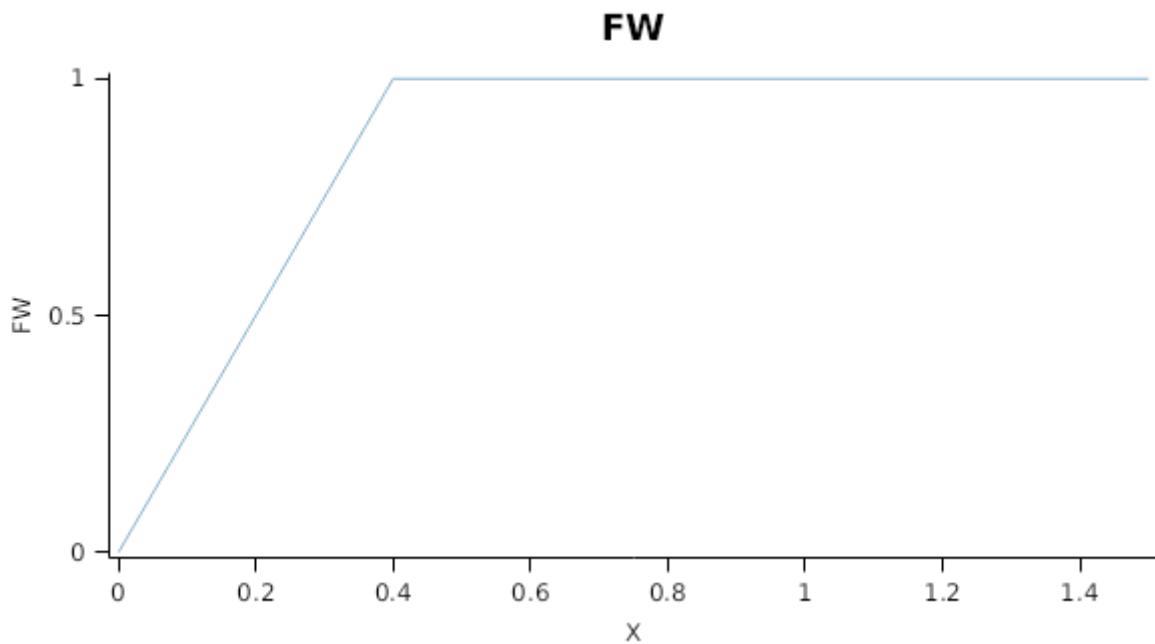
FT





FW is calculated using linear interpolation

X	FW
0.0	0.0
0.4	1.0
1.5	1.0



$$RadnInt = [Leaf].RadiationIntercepted$$

This model assumes no impact of VPD on stomatal conductance or photosynthesis.

$$FVPD = 1 \text{ (0-1)}$$

1.11.2 Constants

$$StructuralFraction = 0.5$$

$$DMConversionEfficiency = 1$$

RemobilisationCost = 0

CarbonConcentration = 0.4

WidthFunction = 0

1.11.3 CO2internal

$CO2_{internal} = (163 - [IWeather].MeanT) / (5 - 0.1 \times [IWeather].MeanT)$

1.11.4 StomatalConductanceCO2Modifier

$StomatalConductanceCO2Modifier = [Leaf].Photosynthesis.FCO2 / RelativeCO2Gradient$

$RelativeCO2Gradient = ([IWeather].CO2 - [Leaf].CO2_{internal}) / (350 - [Leaf].CO2_{internal})$

1.11.5 ThermalTime

$ThermalTime = [Phenology].ThermalTime$

1.11.6 InitialLeaf1

Area = 900

1.11.7 InitialLeaf2

Area = 600

1.11.8 InitialLeaf3

Area = 300

1.11.9 CohortParameters

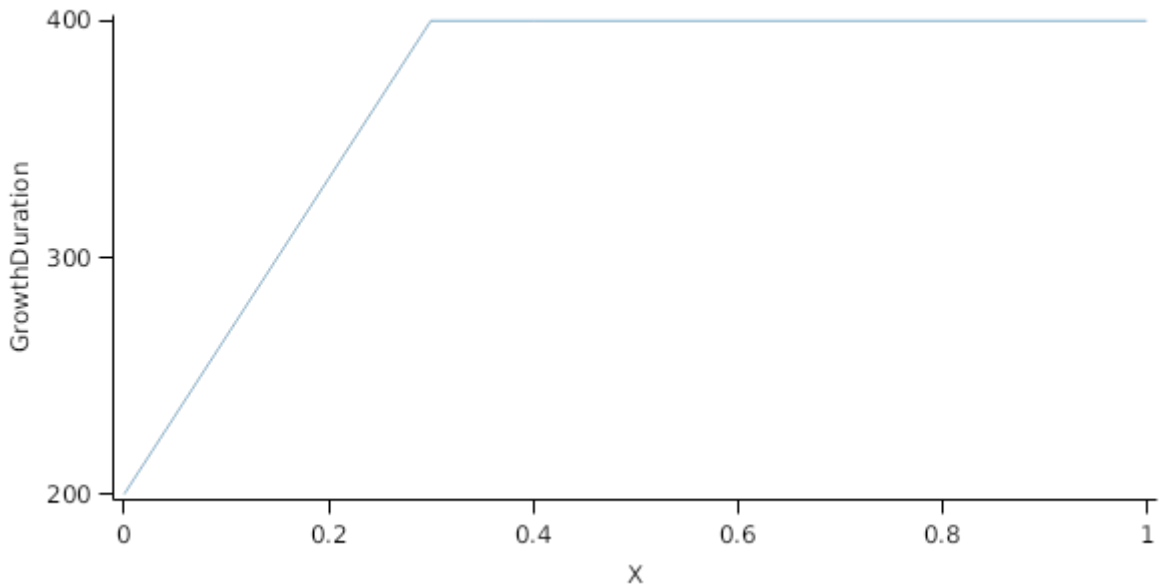
1.11.9.1 Potential Leaf Area index

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	200.0
0.3	400.0
0.4	400.0
1.0	400.0

GrowthDuration

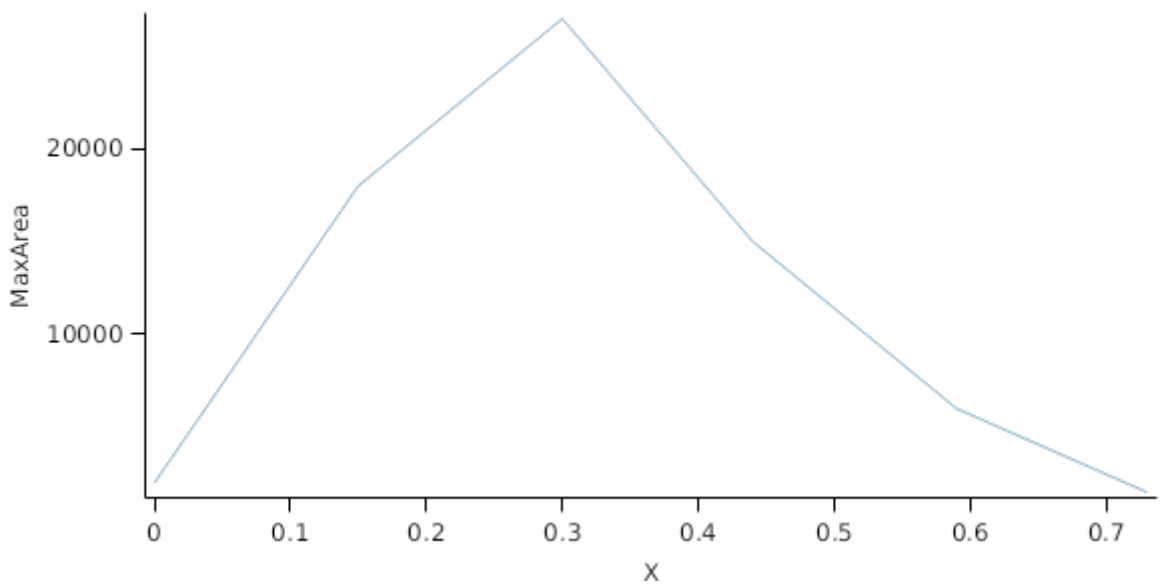


If no stress occurs the leaves will reach a Maximum area (*MaxArea*) at the end of the *GrowthDuration*. The *MaxArea* is defined by:

MaxArea is calculated using linear interpolation

X	MaxArea
0.0	2000.0
0.1	18000.0
0.3	27000.0
0.4	15000.0
0.6	6000.0
0.7	1500.0

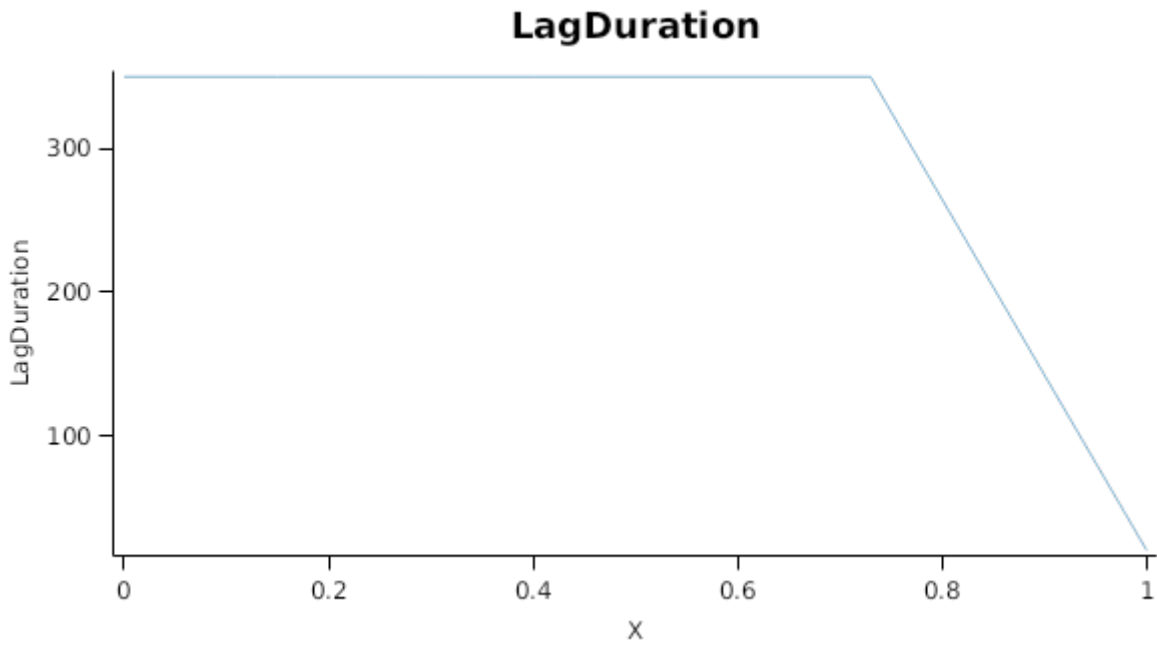
MaxArea



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

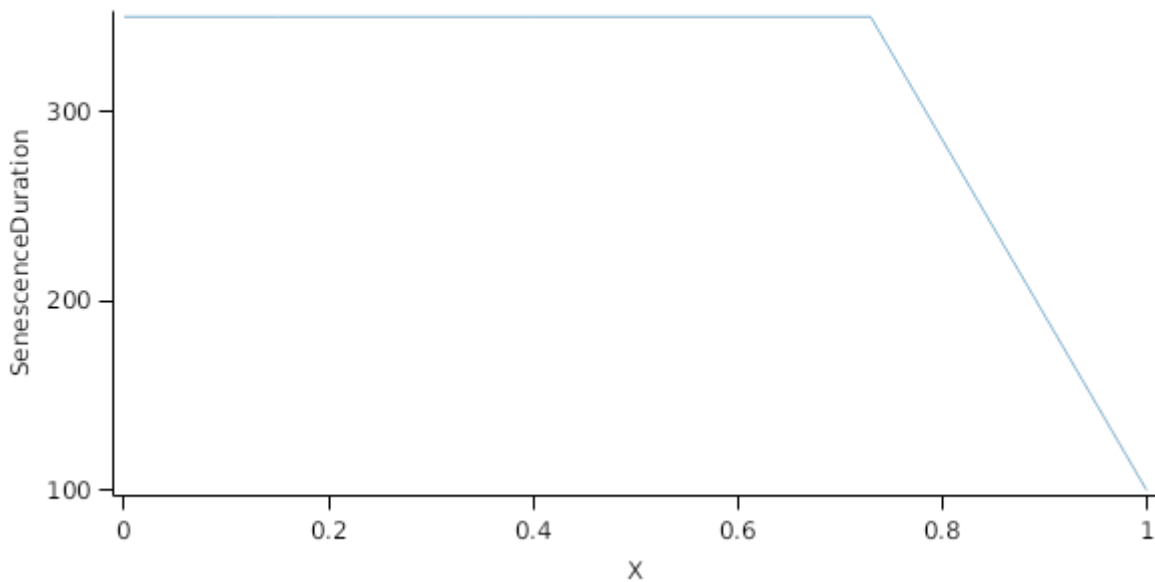
X	LagDuration
0.0	350.0
0.1	350.0
0.3	350.0
0.4	350.0
0.7	350.0
1.0	20.0



SenescenceDuration is calculated using linear interpolation

X	SenescenceDuration
0.0	350.0
0.1	350.0
0.3	350.0
0.4	350.0
0.7	350.0
1.0	100.0

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:

The model assumes no impact of shading on leaf senescence.

ShadeInducedSenescenceRate = 0

1.11.9.2 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:

$CellDivisionStress = 1 - (0-1)$

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 $(Leaf.NConc - MinimumNConc) / (CriticalNConc - MinimumNConc)$.

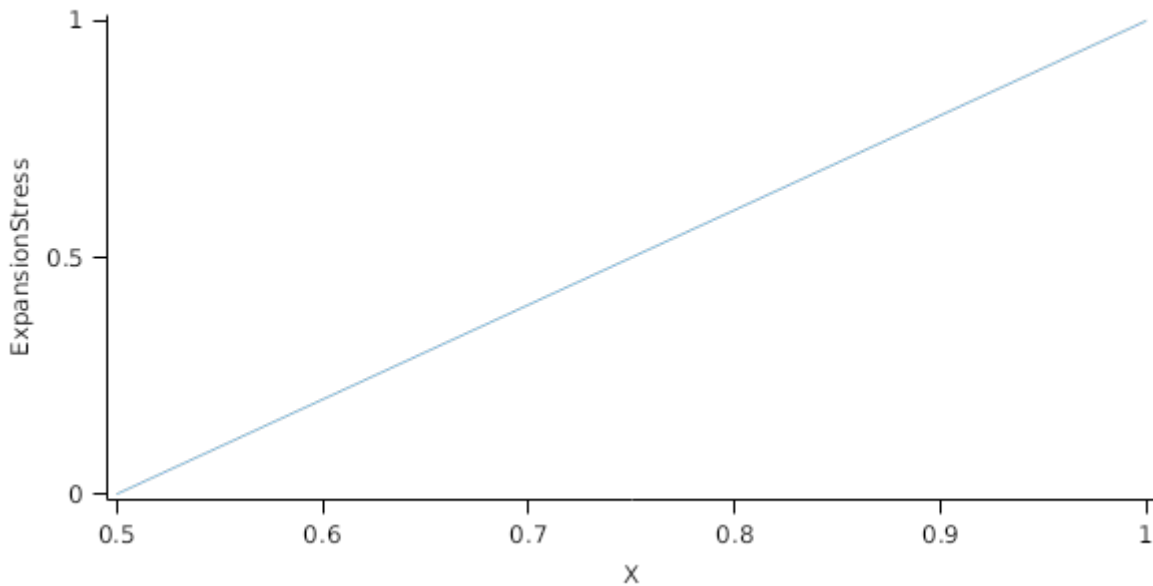
Leaf.FW quantifies water stress and is calculated as $Leaf.Transpiration / Leaf.WaterDemand$, where *Leaf.Transpiration* is the minimum of *Leaf.WaterDemand* and *Root.WaterUptake*

Stress during the $<i>GrowthDuration^*$ of the cohort reduces the size increase of the cohort by multiplying the potential increase by a *ExpansionStress* factor:

ExpansionStress is calculated using linear interpolation

X	ExpansionStress
0.5	0.0
1.0	1.0

ExpansionStress



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.

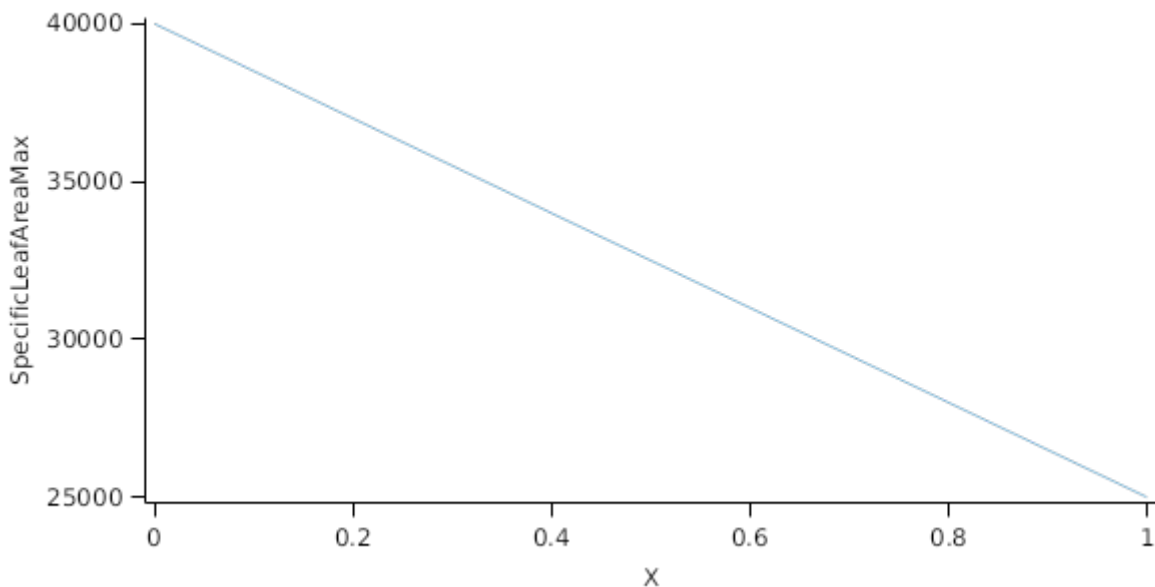
1.11.9.3 Dry matter Demand

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
0.0	40000.0
1.0	25000.0

SpecificLeafAreaMax



SpecificLeafAreaMin = 25000

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

StructuralFraction = 0.5

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

1.11.9.4 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 = 0.03

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

CriticalNConc = 0.05

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 = 0.07

1.11.9.5 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 = 0

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

1.11.9.6 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 :

$N_{RetranslocationFactor} = 0.01$

1.11.9.7 Constants

$SenescingLeafRelativeSize = 1$ (0-1)

$DetachmentLagDuration = 1000000$

$DetachmentDuration = 1000000$

$SpecificLeafAreaMin = 25000$

$StructuralFraction = 0.5$

$MaximumNConc = 0.07$

$MinimumNConc = 0.03$

$InitialNConc = 0$

$N_{ReallocationFactor} = 1$

$N_{RetranslocationFactor} = 0.01$

$CriticalNConc = 0.05$

$DM_{RetranslocationFactor} = 0.5$

The model assumes no impact of shading on leaf senescence.

$ShadeInducedSenescenceRate = 0$

$StorageFraction = 0.2$

$DM_{ReallocationFactor} = 0$

$LeafSizeShapeParameter = 0.01$

$MaintenanceRespirationFunction = 0$

$SenescenceAcceleration = 1$

$CellDivisionStress = 1$ (0-1)

$LagAcceleration = 1$

$RemobilisationCost = 0$

1.11.10 FRGRFunction

$FRGRFunction = \text{Min}(RUE_{FT}, Others)$

Where:

$RUE_{FT} = [Leaf].Photosynthesis.FT$

$Others = \text{Min}(RUE_{FN}, RUE_{FVPD})$

Where:

$RUE_{FN} = [Leaf].Photosynthesis.FN$

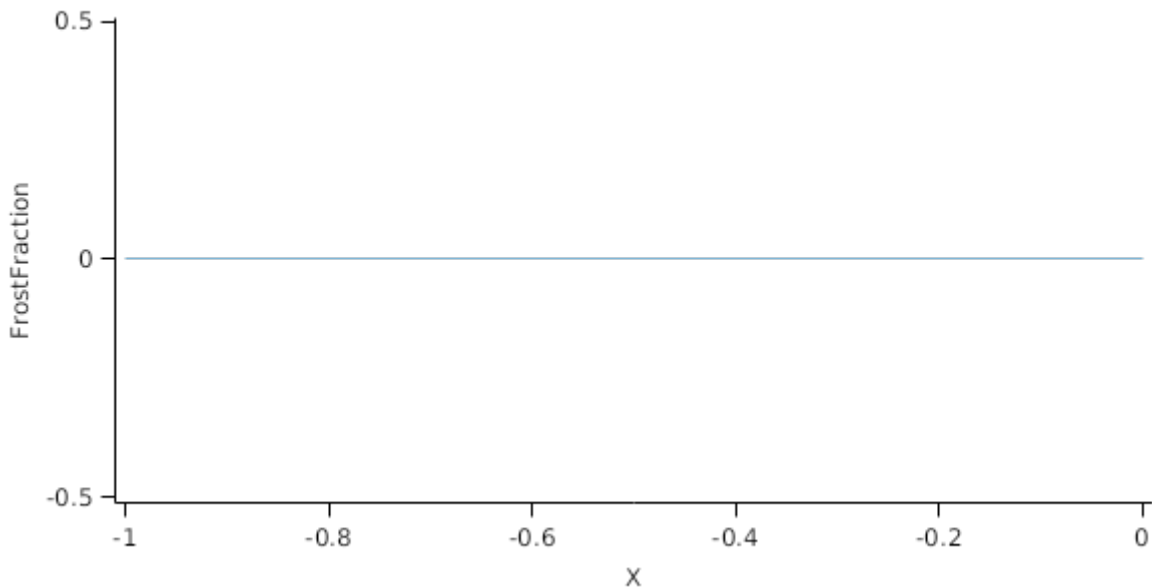
$RUE_{FVPD} = [Leaf].Photosynthesis.FVPD$

1.11.11 FrostFraction

FrostFraction is calculated using linear interpolation

X	FrostFraction
-1.0	0.0
0.0	0.0

FrostFraction



1.11.12 Total

1.11.12.1 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

1.11.13 BiomassRemovalDefaults

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

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	0	0	20	0
Cut	0	0	30	0
Prune	0	0	10	0
Graze	0	0	15	0
Thin	0	0	5	0

1.11.14 DepthFunction

$DepthFunction = [Leaf].Height$

1.11.15 DMDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

1.11.16 NDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

1.12 Stem

1.12.1 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.

1.12.2 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.

1.12.2.1 DMDemands

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

$Structural = DMDemandFunction \times StructuralFraction$

$DMDemandFunction = [Structure].DeltaTipNumber \times InterNodeWt \times [Structure].TotalStemPopn$

InterNodeWt = 0.2

StructuralFraction = 1 (g/g)

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

1.12.3 Nitrogen Demand

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

1.12.3.1 NDemands

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

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

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

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

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

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

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

$NitrogenDemandSwitch = [Stem].nitrogenDemandSwitch$

$MaxNconc = [Stem].maximumNconc$

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

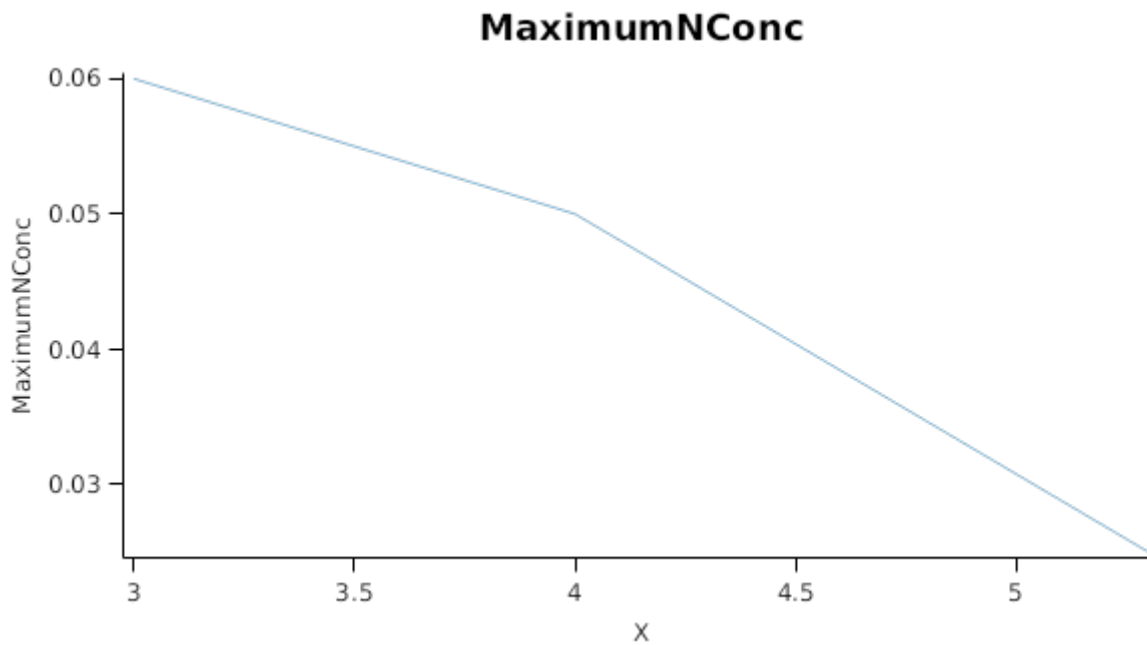
1.12.4 N Concentration Thresholds

MinimumNConc = 0.006

$CriticalNConc = [Stem].MinimumNConc$

MaximumNConc is calculated using linear interpolation

X	MaximumNConc
3.0	0.1
4.0	0.1
5.3	0.0



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

NitrogenDemandSwitch has a value between Planting and FinalLeaf calculated as:

Constant = 1

1.12.5 Dry Matter Supply

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

Stem does not retranslocate non-structural DM.

1.12.6 Nitrogen Supply

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

Stem can retranslocate up to 5% of non-structural N each day if required by the plant arbitrator to meet N demands.

1.12.7 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	0	0	20	0
Cut	0	0	30	0
Prune	0	0	10	0
Graze	0	0	15	0
Thin	0	0	5	0

1.13 AboveGround

1.13.1 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

1.14 BelowGround

1.14.1 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
- * Tuber

1.15 Total

1.15.1 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
- * Root
- * Tuber

1.16 TotalLive

1.16.1 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
- * Root
- * Tuber

1.17 TotalDead

1.17.1 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
- * Root
- * Tuber

1.18 CultivarFolder

1.18.1 Coliban

Coliban overrides the following properties:

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.10,0.19

[Phenology].Vegetative.Target.FixedValue = 200

1.18.2 Horizon

Horizon overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 28

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.12,0.16

1.18.3 IlamHardy

IlamHardy overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 26

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.18

[Phenology].Vegetative.Target.FixedValue = 200

1.18.4 JerseyBenny

JerseyBenny overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 19

[Structure].BranchingRate.PotentialBranchingRate.XYPairs.X = 0,6,7,10,11

[Structure].BranchingRate.PotentialBranchingRate.XYPairs.Y = 0,0,0.5,0.5,0

[Tuber].DMDemands.Structural.DMDemandFunction.TuberFilling.DailyDemand.OrganNumber.TuberNumberPerStem.FixedValue = 3

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.2

1.18.5 Karaka

Karaka overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 29

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.2

1.18.6 Nadine

Nadine overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 29

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.11,0.15

1.18.7 Nooksac

Nooksac overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 36

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.21

1.18.8 RedRascal

RedRascal overrides the following properties:

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.2

1.18.9 Rua

Rua overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 29

[Tuber].LiveFWt.DryMatterContent.XYPairs.Y = 0.13,0.21

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.X = 0,0.15,0.74,0.75,1

[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.Y = 5,40,40,60,60

[Structure].BranchingRate.PotentialBranchingRate.XYPairs.X = 0,6,7,20,21

[Structure].BranchingRate.PotentialBranchingRate.XYPairs.Y = 0,0,0.5,0.5,0

1.18.10 RussetBurbank

RussetBurbank overrides the following properties:

[Structure].FinalLeafNumber.FixedValue = 34

1.19 MortalityRate

MortalityRate = 0

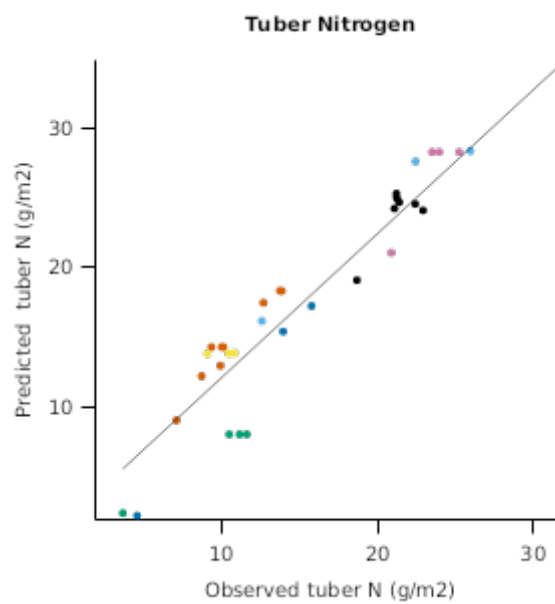
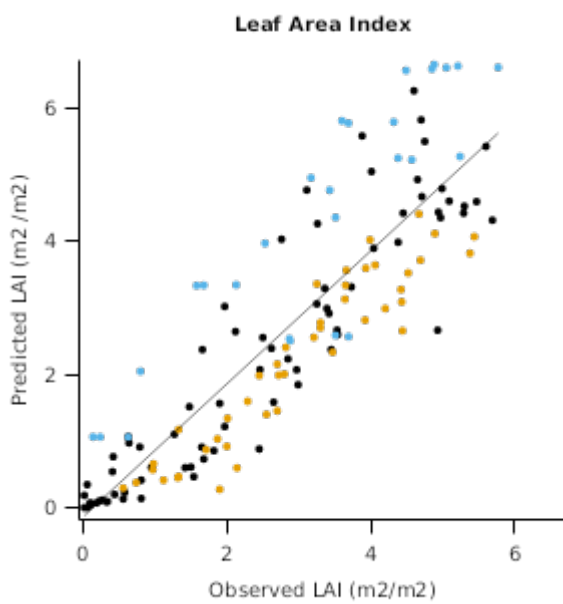
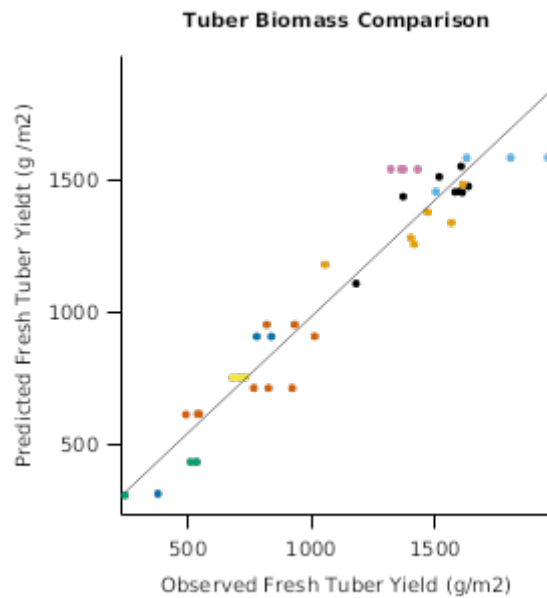
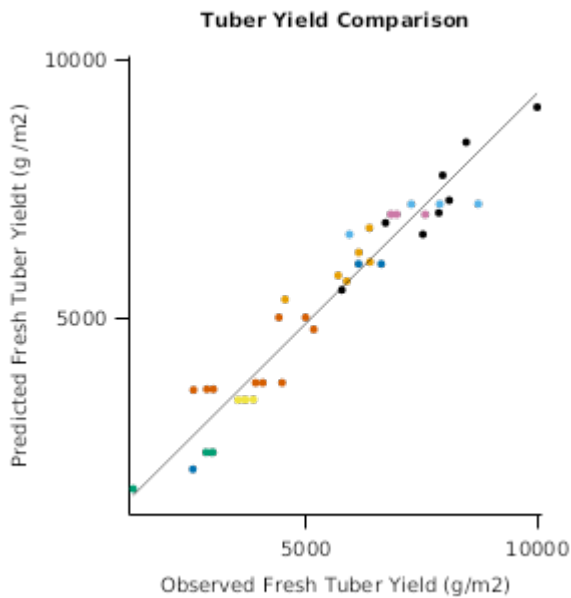
2 Validation

A test dataset has been developed to test the APSIM Potato model for a range of environmental (soil and climate) conditions, management options (sowing dates, populations, nitrogen rates) and genetic backgrounds (different regions, cultivar types). These tests have been grouped into various geographical regions to allow users to evaluate the suitability of the model for their particular region of interest. Graphs of model performance are provided for yield, biomass production, canopy development and nitrogen uptake.



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, sowing date).



2.2 New Zealand

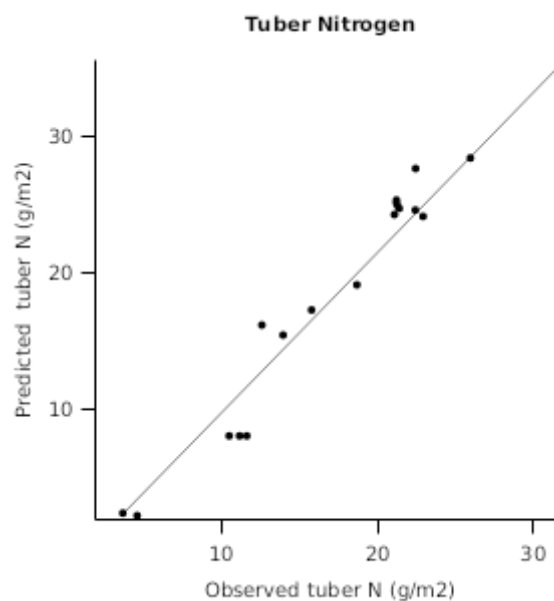
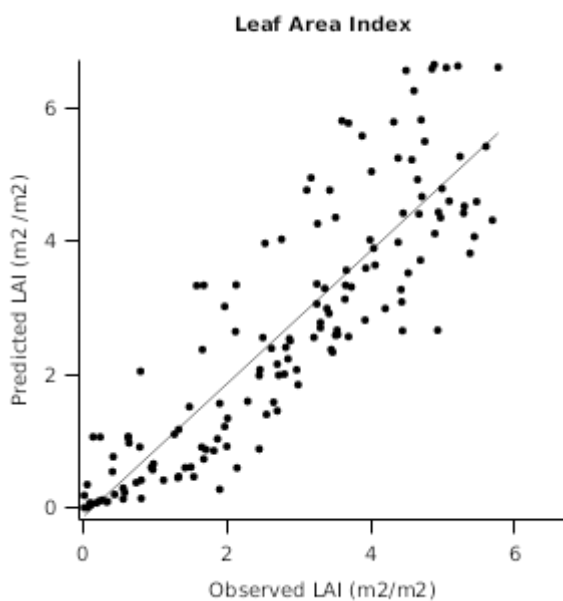
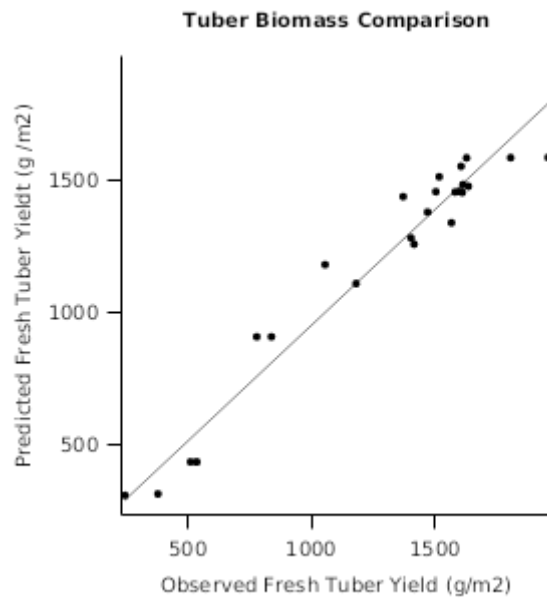
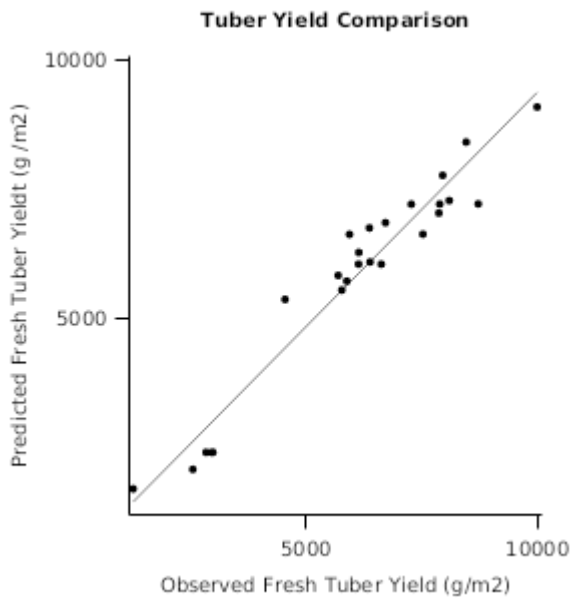
Model development

A range of experimental data has been collected by Plant and Food Research over a long period and was available for fitting the growth and development parameters for the potato model.

- * **ABlock** experiment was used for fitting leaf appearance and size parameters and biomass production and partitioning
- * **Baxters** experiment was used for fitting branching parameters
- * **FUE** experiment was used for fitting nitrogen responses
- * **FUE_NI** experiment was used for fitting nitrogen response in winter grown potato
- * **FSP** experiment was used for fitting growth and development parameters in cooler situations

List of experiments.

Experiment Name	Design (Number of Treatments)
ABlock	Cultivar (8)
Baxters	PlantingDate x PlantDensity (6)
FUE	Nitrogen (4)
FSP	Nitrogen (4)
FUE_NI	Nitrogen (3)



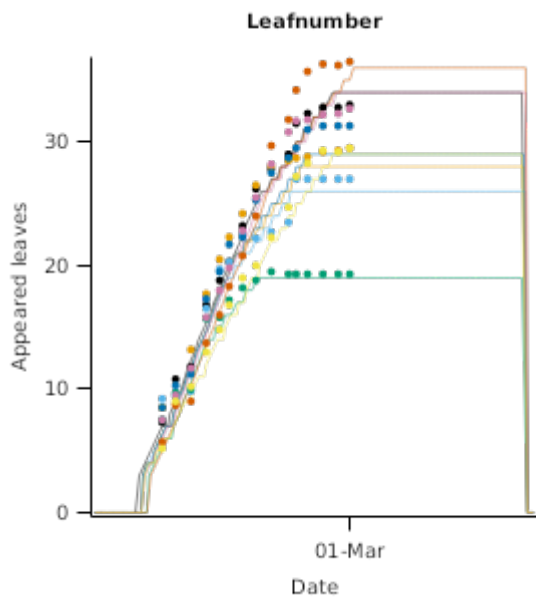
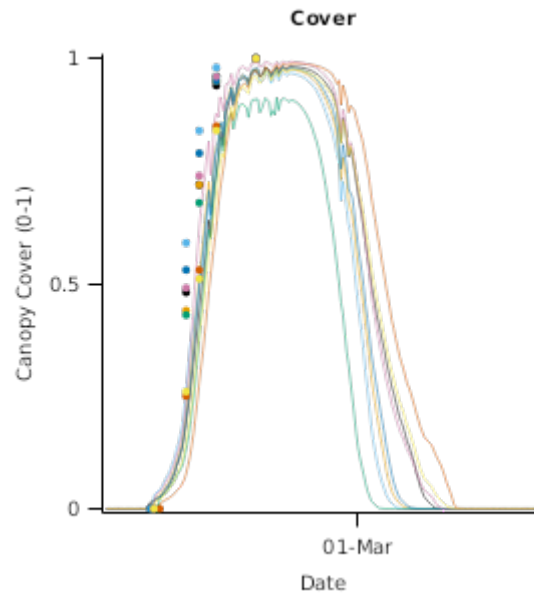
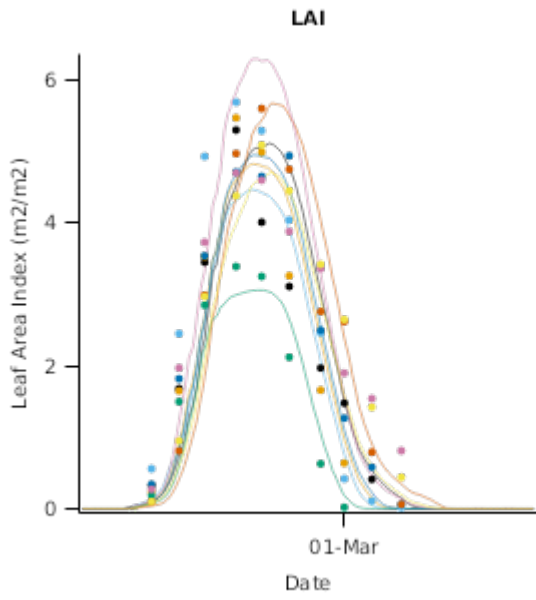
2.2.1 ABlock

This trial was run in A Block of the Plant and Food Research farm on a templeton silt loam, which is a deep, stone free soil of high plant available water capacity (160 mm/m). 8 cultivars of potato were planted representing a range of commercially important varieties in New Zealand:

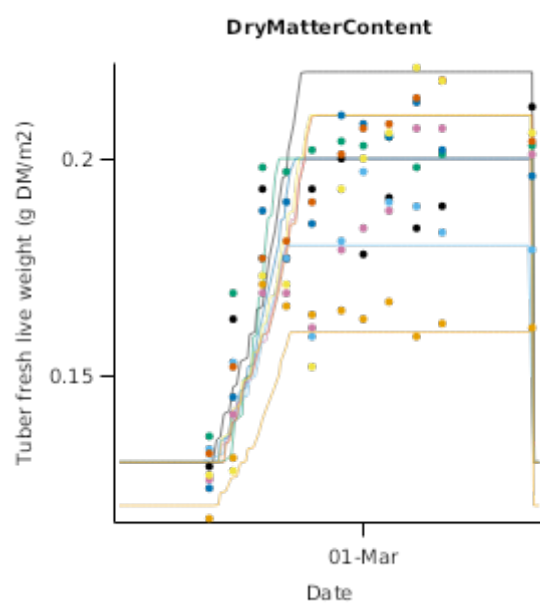
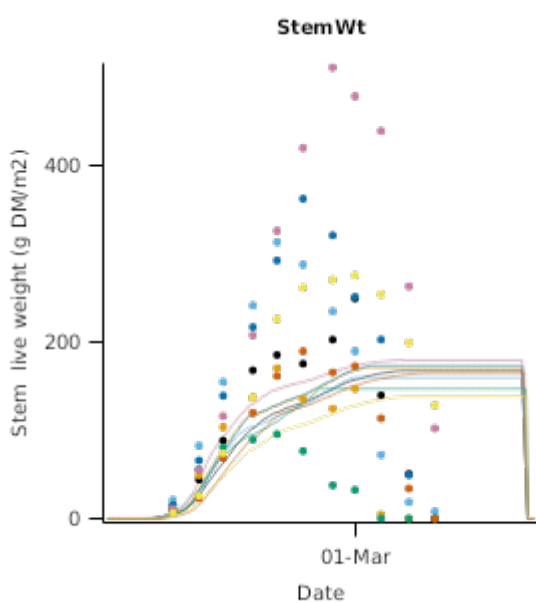
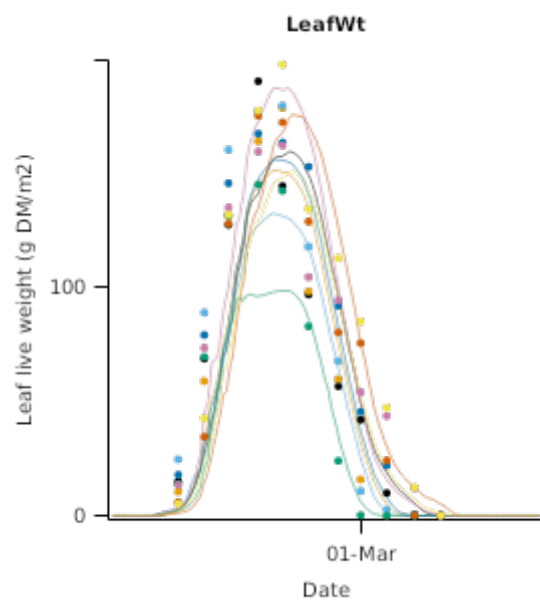
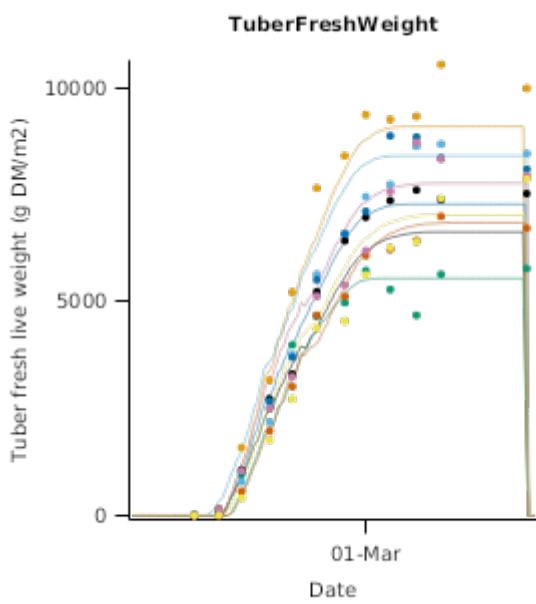
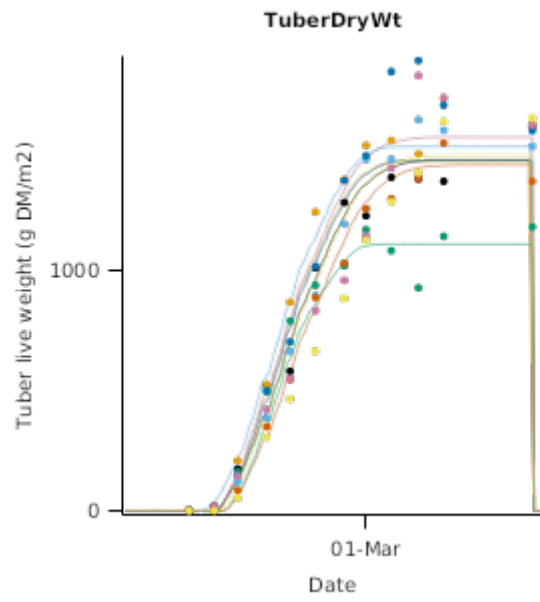
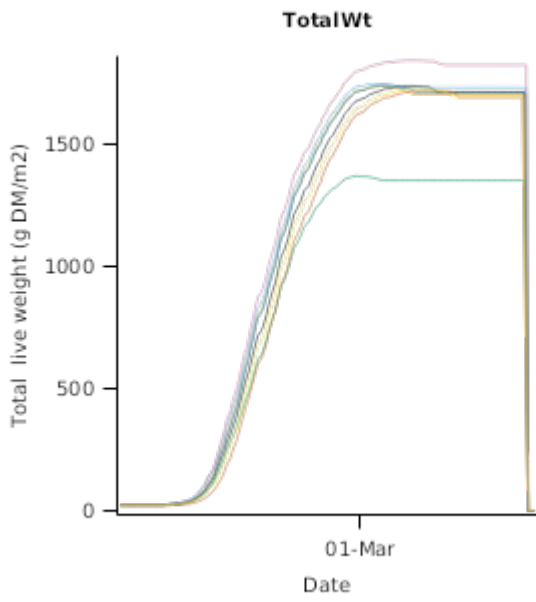
- * Russet Burbank is a main crop variety grown for processing into fries
- * Jersey is a short variety grown for early table potato production
- * Red Rascal is a main crop table potato variety
- * Horizon is a main crop table variety
- * Ilam hardy is commonly grown as an early table variety
- * Karaka is a crisping variety
- * Nooksac is a main crop variety grown for processing into fries
- * Rua is a main crop table potato variety

All potatoes were planted on 20/10/2003 and irrigated, fertiliser and pesticide were applied to ensure the crop remained unstressed and healthy. Each variety was intensively sampled throughout the growth period to provide data for developing cultivar parameters for different varieties.

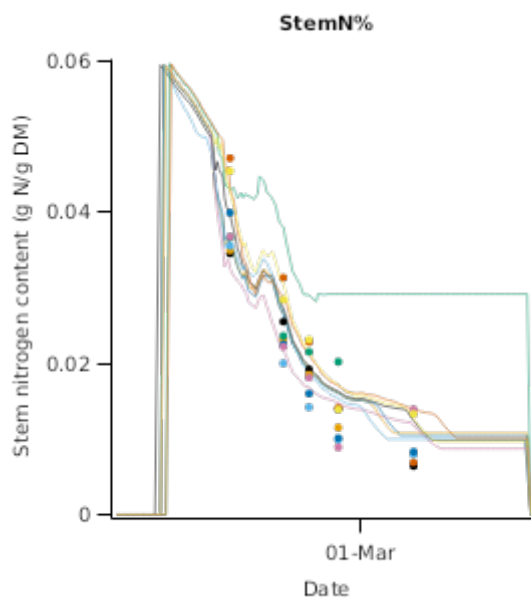
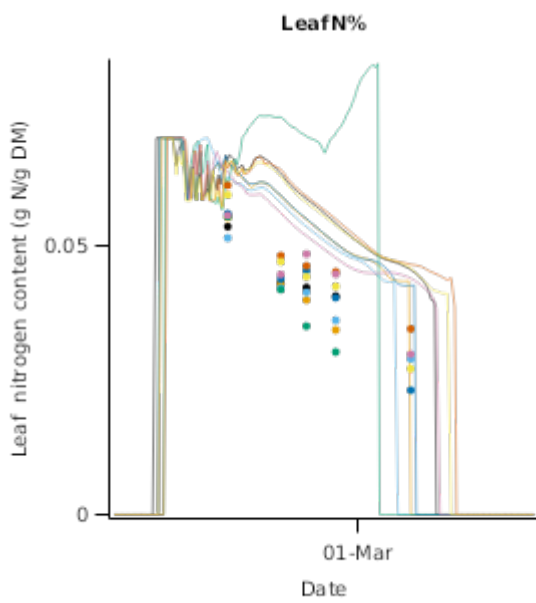
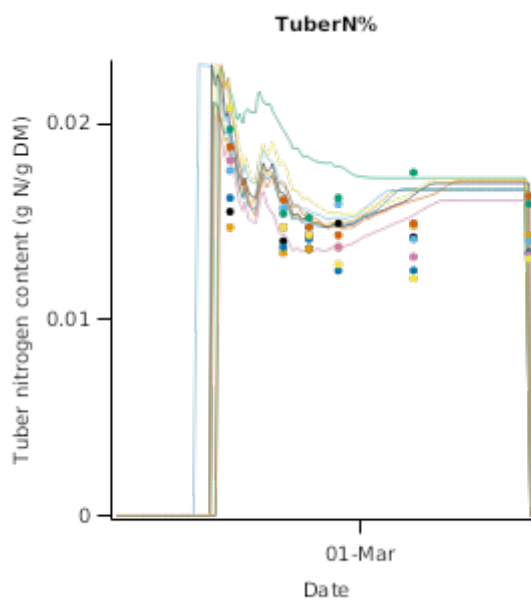
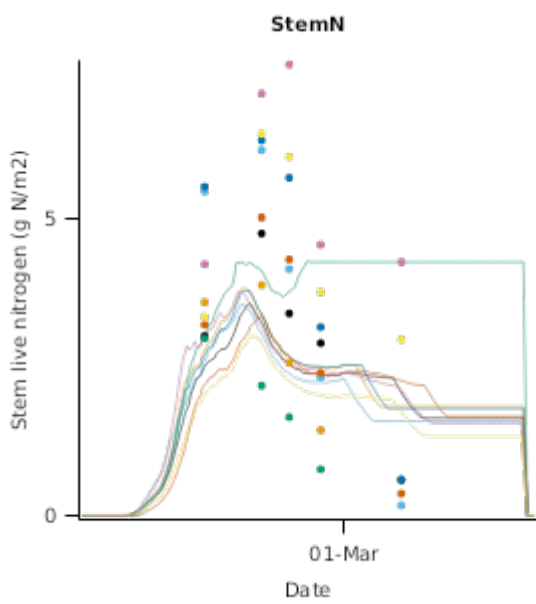
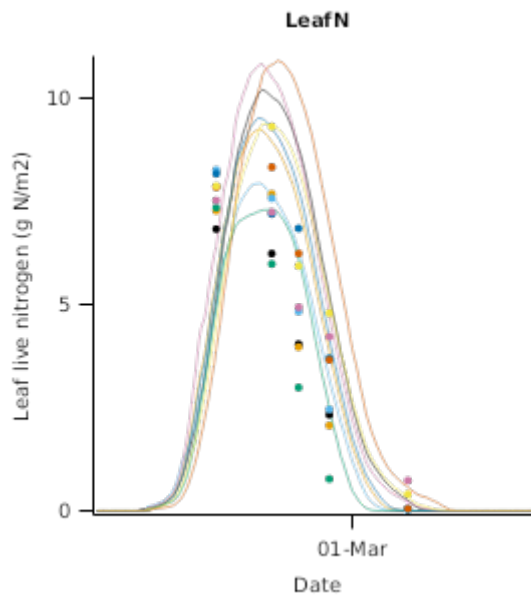
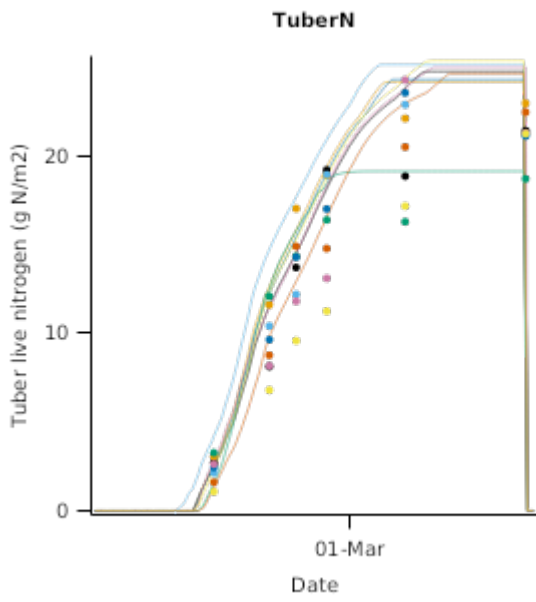
2.2.1.1 Canopy



2.2.1.2 Biomass



2.2.1.3 Nitrogen



2.2.2 Baxters

Sowing date and Sowing Density Experiment

This trial was run in Baxters block of the Plant and Food Research farm on a shallow templeton silt loam, which had a compacted layer from 30-60 cm above a stony sub soil below 60cm. Russet Burbank potatoes were planted on 3 dates:

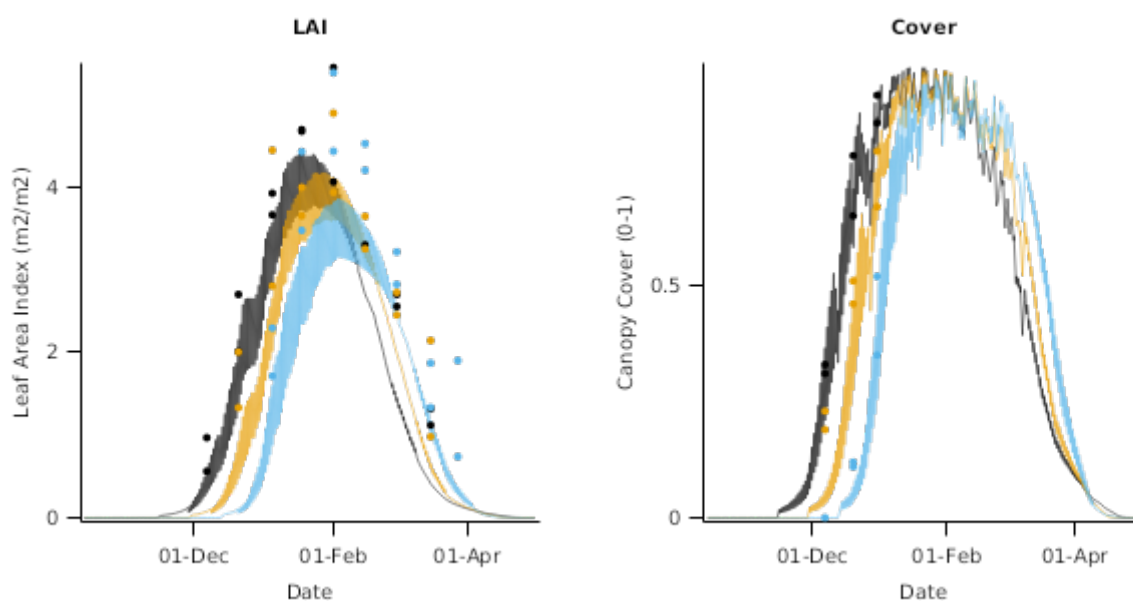
- * 14 October 1999
- * 2 November 1999
- * 23 November 1999

Two populations were established at each planting:

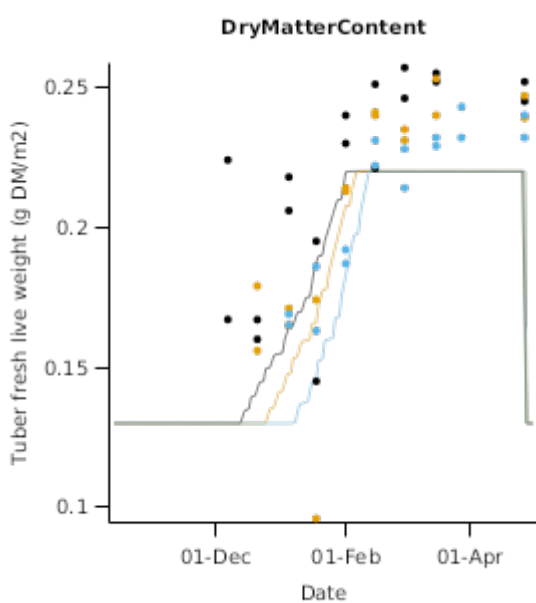
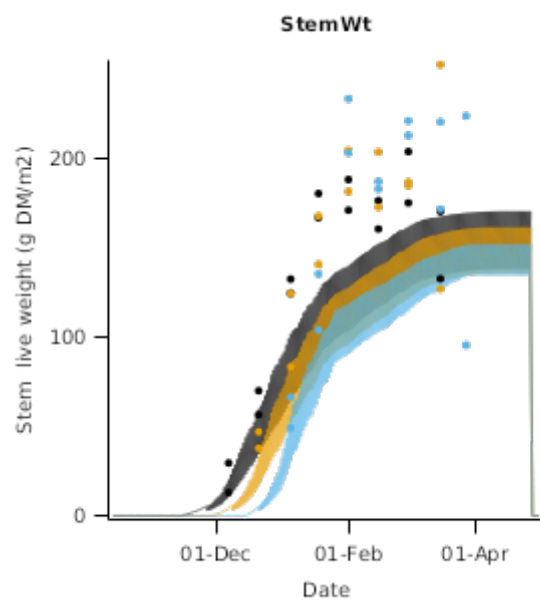
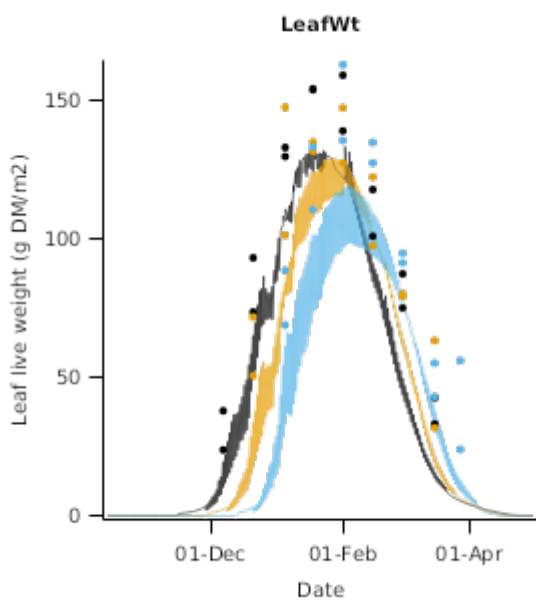
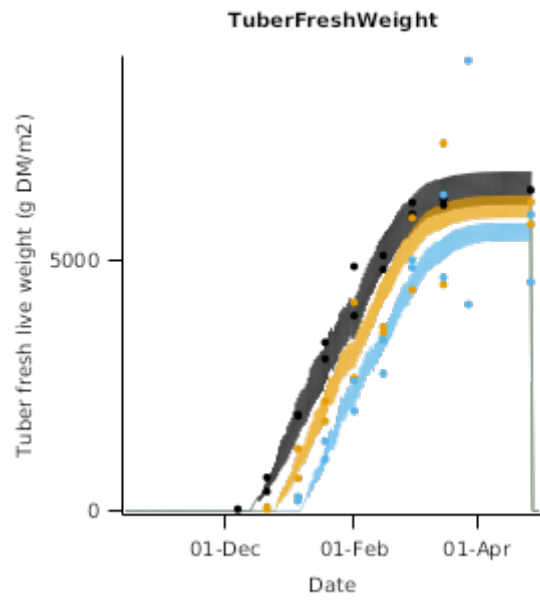
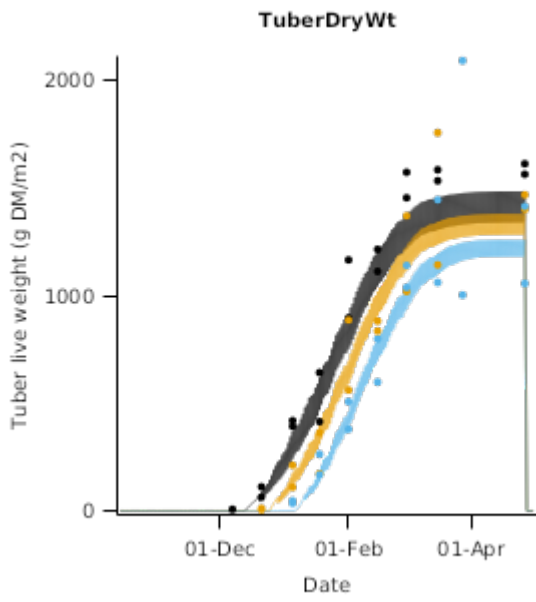
- * 2 Plants/m²
- * 4 Plants/m²

Fertiliser and pesticide were applied to ensure the crop remained unstressed and healthy but irrigation was somewhat limited and did constrain growth at times. Each treatment intensively sampled throughout the growth period to provide data for developing cultivar parameters for different varieties. The same seed line was used for each sowing date so seed was becoming progressively more mature with later plantings. As such the stem number per plant increased from 3 - 5.7 from the early to the late sowing and the final leaf number decreased. The differences in canopy and production between the population treatments were very small because the low population treatment branched extensively. This combination of different stem numbers and populations provided an excellent data set for parameterising branching.

2.2.2.1 Canopy



2.2.2.2 Biomass



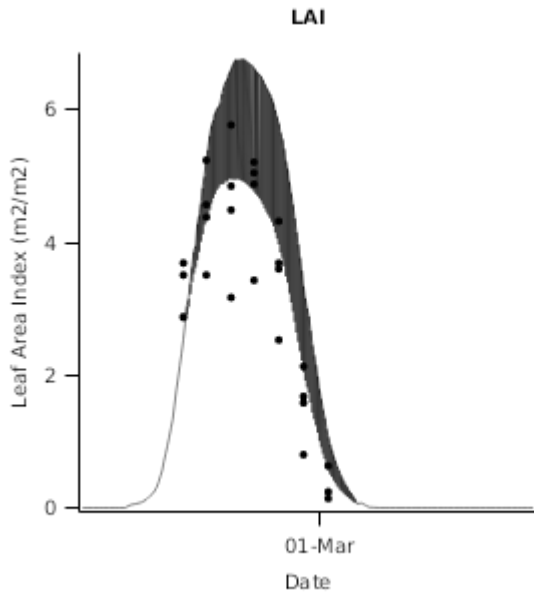
2.2.3 FUE

Fertiliser Use Efficiency experiment (FUE)

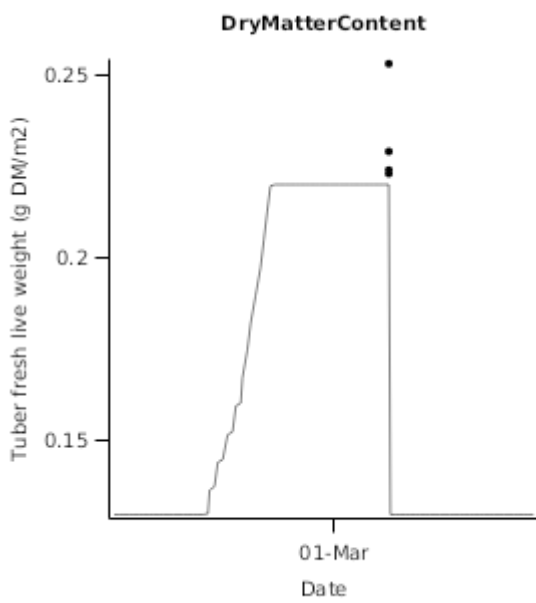
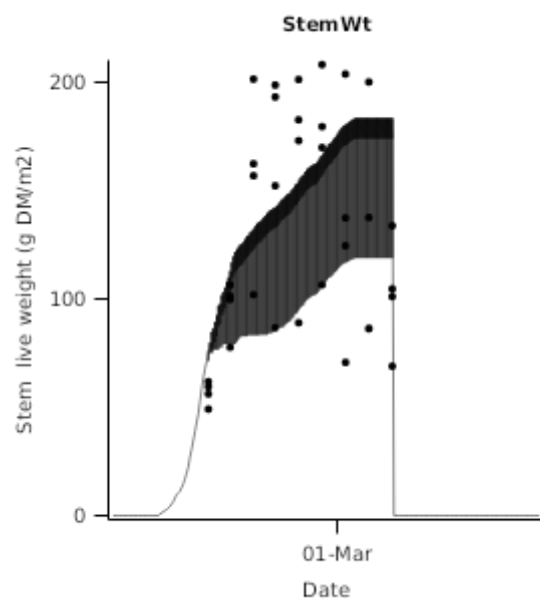
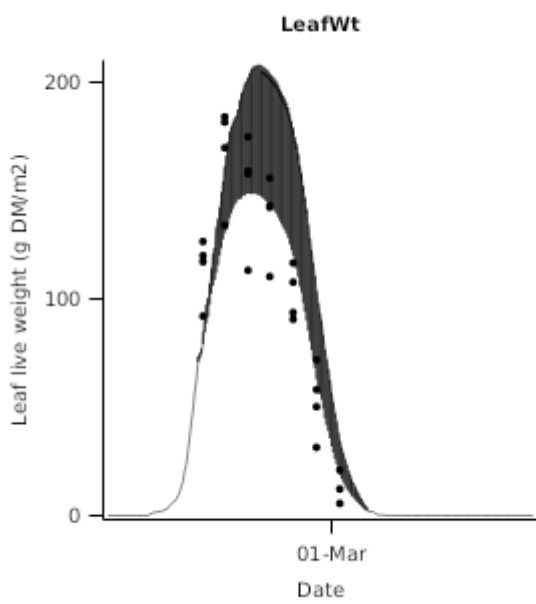
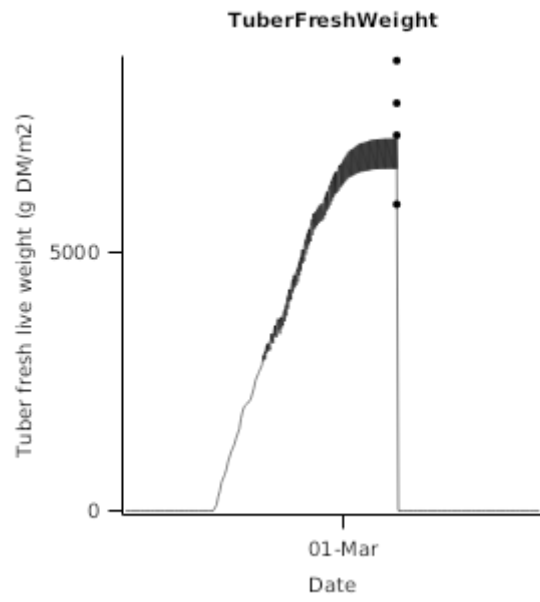
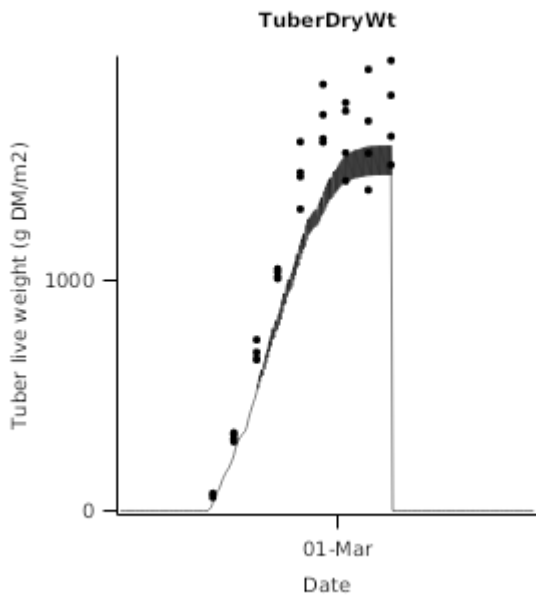
This trial was run in A Block of the Plant and Food Research farm on a templeton silt loam, which is a deep, stone free soil of high plant available water capacity (160 mm/m). Russet burbank potatos were planted on the 21st of October 1999 and irrigation and pesticide were applied to avoid stresses. Different nitrogen fertiliser treatments were applied:

- * Nil
- * 150 kg/ha applied at planting
- * 300 kg/ha applied at planting
- * 275 kg/ha in split application of 25kg/ha weekly

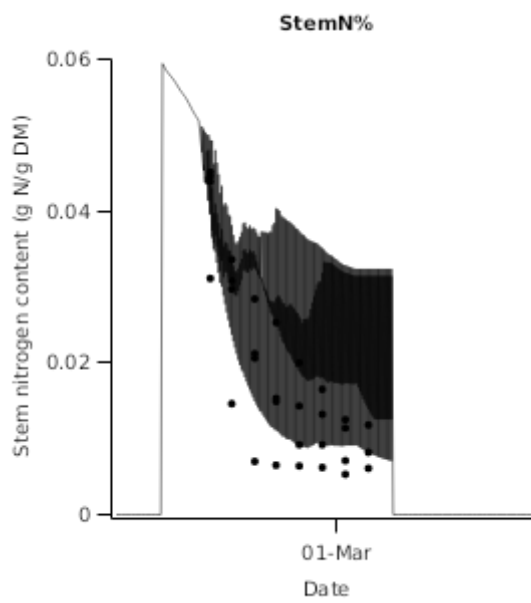
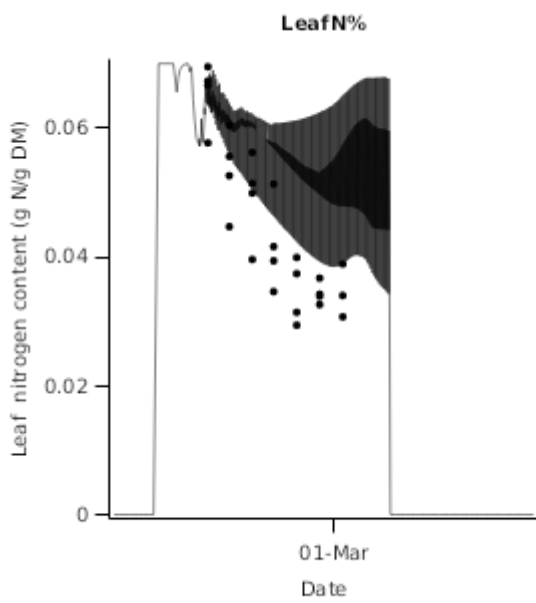
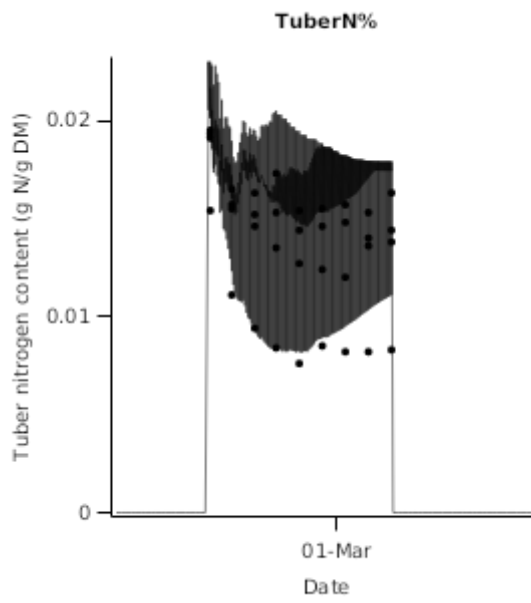
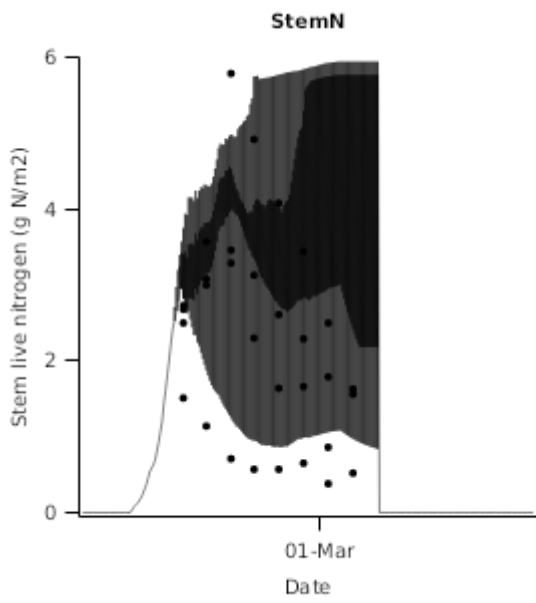
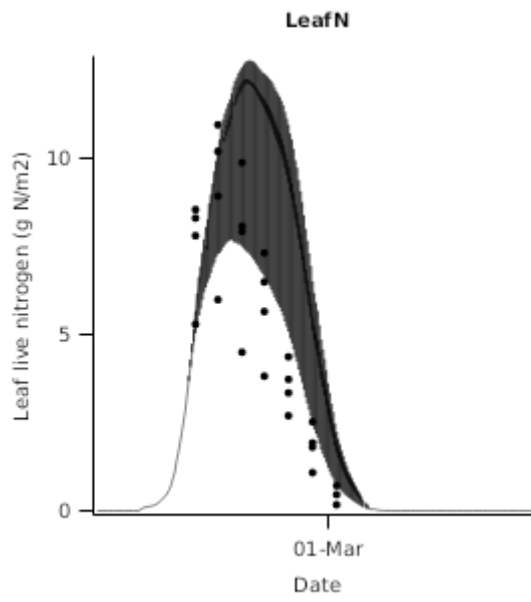
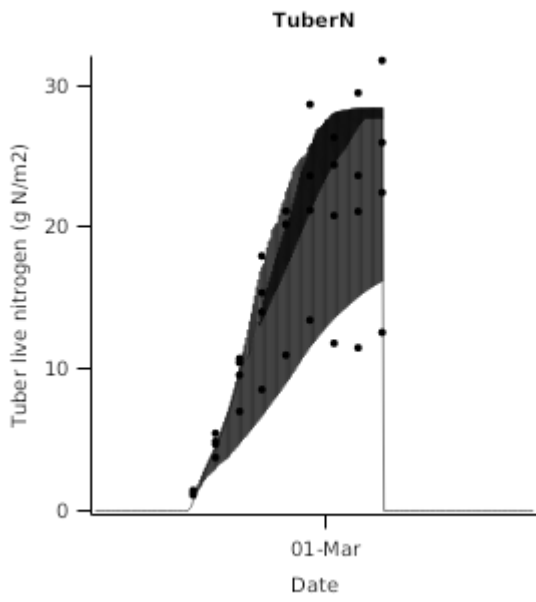
2.2.3.1 Canopy



2.2.3.2 Biomass



2.2.3.3 Nitrogen



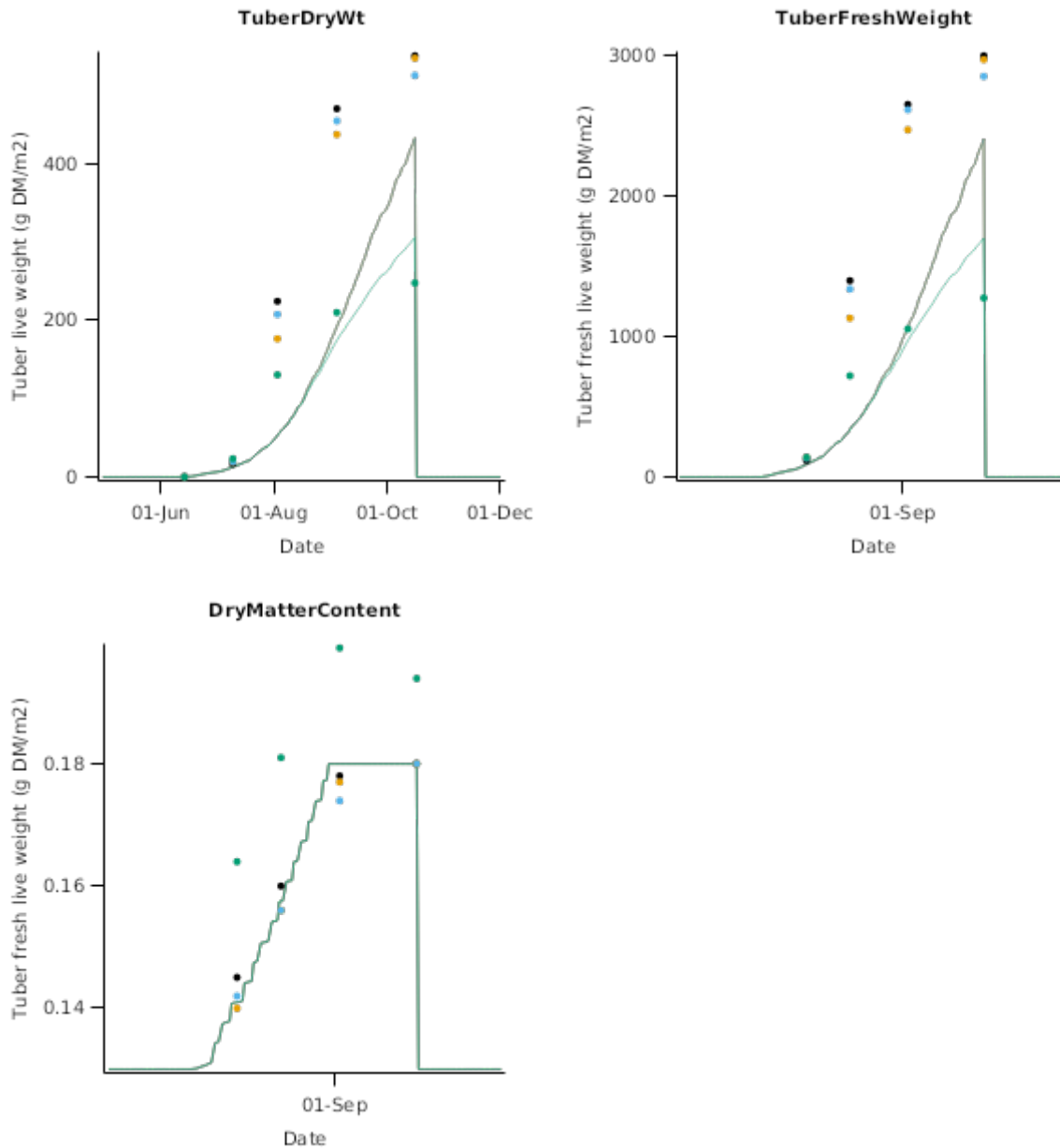
2.2.4 FSP

Fertiliser Use Efficiency experiment (FUE)

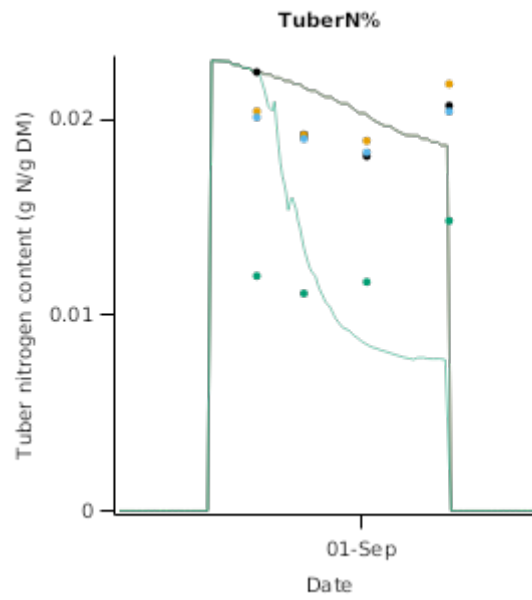
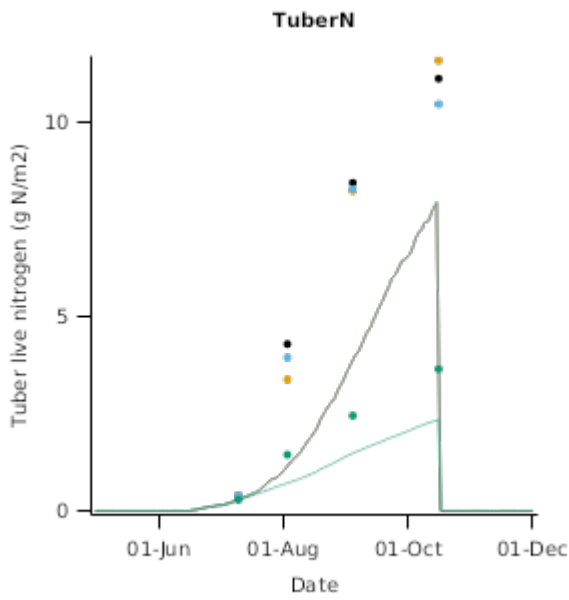
This trial was run in A Block of the Plant and Food Research farm on a templeton silt loam, which is a deep, stone free soil of high plant available water capacity (160 mm/m). Russet burbank potatoes were planted on the 21st of October 1999 and irrigation and pesticide were applied to avoid stresses. Different nitrogen fertiliser treatments were applied:

- * Nil
- * 150 kg/ha applied at planting
- * 300 kg/ha applied at planting
- * 275 kg/ha in split application of 25kg/ha weekly

2.2.4.1 Biomass

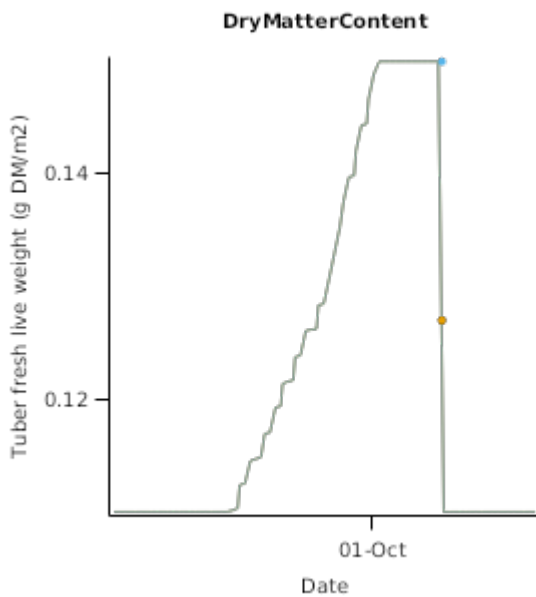
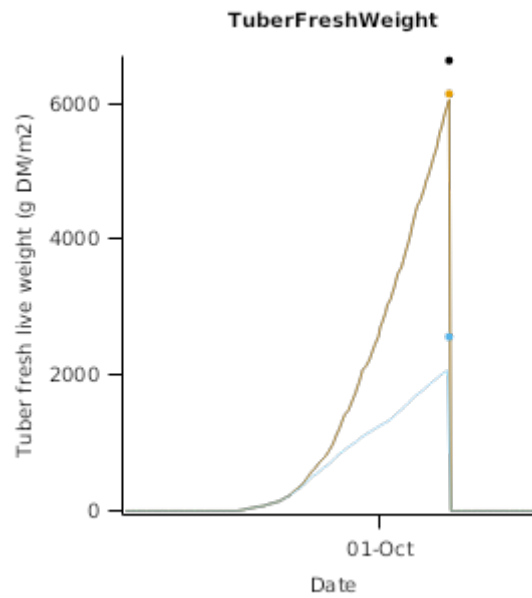
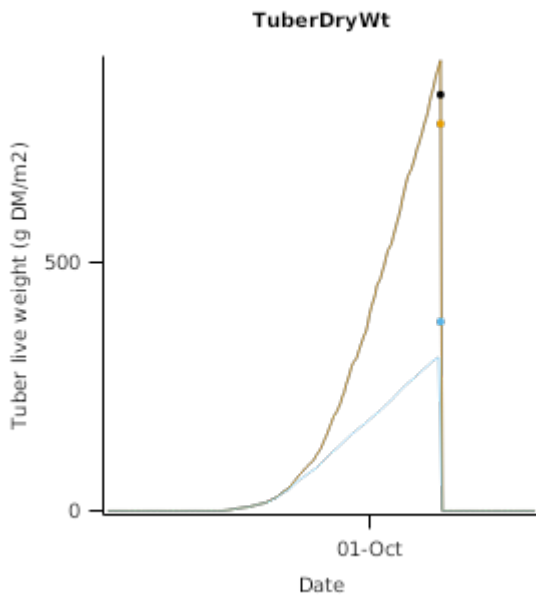


2.2.4.2 Nitrogen

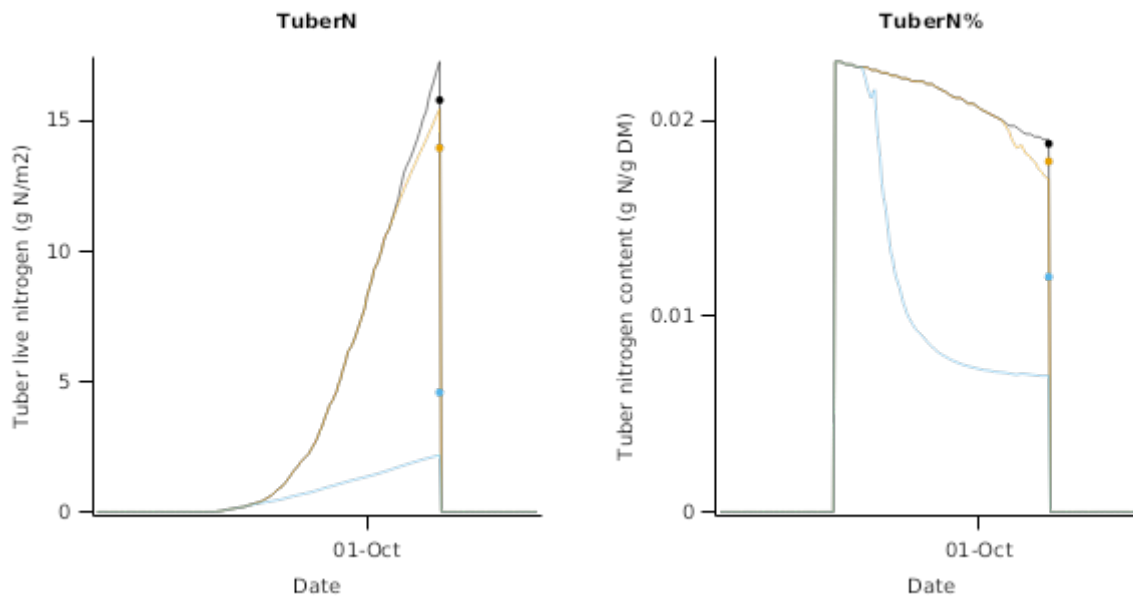


2.2.5 FUE_NI

2.2.5.1 Biomass



2.2.5.2 Nitrogen



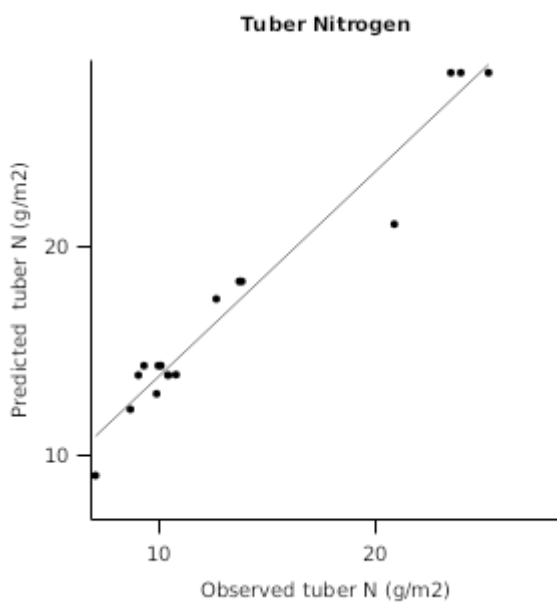
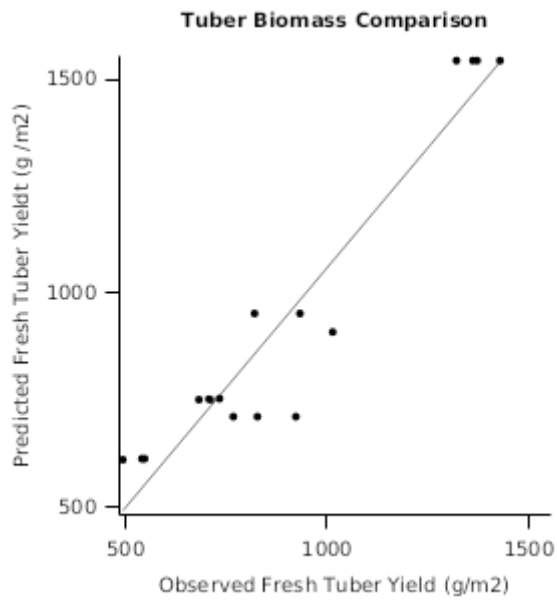
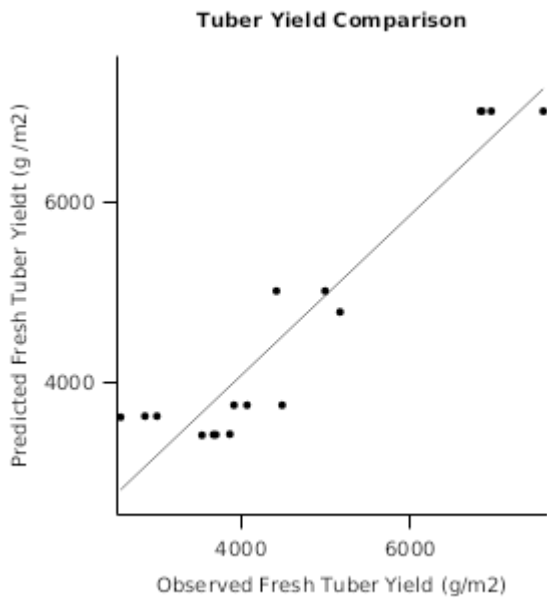
2.3 Australia

Model development

A range of experimental data has been collected by SARDI at Parilla, Penola and Glenroy in South Australia.

List of experiments.

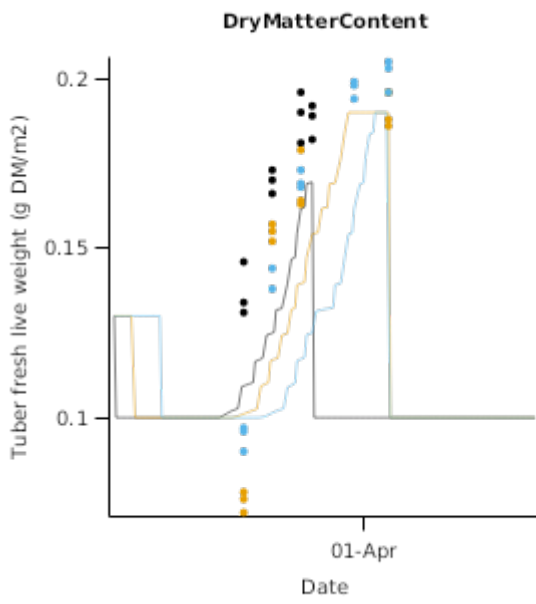
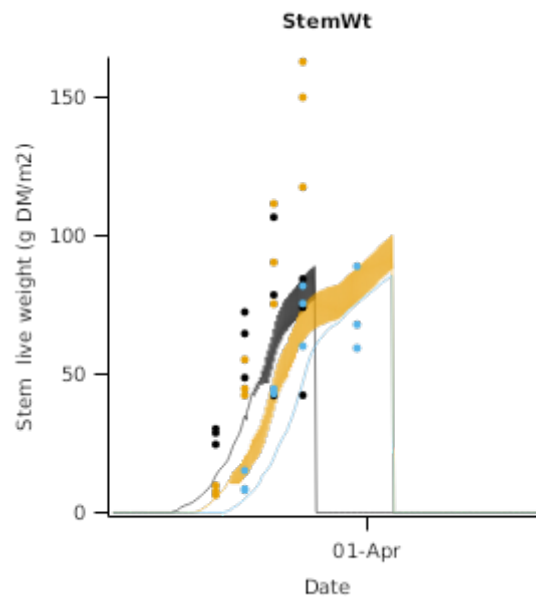
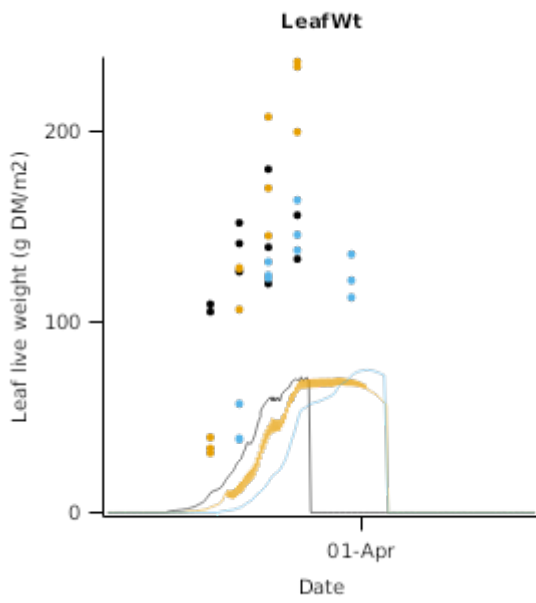
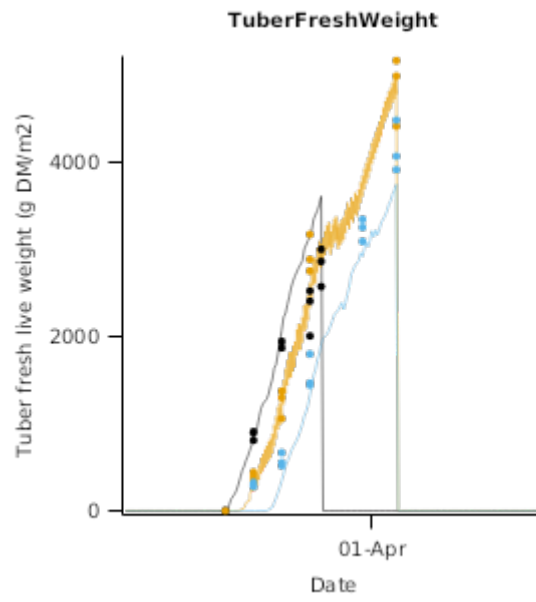
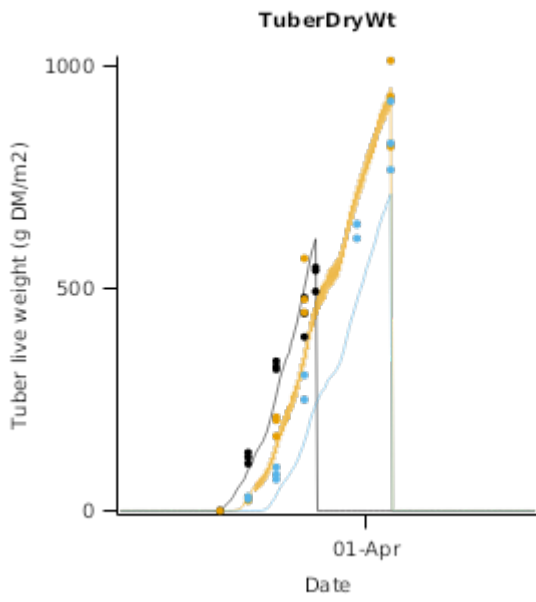
Experiment Name	Design (Number of Treatments)
Parilla	PlantingDate x Nitrogen (9)
Penola_0607	Nitrogen (4)
Glenroy_0607	Nitrogen (4)



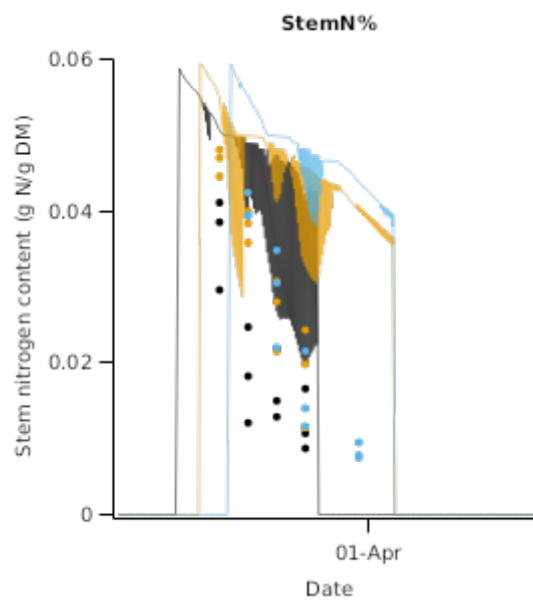
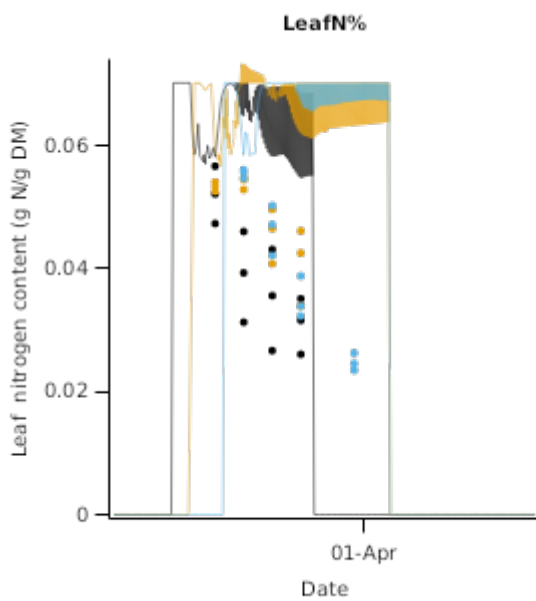
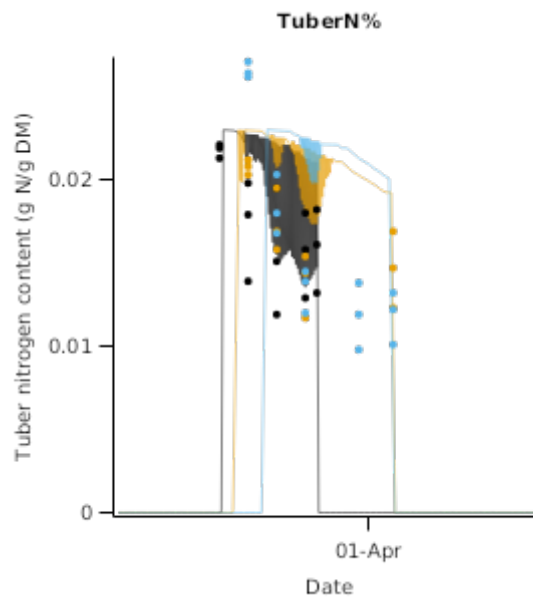
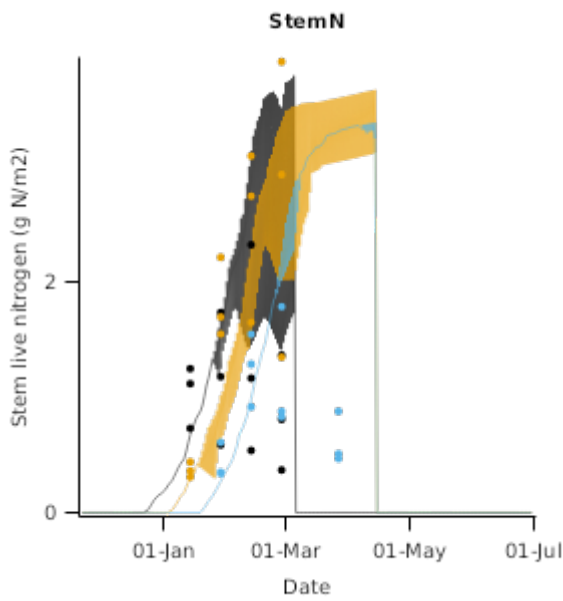
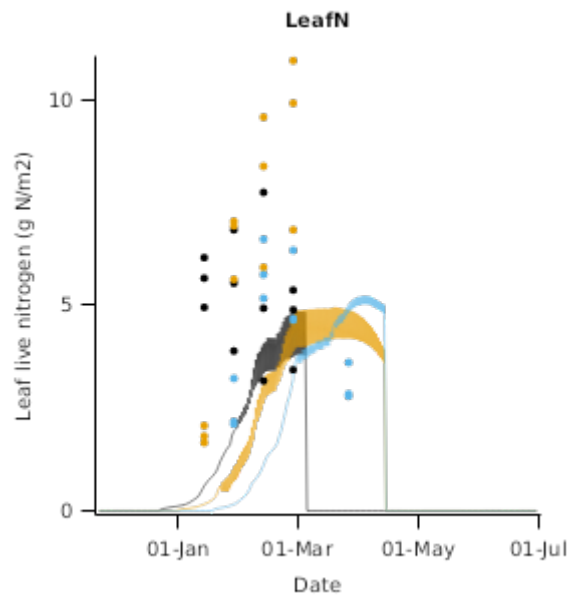
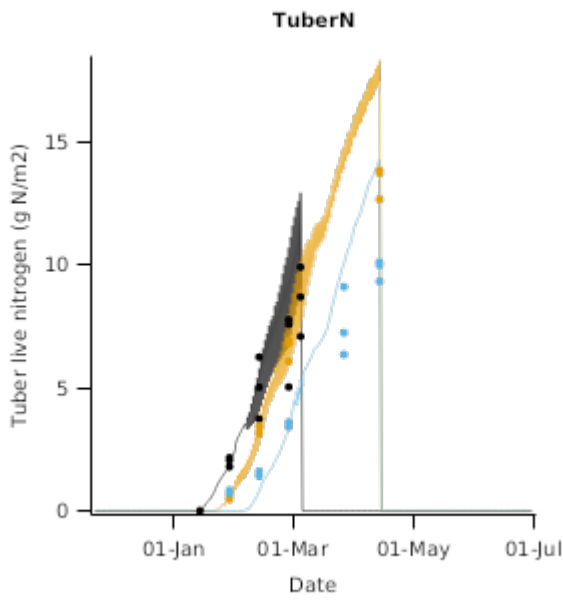
2.3.1 Parilla

This experiment was conducted within a commercial potato field in the South Australia Murrumbidgee region. The fresh potato variety 'Coliban' was planted on three different dates (23/11, 3/12, 17/12) in 2007 on a free draining sandy soil. Treatments received 10 mm of irrigation daily and 260 kg/ha of N as fertigation throughout the experiment. In addition three rates of side dressed N (0, 100 and 200 kg) were applied during the season.

2.3.1.1 Biomass



2.3.1.2 Nitrogen



2.3.2 Penola_0607

This experiment was conducted within a commercial field of 'Russet Burbank' potatoes grown at Penola west in South Australia. The paddock contained a Parapanic, Humic-Humosesquic, Semiaquic, Podsol (sandy loam) soil and the crop

was sown on the 10/10/2006. Nitrogen treatments were applied at different rates (0, 250, 500 and 750 kg/ha) in 8 split applications.

2.3.2.1 Biomass

TuberDryWt

TuberFreshWeight

LeafWt

StemWt

DryMatterContent

2.3.2.2 Nitrogen

TuberN

LeafN

StemN

TuberN%

LeafN%

StemN%

2.3.3 Glenroy_0607

This experiment was conducted within a commercial field of 'Russet Burbank' potatoes grown at Glenroy in South Australia. The paddock contained a Melanic, Petrocalcic, Black Dermosol which had 30 cm of black clay over a compacted calcitrate sub soil. The crop was sown on the 18/10/2005. Nitrogen treatments were applied at different rates (0, 125, 250 and 500 kg/ha) in 8 split applications.

2.3.3.1 Biomass

TuberDryWt

TuberFreshWeight

LeafWt

StemWt

DryMatterContent

2.3.3.2 Nitrogen

TuberN

LeafN

StemN

TuberN%

LeafN%

StemN%

3 References

[Brown, H.E, Huth, N, Holzworth, D., 2011. A potato model build using the APSIM Plant.NET framework., 961-967.](#)

- 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.
- Jones, C.A., Kiniry, J.R., Dyke, P.T., 1986. CERES-Maize: a simulation model of maize growth and development..
- Lawless, Conor, Semenov, MA, Jamieson, PD, 2005. A wheat canopy model linking leaf area and phenology. *European Journal of Agronomy* 22 (1), 19-32.
- 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.
- Oliveira, J.S., 2015. Growth and Development of potato (*Solanum tuberosum* L.) crops after different cool season storage. Thesis. Lincoln University.
- 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.
- Van Keulen, H., Stol, W., 1995. Agro-ecological zonation for potato production. *Potato Ecology And modelling of crops under conditions limiting growth: Proceedings of the Second International Potato Modeling Conference, held in Wageningen 17-19 May, 1994*, Eds: Haverkort, A. J. and MacKerron, D. K. L., 357-371.