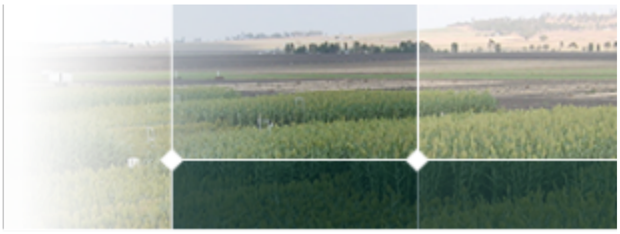


1 The APSIM Sorghum Model

McLean, G., Brider, J., Doherty, A.

The APSIM sorghum 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 consists of:

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

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

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

List of Plant Model Components.

Component Name	Component Type
Arbitrator	Models.PMF.OrganArbitrator
Phenology	Models.PMF.Phen.Phenology
LeafCulms	Models.PMF.Struct.LeafCulms
Grain	Models.PMF.Organs.ReproductiveOrgan
Root	Models.PMF.Organs.Root
Leaf	Models.PMF.Organs.SorghumLeaf
Rachis	Models.PMF.Organs.GenericOrgan
Stem	Models.PMF.Organs.GenericOrgan
TotalPlantDemand	Models.Functions.AddFunction
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 DoPotentialDMAllocation() 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, DoActualDMAllocation 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.

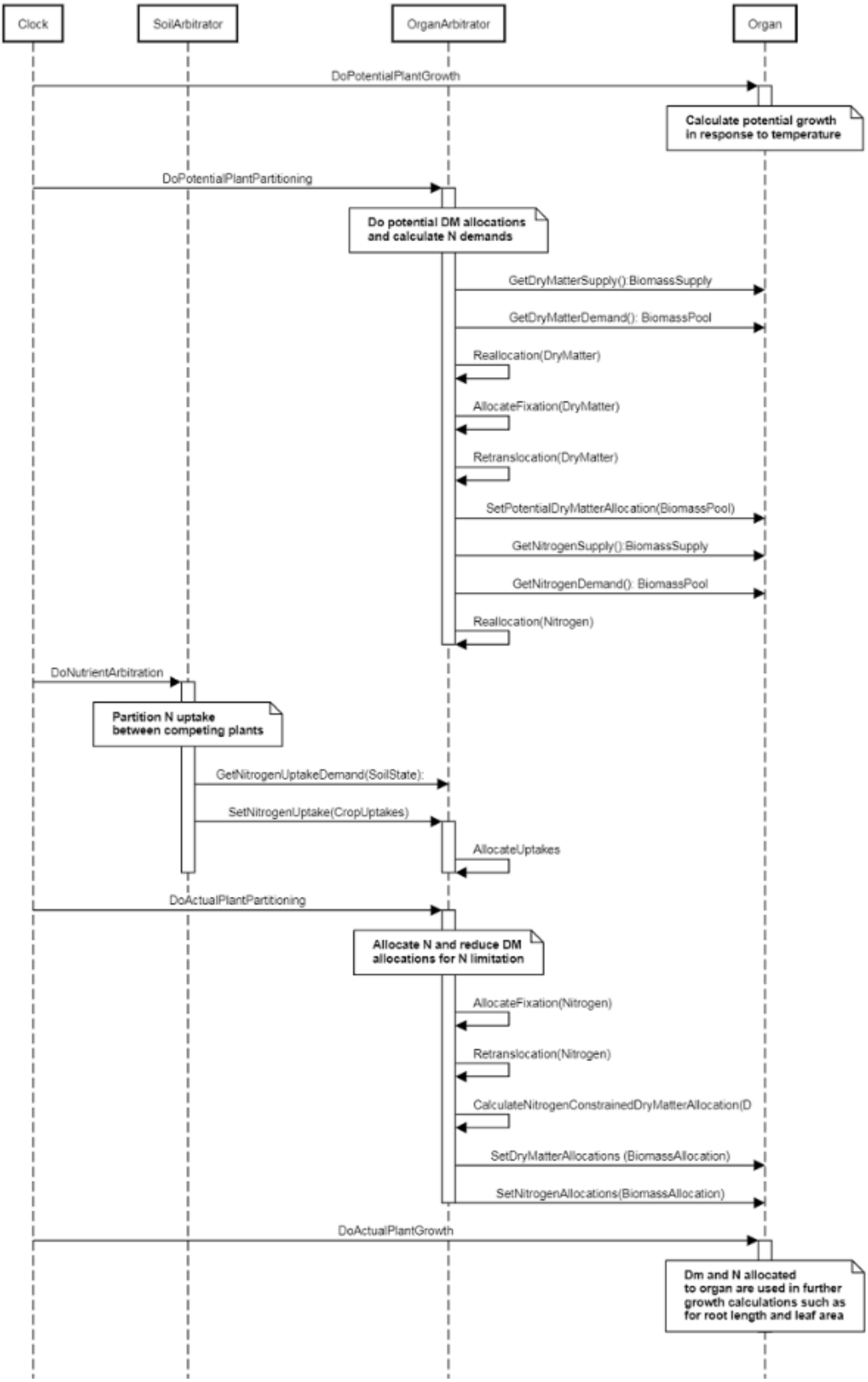


Figure 1: Schematic showing the procedure for arbitration of biomass partitioning. Pink boxes represent events that occur every day and their numbering shows the order of calculations. Blue boxes represent the methods that are called when these events occur. Orange boxes contain properties that make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

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

BeforeFlowering has a value between Germination and Flowering calculated as:

DltTT = [Phenology].DltTT

PostFlowering has a value between Flowering and Maturity calculated as:

DltTTFM = [Phenology].DltTTFM

MaturityToRipe has a value between Maturity and HarvestRipe calculated as:

DltTT = [Phenology].DltTT

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

Phase Number	Phase Name	Initial Stage	Final Stage
1	Germinating	Sowing	Germination
2	Emerging	Germination	Emergence
3	Juvenile	Emergence	EndJuvenile
4	JuvenileToFloralInit	EndJuvenile	FloralInitiation
5	FloralInitToFlagLeaf	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:

Target = SowingDepth x *ShootRate* + *ShootLag*

Where:

ShootRate = 0.6 (deg day/mm),

ShootLag = 15 (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:

ThermalTime = [Phenology].ThermalTime

1.2.4 Juvenile

This phase goes from emergence to endjuvenile.

The *Target* for completion is calculated as:

Target = 100 (oD)

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

Progression = [Phenology].ThermalTime

1.2.5 JuvenileToFloralInit

This phase goes from endjuvenile to floralinitiation.

The *Target* for completion is calculated as:

Target = [Phenology].TTTargetPhotoSensitive

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

Progression = [Phenology].ThermalTime

1.2.6 FloralInitToFlagLeaf

This phase goes from floralinitiation to flagleaf.

The *Target* for completion is calculated as:

Target = [Phenology].TTEmergToFlagLeaf - [Phenology].Juvenile.Target - [Phenology].JuvenileToFloralInit.Target

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

Progression = [Phenology].ThermalTime

1.2.7 FlagLeafToFlowering

This phase goes from flagleaf to flowering.

The *Target* for completion is calculated as:

Target = 170 (oD)

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 = 80 (oD)

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

Progression = [Phenology].DltTTFM

1.2.9 GrainFilling

This phase goes from startgrainfill to endgrainfill.

The *Target* for completion is calculated as:

Target = [Phenology].TTFlowerToMaturity - [Phenology].FloweringToGrainFilling.Target - [Phenology].TTGrainFillToMaturity

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

Progression = [Phenology].DltTTFM

1.2.10 Maturing

This phase goes from endgrainfill to maturity.

The *Target* for completion is calculated as:

Target = [Phenology].TTGrainFillToMaturity

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

Progression = [Phenology].DltTTFM

1.2.11 MaturityToHarvestRipe

This phase goes from maturity to harvestripe.

The *Target* for completion is calculated as:

Target = 1 (oD)

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

Progression = [Phenology].ThermalTime

1.2.12 ReadyForHarvesting

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

1.2.13 Constants

TTFlowerToMaturity = 761

TTEndJuvToInit = 160

LeafNoAtEmergence = 1

LeafNoRateChange = 3.5

1.2.14 DltTT

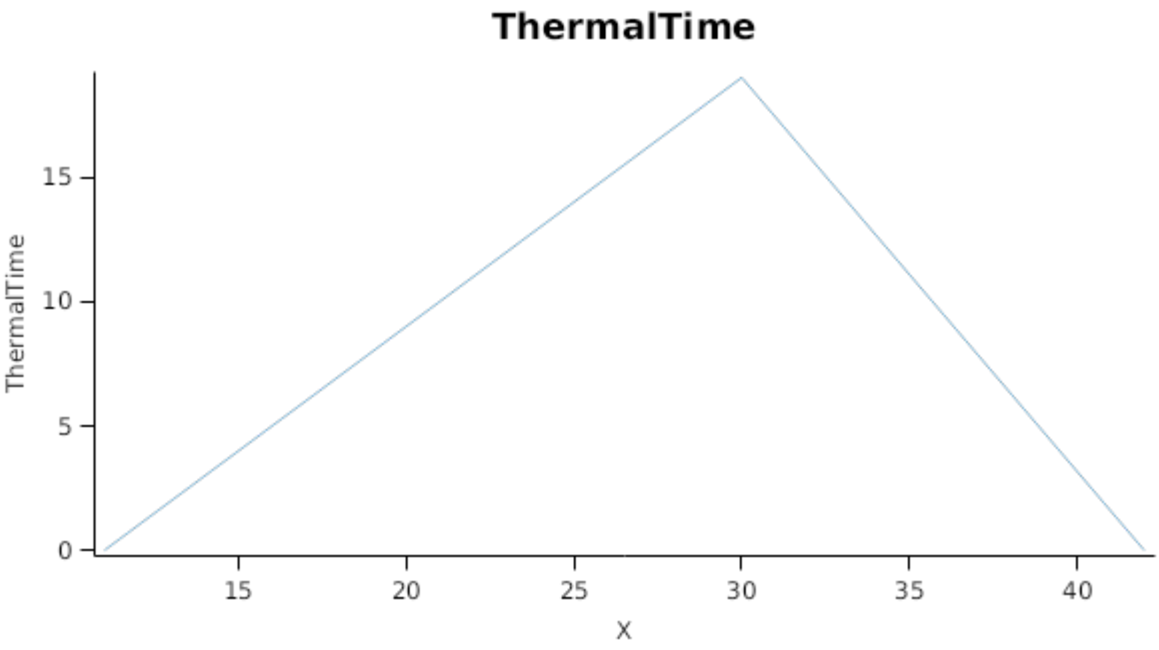
DltTT = ThermalTime x *Stress*

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

Firstly 3-hourly estimates of air temperature (Ta) are interpolated using the method of Jones et al., 1986 which assumes a sinusoidal temperature. pattern between Tmax and Tmin.

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

X	ThermalTime
11.0	0.0
30.0	19.0
42.0	0.0



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

TT has a value between Sowing and EndJuvenile calculated as:

PhenoWaterStress = [Phenology].PhenoWaterStress

PreFlowering has a value between EndJuvenile and Flowering calculated as:

MinimumFunction = Min(*PhenoWaterStress*, *MaximumFunction*)

Where:

PhenoWaterStress = [Phenology].PhenoWaterStress

MaximumFunction = Max(*Constant*, *NitrogenPhenoStress*)

Where:

Constant = 0.5

NitrogenPhenoStress = [Leaf].NitrogenPhenoStress

PostFlowering has a value between Flowering and HarvestRipe calculated as:

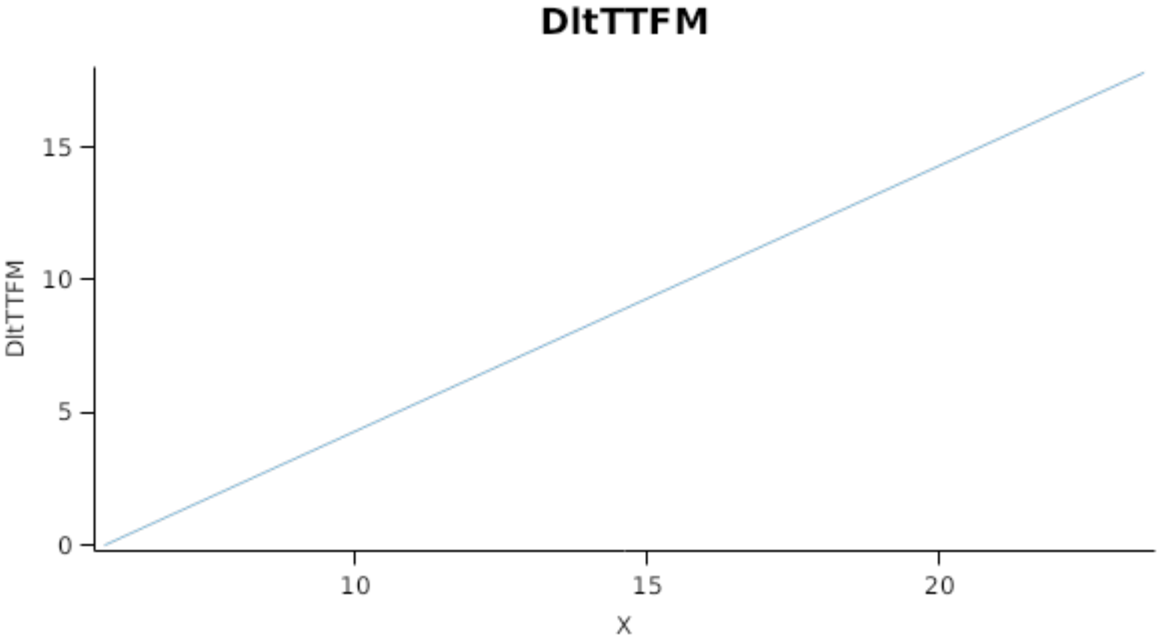
Constant = 1

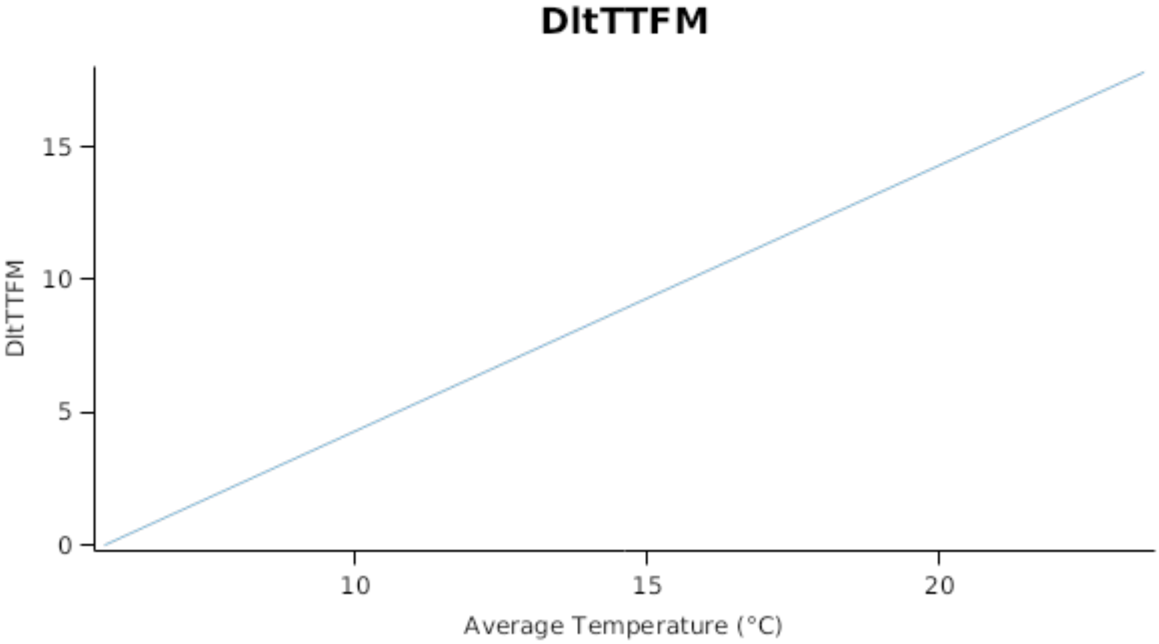
1.2.15 DltTTFM

DltTTFM 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	DltTTFM
5.7	0.0
23.5	17.8





1.2.16 Photoperiod

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

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

Twilight = -2.2 (degrees)

1.2.17 FloweringDAS

Before Flowering

PreEventValue = 0

On Flowering the value is set to:

PostEventValue = [Plant].DaysAfterSowing

1.2.18 MaturityDAS

Before Maturity

PreEventValue = 0

On Maturity the value is set to:

PostEventValue = [Plant].DaysAfterSowing

1.2.19 TTGrainFillToMaturity

TTGrainFillToMaturity = [Phenology].TTFlowerToMaturity x *Proportion*

Proportion = 0.05

1.2.20 TTTargetPhotoSensitive

TTTargetPhotoSensitive 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 Sowing and Emergence calculated as:

PreEventValue = [Phenology].TTEndJuvToInit

EmergenceToEndJuve has a value between Emergence and EndJuvenile calculated as:

Add = [Phenology].PhotoModifier + [Phenology].TTEndJuvToInit

EndJuveToMaturity has a value between EndJuvenile and HarvestRipe calculated as:

Before EndJuvenile

PreEventValue = [Phenology].TTEndJuvToInit

On EndJuvenile the value is set to:

PostEventValue = [Phenology].PhotoModifier + [Phenology].TTEndJuvToInit

1.2.21 FinalLeafNo

FinalLeafNo = [Leaf].FinalLeafNo

1.2.22 TTEmergToFlagLeaf

TTEmergToFlagLeaf = LeafAppearingPhase1 + *LeafAppearingPhase2*

LeafAppearingPhase1 = NoLeavesRate1 x *LeafAppearanceRate1*

NoLeavesRate1 = [Phenology].FinalLeafNo - [Phenology].LeafNoAtEmergence - [Phenology].LeafNoRateChange

LeafAppearanceRate1 = 41

LeafAppearingPhase2 = LeafAppearanceRate2 x [Phenology].LeafNoRateChange

LeafAppearanceRate2 = 20

1.2.23 TTFromEmergence

TTFromEmergence = Accumulated *ThermalTime* between emergence and maturity

ThermalTime = [Phenology].CurrentPhase.ProgressionForTimeStep

1.2.24 TTFMFromFlowering

TTFMFromFlowering = Accumulated *ThermalTime* between flowering and maturity

ThermalTime = [Phenology].CurrentPhase.ProgressionForTimeStep

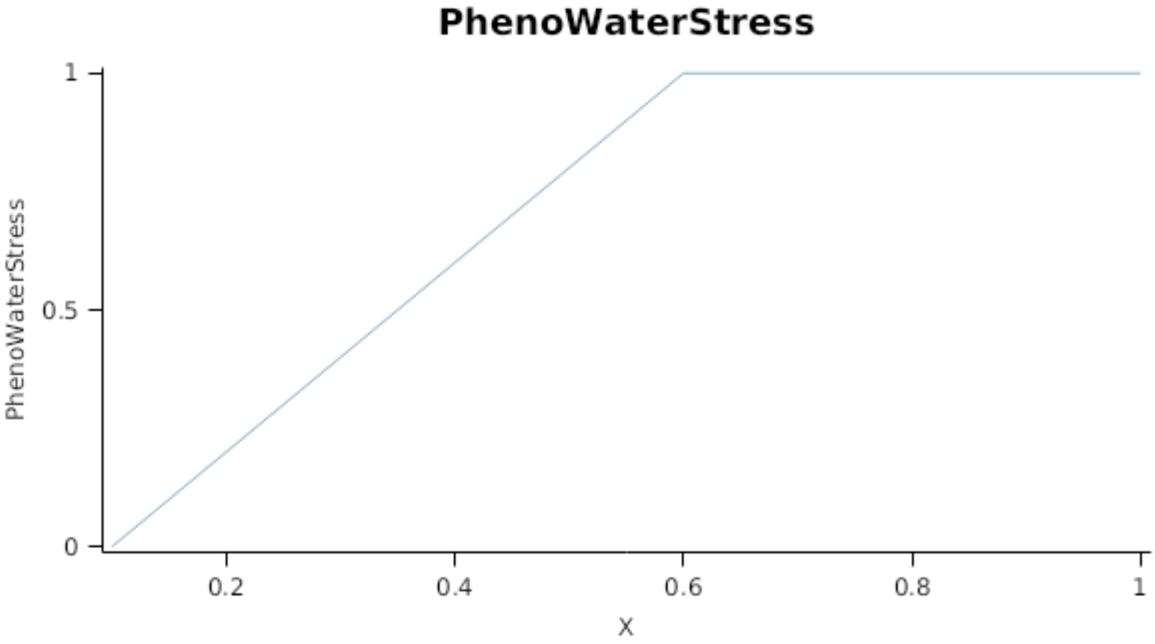
1.2.25 TTEmergToFloralInit

TTEmergToFloralInit = [Phenology].Juvenile.Target + [Phenology].JuvenileToFloralInit.Target

1.2.26 PhenoWaterStress

PhenoWaterStress is calculated using linear interpolation

X	PhenoWaterStress
0.1	0.0
0.6	1.0
1.0	1.0



1.2.27 TTFMSgfToMaturity

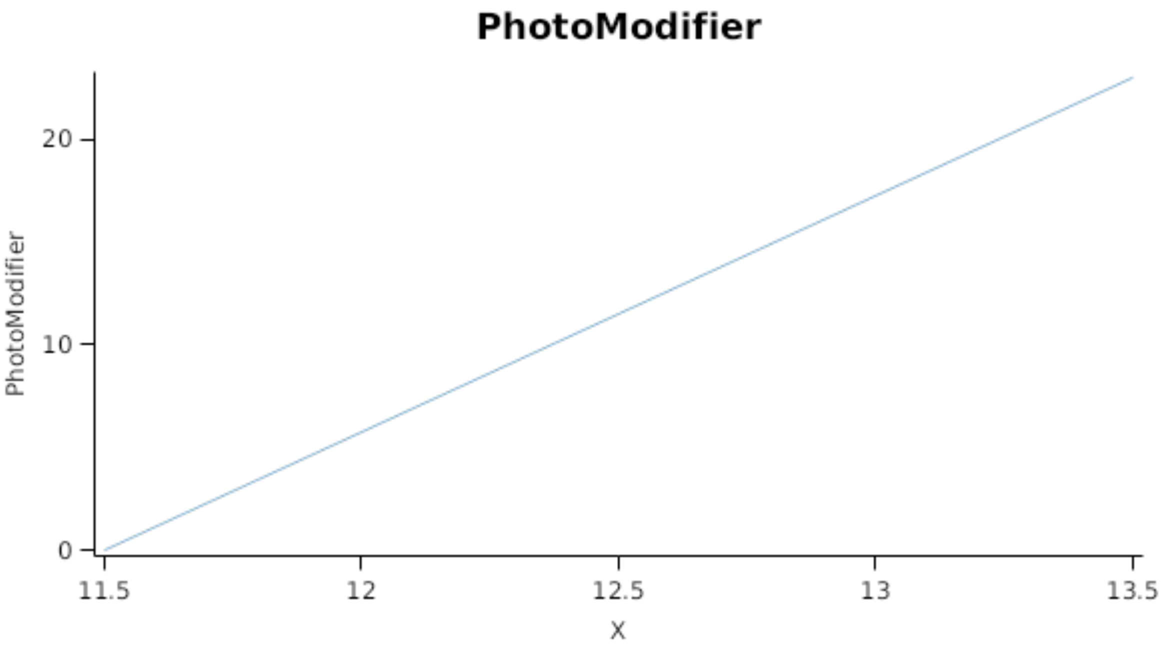
TTFMSgfToMaturity = Accumulated *ThermalTime* between startgrainfill and maturity

ThermalTime = [Phenology].CurrentPhase.ProgressionForTimeStep

1.2.28 PhotoModifier

PhotoModifier is calculated using linear interpolation

X	PhotoModifier
11.5	0.0
13.5	23.0



1.2.29 TTToFlowering

TTToFlowering = [Emerging].Target + [Juvenile].Target + [JuvenileToFloralInit].Target + [FloralInitToFlagLeaf].Target + [FlagLeafToFlowering].Target

1.3 LeafCulms

1.3.1 LeafCulms

LeafCulms model ported from LeafCulms and LeafCulms_Fixed in the apsim classic sorghum model.

1.3.1.1 TODO:

- Implement constants as IFunctions.
- Fix case to match style guidelines.

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

InitialGrainProportion = 0.05

MaximumPotentialGrainSize = 0.3

MinimumNConc = 0.008

WaterContent = 0.13

DMPerSeed = 0.00083

DMConversionEfficiency = 1

RemobilisationCost = 0

CarbonConcentration = 0.4

1.4.2 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:

InitialNconc = 0.05

LinearPhase has a value between StartGrainFill and Maturity calculated as:

FinalNconc = 0.0175

1.4.3 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 StartGrainFill and Maturity calculated as:

IF gfFract < GrainFillFraction THEN

InitialGrainFilling = Min(*FillingRate*, *FinalGrainFilling*)

Where:

FillingRate = [Grain].NumberFunction x [Phenology].DltTTFM x *DivideFunction*

DivideFunction = *GrainFillRate* / *Thousand*

GrainFillRate = 0.001 (mg/grain/dd)

Thousand = 1000 (mg/g)

FinalGrainFilling = [Grain].potentialDMAllocation.Structural x [Grain].MaximumNConc

ELSE

FinalGrainFilling = [Grain].potentialDMAllocation.Structural x [Grain].MaximumNConc

1.4.4 FinalGrainNum

Before StartGrainFill

PreEventValue = 0

On StartGrainFill the value is set to:

PostEventValue = *GrowthRate* / [Grain].DMPerSeed

GrowthRate = *PlantGrowth* / *DaysFIToStartGrainFill*

Maximum function here prevents grain from returning negative dm demand in the event that plant growth is negative.

PlantGrowth = Max(*PlantGrowthSinceFloralInit*, *Zero*)

Where:

PlantGrowthSinceFloralInit = [Sorghum].Total.Wt - *GreenWtAtFI*

Before FloralInitiation

PreEventValue = 0

On FloralInitiation the value is set to:

PostEventValue = [Sorghum].Total.Wt

Zero = 0

DaysFIToStartGrainFill = Accumulated *One* between floralinitiation and startgrainfill

One = 1

1.4.5 NumberFunction

1.4.5.1 NumberFunction

Calculates the current grain number.

1.4.6 DMCaryopsis

DMCaryopsis = [Arbitrator].DM.TotalFixationSupply / [Grain].NumberFunction / [Phenology].DltTTFM

1.4.7 PotGrainFillRate

PotGrainFillRate = Min(*MultiplyFunction*, *MaxGrainFillRate*)

Where:

MultiplyFunction = *AddFunction* x 1000

AddFunction = *MultiplyFunction* + 3.19E-05

MultiplyFunction = [Grain].DMCaryopsis x 0.4026

MaxGrainFillRate = 0.09

1.4.8 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.

SowingToMaturity has a value between Sowing and Maturity calculated as:

DMDemandFunction = [Grain].PotGrainFillRate x [Phenology].DltTTFM x [Grain].NumberFunction x 0.001

1.4.9 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

1.4.10 DMDemandPriorityFactors

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

Structural = 1

Metabolic = 1

Storage = 1

1.4.11 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.

Root Depth Increase = RootFrontVelocity x XF_i *x 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

$RAw_i = -WaterUptake_i / LiveRootWt_i \times LayerThickness_i \times ProportionThroughLayer$

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

$DMAllocated_i = TotalDMAllocated \times RAw_i / TotalRAw$

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 then 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 (kNO3 or kNH4), the concentration of N form (ppm), and a soil moisture factor (NUptakeSWFactor) which typically decreases as the soil dries. $NO3\ uptake = NO3_i \times kNO3 \times NO3_{ppm,\ i} \times NUptakeSWFactor$ _NH4 uptake = $NH4_i$ x kNH4 x $NH4_{ppm,\ i}$ x 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 demandeach 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

SoilWaterEffect = 1

TemperatureEffect = 1

MaxDailyNUptake = 20

SenescenceRate = 0.002

MaximumNConc = 0.02

MinimumNConc = 0.002

MaximumRootDepth = 1000000

SpecificRootLength = 100000 (m/g)

DMConversionEfficiency = 1

MaintenanceRespirationFunction = 1

RemobilisationCost = 0

CarbonConcentration = 0.4

Setting this to 0 will force sorghum to use the generic uptake *and* root growth routines

RootFrontCalcSwitch = 1

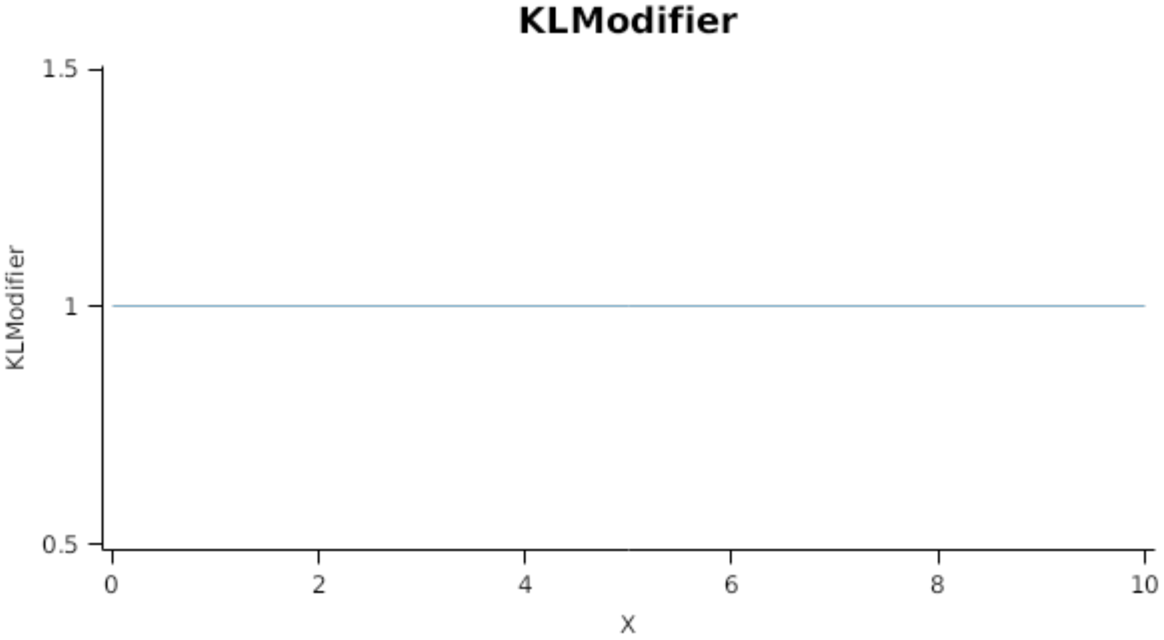
1.5.7 RootShape

This model calculates the proportion of each soil layer occupided by roots. The formula used for the circle is wrong as it does not account for the coordinate of the centre!

1.5.8 KLModifier

KLModifier is calculated using linear interpolation

X	KLModifier
0.0	1.0
10.0	1.0



1.5.9 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.

PreGrainFill has a value between Germination and StartGrainFill calculated as:

Function = 27

GrainfillToMaturity has a value between StartGrainFill and Maturity calculated as:

Function = 0

1.5.10 NitrogenDemandSwitch

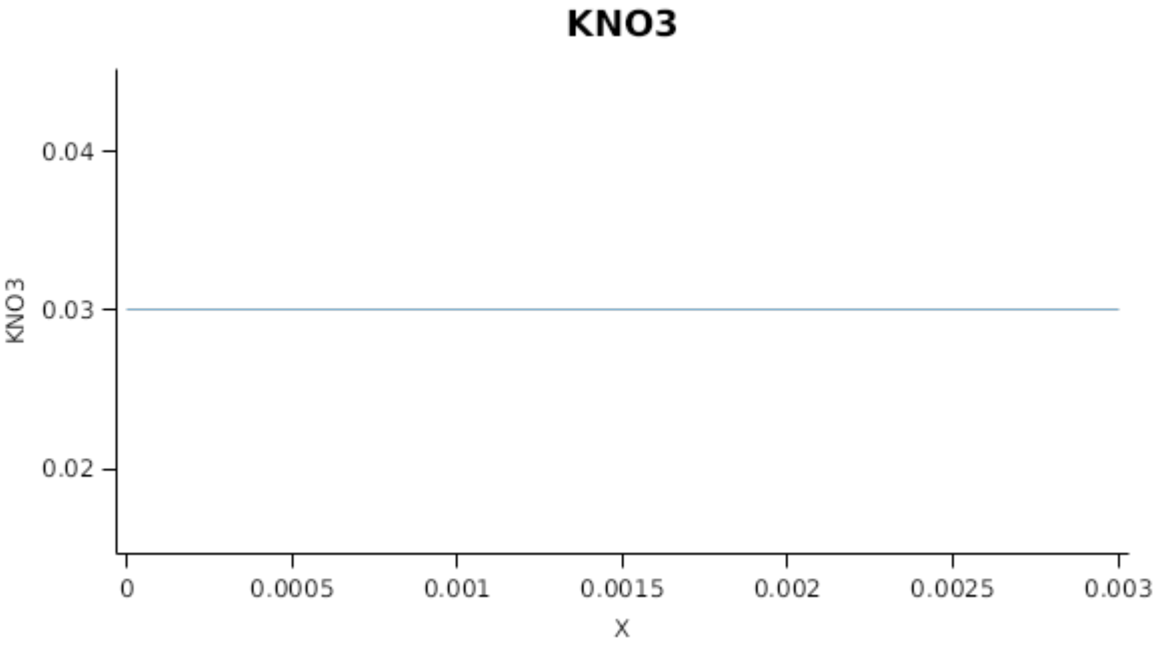
NitrogenDemandSwitch has a value between Germination and Maturity calculated as:

Constant = 1

1.5.11 KNO3

KNO3 is calculated using linear interpolation

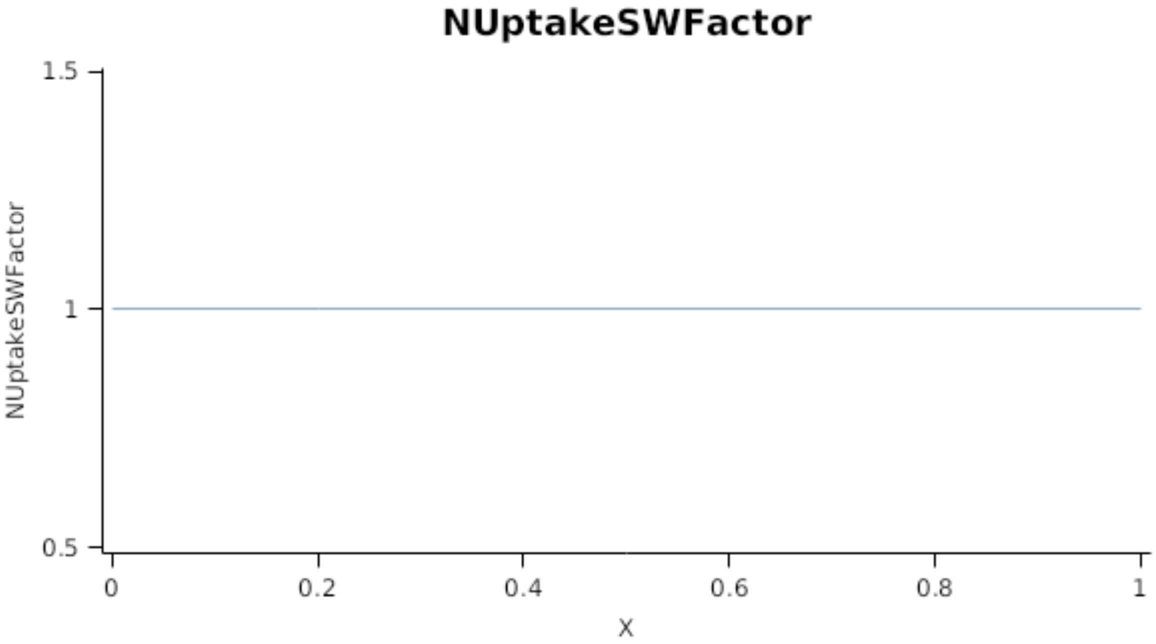
X	KNO3
0.0	0.0
0.0	0.0



1.5.12 NUptakeSWFactor

NUptakeSWFactor is calculated using linear interpolation

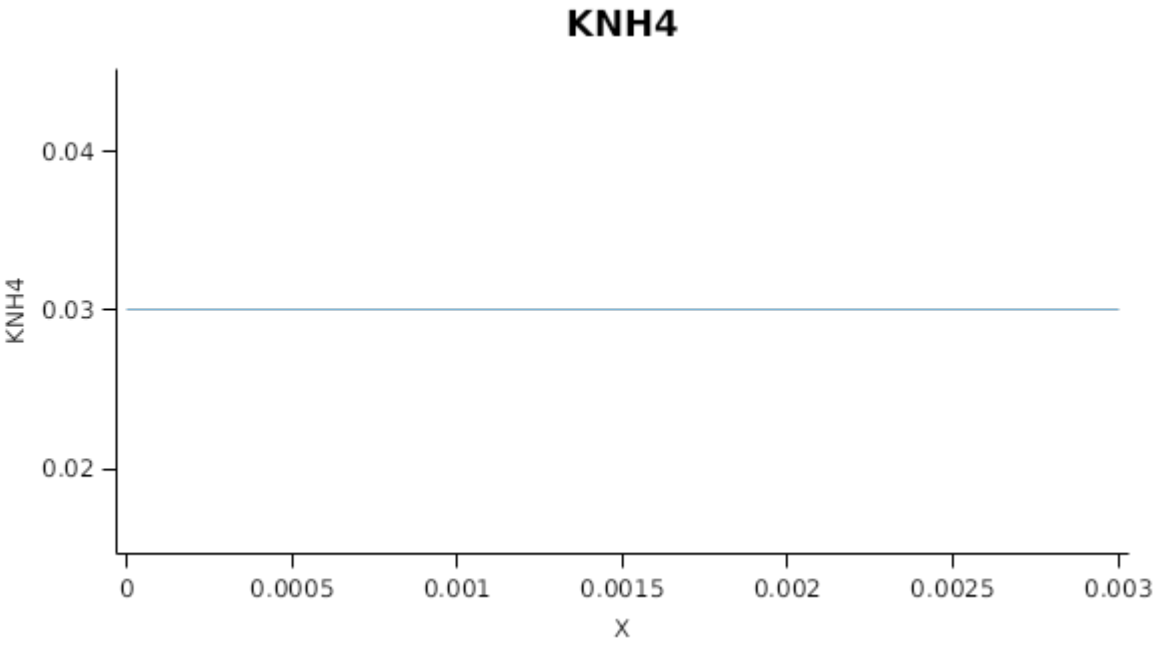
X	NUptakeSWFactor
0.0	1.0
0.2	1.0
1.0	1.0



1.5.13 KNH4

KNH4 is calculated using linear interpolation

X	KNH4
0.0	0.0
0.0	0.0



1.5.14 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

1.5.15 DMDemands

1.5.15.1 DMDemands

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

Structural = *ratioRootShoot* x [Leaf].DMSupplyFunction

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

This is a straight lookup table not an lineaer relationship

BetweenEmergenceAndFloralInit has a value between Emergence and FloralInitiation calculated as:

rootFraction = 1 (0-1)

BetweenFloralInitAndFlowering has a value between FloralInitiation and Flowering calculated as:

rootFraction = 0.33 (0-1)

BetweenFloweringAndGrainFill has a value between Flowering and StartGrainFill calculated as:

rootFraction = 0.087 (0-1)

BetweenFloweringAndGrainFill1 has a value between StartGrainFill and HarvestRipe calculated as:

rootFraction = 0 (0-1)

Metabolic = 0

Storage = 0

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

1.5.16 NDemands

1.5.16.1 NDemands

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

Structural = Max(*zero*, *SubtractFunction*)

Where:

zero = 0

SubtractFunction = * required* - [Root].Live.N

* required* = [Root].minimumNconc x *AddFunction*

AddFunction = [Root].potentialDMAllocation.Structural + [Root].Live.Wt

Metabolic = 0

Storage = 0

QStructuralPriority = 1

QMetabolicPriority = 1

QStoragePriority = 1

1.5.17 NitrogenRootCalcSwitch

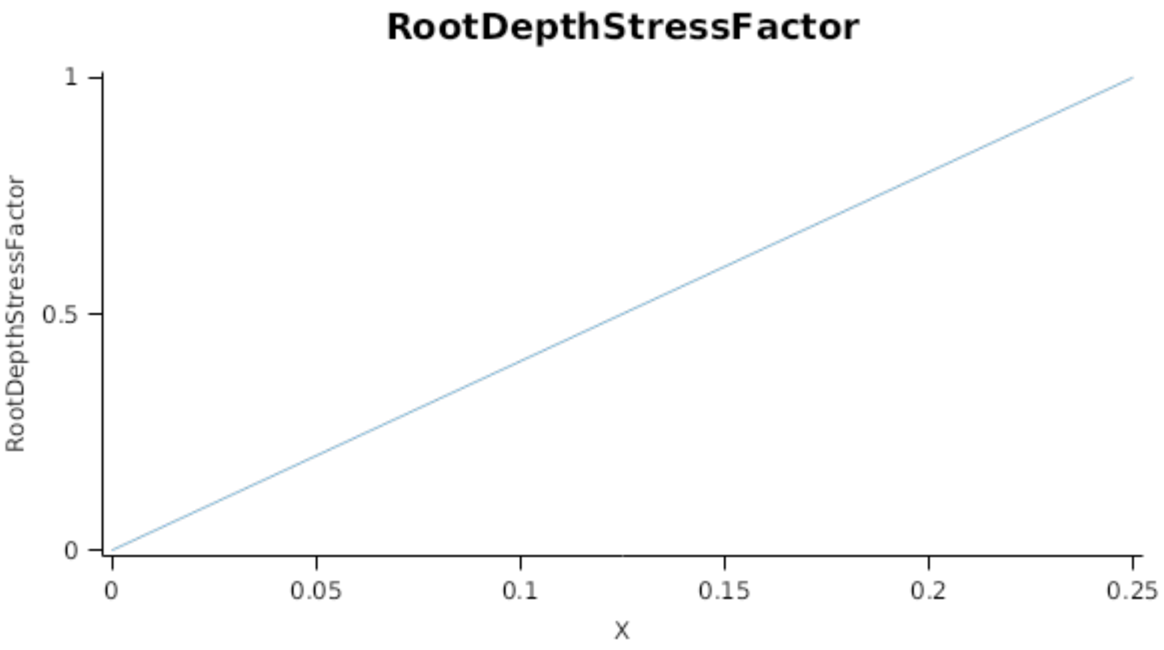
NitrogenRootCalcSwitch has a value between Germination and Maturity calculated as:

Constant = 1

1.5.18 RootDepthStressFactor

RootDepthStressFactor is calculated using linear interpolation

X	RootDepthStressFactor
0.0	0.0
0.2	1.0



1.5.19 SWAvailabilityRatio

1.5.19.1 SWAvailabilityRatio

[DocumentMathFunction /] Returns special values if the numerator is 0 or if the denominator is 0.

Currently used in sorghum/maize code to mimic divide functions in old apsim which return 10 if the denominator is 0 or 0 if the numerator is 0.

1.5.20 DltThermalTime

IF [Phenology].TTFMFromFlowering < NCeaseUptake THEN

ThermalTime = [Phenology].DltTT

ELSE

Zero = 0

1.5.21 ratioRootShootInverse

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

BetweenEmergenceAndEndJuvenile has a value between Emergence and EndJuvenile calculated as:

SubtractFunction = rootFraction - DivideFunction

rootFraction = 1 (0-1)

DivideFunction = inverse / rootFraction

inverse = 1 (0-1)

rootFraction = 2 (0-1)

BetweenEndJuvenileAndFlagLeaf has a value between EndJuvenile and FlagLeaf calculated as:

SubtractFunction = rootFraction - DivideFunction

rootFraction = 1 (0-1)

DivideFunction = inverse / rootFraction

inverse = 1 (0-1)

rootFraction = 1.33 (0-1)

This is a straight lookup table not an lineaer relationship

BetweenFlagLeafAndGrainFill has a value between FlagLeaf and StartGrainFill calculated as:

SubtractFunction = *rootFraction* - *DivideFunction*

rootFraction = 1 (0-1)

DivideFunction = *inverse* / *rootFraction*

inverse = 1 (0-1)

rootFraction = 1.087 (0-1)

1.5.22 CriticalNConc

CriticalNConc = [Root].MinimumNConc

1.5.23 InitialWt

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

Structural = 0.1 (g/plant)

Metabolic = 0

Storage = 0

1.6 Leaf

1.6.1 Leaf

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

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

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

Initial DM mass = 0.1 gm^{-2}

1.6.2 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.

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

Structural = *DMDemandFunction* x *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.

BetweenEmergenceAndFlag has a value between Emergence and FlagLeaf calculated as:

MinimumFunction = Min(*MultiplyFunction*, *dltDmLeafMax*)

Where:

MultiplyFunction = [Leaf].LeafPartitioningCoefficient x [Leaf].DMSupplyFunction *dltDmLeafMax* = [Leaf].DltStressedLAI / *MultiplyFunction*

MultiplyFunction = [Leaf].SlaMin x *smm2sm*

smm2sm = 1E-06

StructuralFraction = 1 (0-1)

Metabolic = 0

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

StorageFraction = 1 - [Leaf].DMDemands.Structural.StructuralFraction

1.6.3 Nitrogen Demand

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

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

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

BetweenEmergenceAndFlower has a value between Emergence and Flowering calculated as:

NewLeafNDemand = [Leaf].DltLAI x [Leaf].NewLeafSLN

Metabolic = Max(*SubtractFunction*, *Zero*)

Where:

SubtractFunction = *nRequired* - [Leaf].Live.N

nRequired = *laiToday* x [Leaf].TargetSLN

laiToday = lai + dltLai - dltSlai

laiToday = *AddFunction* - [Leaf].DltSenescedLai

AddFunction = [Leaf].LAI + [Leaf].DltLAI

Zero = 0

Storage = 0

Minimum N Concentration = 0

Critical N Concentraion = 0

Maximum N Concentration = 0

1.6.4 Dry Matter Supply

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

1.6.5 DM Retranslocation Factor

Leaf does not retranslocate non-structural DM.

1.6.6 Photosynthesis

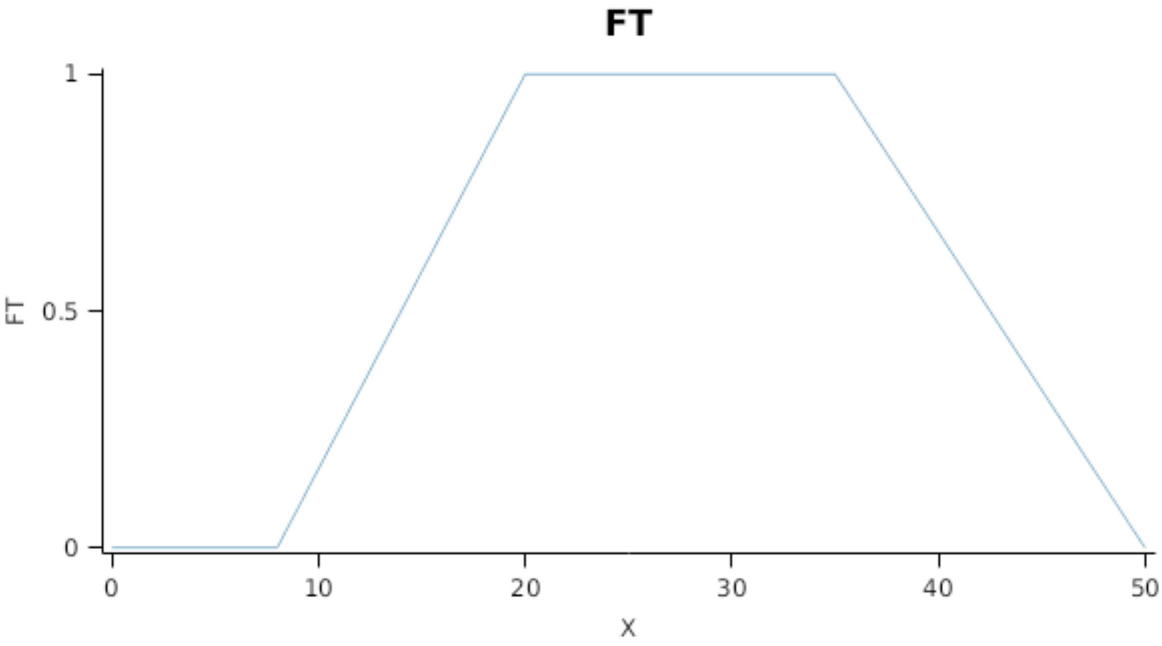
Biomass fixation is modelled as the product of intercepted radiation and its conversion efficiency, the radiation use efficiency (RUE) (Monteith et al., 1977).
This approach simulates net photosynthesis rather than providing separate estimates of growth and respiration.
The potential photosynthesis calculated using RUE is then adjusted according to stress factors, these account for plant nutrition (FN), air temperature (FT), vapour pressure deficit (FVPD), water supply (FW) and atmospheric CO_2 concentration (FCO2).
NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

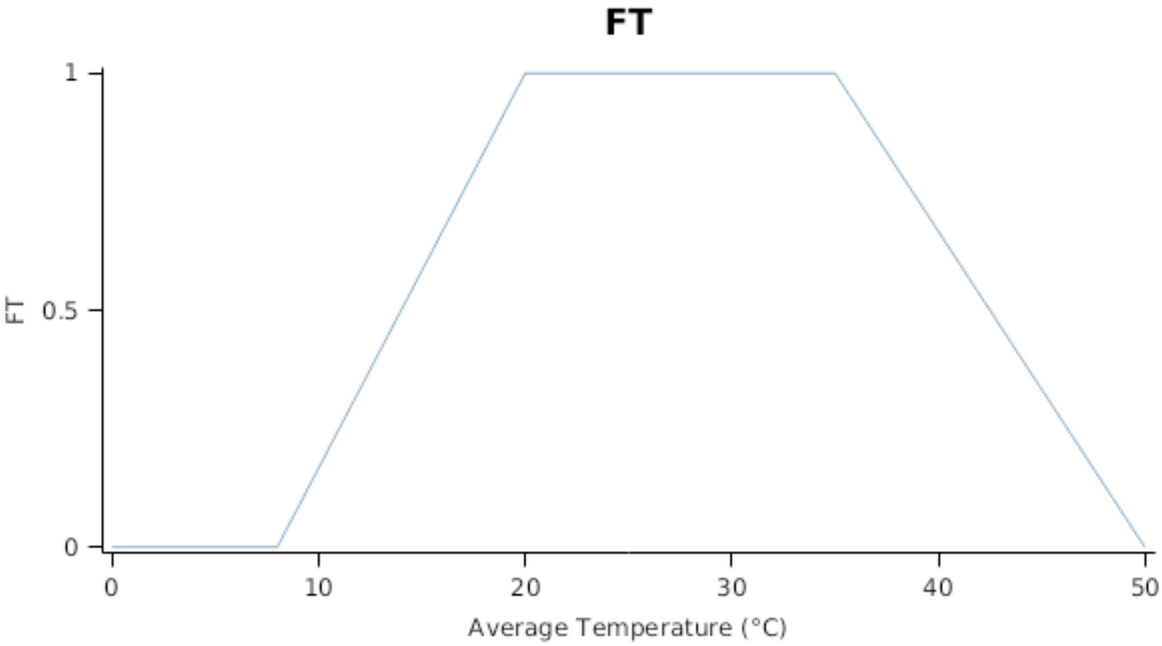
RUE = 1.25

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
0.0	0.0
8.0	0.0
20.0	1.0
35.0	1.0
50.0	0.0





FN = [Leaf].NitrogenPhotoStress

RadnInt = [Leaf].RadiationIntercepted

FVPD = 1

FW = 1

This model calculates the CO_2 impact on RUE using the approach of Reyenga et al., 1999.

For C3 plants,

$F_{CO2} = (CO_2 - CP) \times (350 + 2 \times CP)/(CO_2 + 2 \times CP) \times (350 - CP)$

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

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

For C4 plants,

$F_{CO2} = 0.000143 * CO_2 + 0.95$

When pathway is set to "C4", this model modifier will be have the same as the `double Plant::rue_cow_modifier()` function in the APSIM Classic sorghum model.

1.6.7 Nitrogen Supply

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

1.6.8 Nitrogen Retranslocation Factor

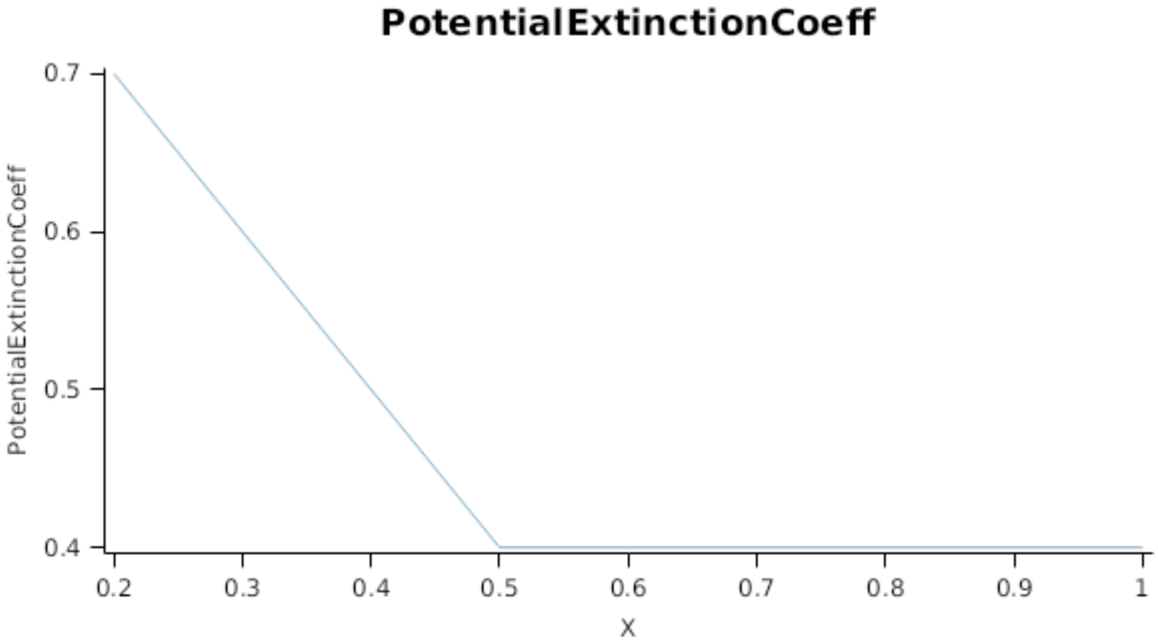
Leaf does not retranslocate non-structural N.

1.6.9 Canopy Properties

ExtinctionCoefficientFunction = *PotentialExtinctionCoeff*

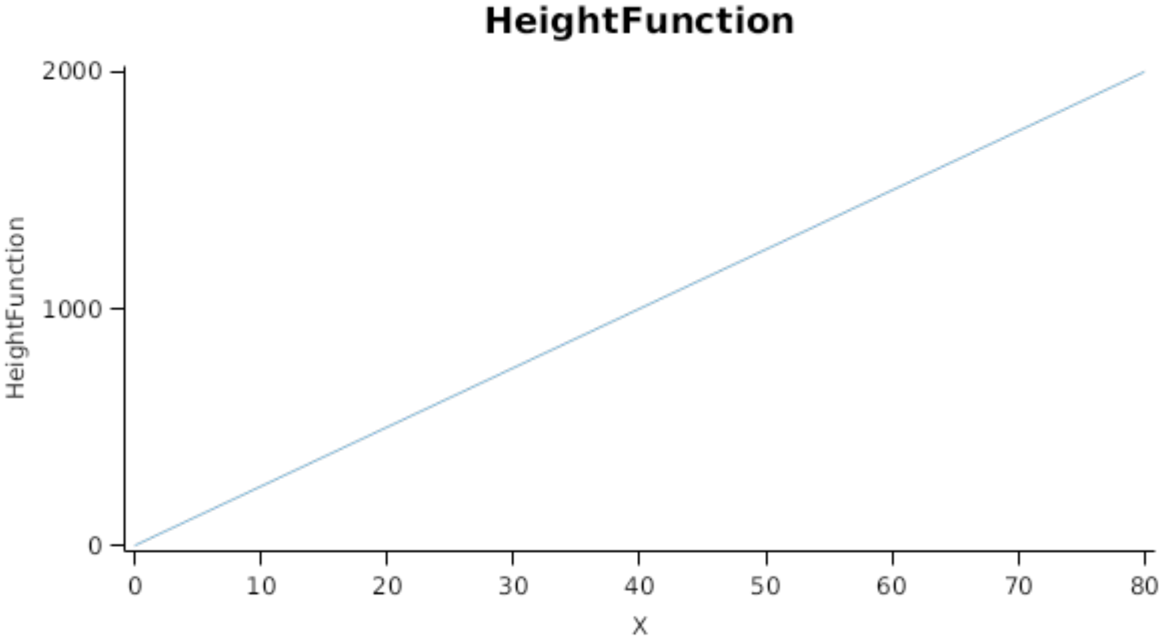
PotentialExtinctionCoeff is calculated using linear interpolation

X	PotentialExtinctionCoeff
0.2	0.7
0.5	0.4
1.0	0.4



HeightFunction is calculated using linear interpolation

X	HeightFunction
0.0	0.0
80.0	2000.0



1.6.10 Senescence and Detachment

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

Leaf 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	30	0
Cut	80	0	0	0
Prune	0	0	60	0
Graze	60	0	10	0

1.7 Rachis

1.7.1 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.

1.7.2 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* x *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.

BetweenFIAndFlower has a value between FloralInitiation and Flowering calculated as:

MultiplyFunction = [Leaf].DMRemaining x [Stem].StemToFlowerFrac

BetweenFlagAndFlowering has a value between FlagLeaf and Flowering calculated as:

MultiplyFunction = [Arbitrator].DM.TotalSupplyFunction x [Stem].StemToFlowerFrac

StructuralFraction = 1

Metabolic = 0

Storage = 0

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 = Max(*zero*, *SubtractFunction*)

Where:

zero = 0

SubtractFunction = * required* - [Rachis].Live.N

* required* = [Rachis].minimumNconc x *AddFunction* x [Rachis].nitrogenDemandSwitch

AddFunction = [Rachis].potentialDMAllocation.Structural + [Rachis].Live.Wt

Metabolic = Max(*zero*, *SubtractFunction*)

Where:

zero = 0

SubtractFunction = * required* - [Rachis].Live.N

* required* = [Rachis].criticalNConc x *AddFunction* x [Rachis].nitrogenDemandSwitch

AddFunction = [Rachis].potentialDMAllocation.Structural + [Rachis].Live.Wt

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

Storage = [Rachis].maximumNconc × ([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

1.7.4 N Concentration Thresholds

MinimumNConc = 0.003

CriticalNConc = 0.01

MaximumNConc = 0.008

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

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

Rachis does not retranslocate non-structural DM.

1.7.6 Nitrogen Supply

Rachis 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 = *NRetranslocationAmount*

NRetranslocationAmount = Max(*MinimumFunction*, *Zero*)

Where:

MinimumFunction = Min(*availableDilnN*, *SubtractFunction*)

Where:

availableDilnN = *DivideFunction* x *DMToday*

DivideFunction = *dltStemNConc* / 100

dltStemNConc = *AddFunction* x [Phenology].ThermalTime

AddFunction = *MultiplyFunction* + *dilnNInt*

MultiplyFunction = *dilnNSlope* x *stemNConc*

dilnNSlope = 0.0062

stemNConc = *divide* x *Percentage*

divide = [Rachis].Live.N / *DMToday*

DMToday = [Rachis].potentialDMAllocation.Structural + [Rachis].Live.Wt

Percentage = 100

dilnNInt = -0.001

DMToday = [Rachis].potentialDMAllocation.Structural + [Rachis].Live.Wt

SubtractFunction = [Rachis].Live.N - [Rachis].NDemands.Structural

Zero = 0

1.7.7 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

1.8 Stem

1.8.1 Stem

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

1.8.2 Dry Matter Demand

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

1.8.2.1 DMDemands

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

Structural = *DMDemandFunction* x *StructuralFraction*

DMDemandFunction = Max(*DMDemandFunction*, *Zero*)

Where:

DMDemandFunction = [Leaf].DMSupplyFunction - [Leaf].DMDemand.Structural - [Rachis].DMDemand.Structural - [Grain].DMDemand.Structural

Zero = 0

StructuralFraction = 1

Metabolic = 0

Storage = 0

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 = Max(*zero*, *SubtractFunction*)

Where:

zero = 0

SubtractFunction = * required* - [Stem].Live.N

* required* = [Stem].CriticalNConc x *AddFunction* x [Stem].nitrogenDemandSwitch

AddFunction = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

Metabolic = Max(*zero*, *SubtractFunction*)

Where:

zero = 0

SubtractFunction = * required* - [Stem].Live.N

* required* = [Stem].MinimumNConc x *AddFunction* x [Stem].nitrogenDemandSwitch

AddFunction = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

Storage = 0

QStructuralPriority = 1

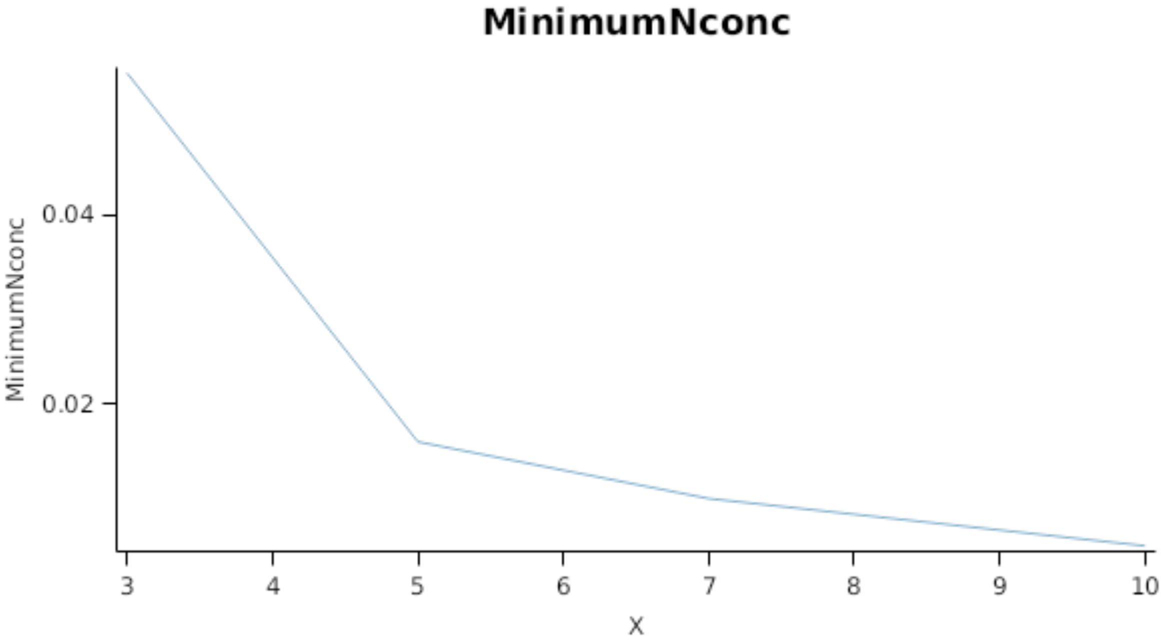
QMetabolicPriority = 1

QStoragePriority = 1

1.8.4 N Concentration Thresholds

MinimumNconc is calculated using linear interpolation

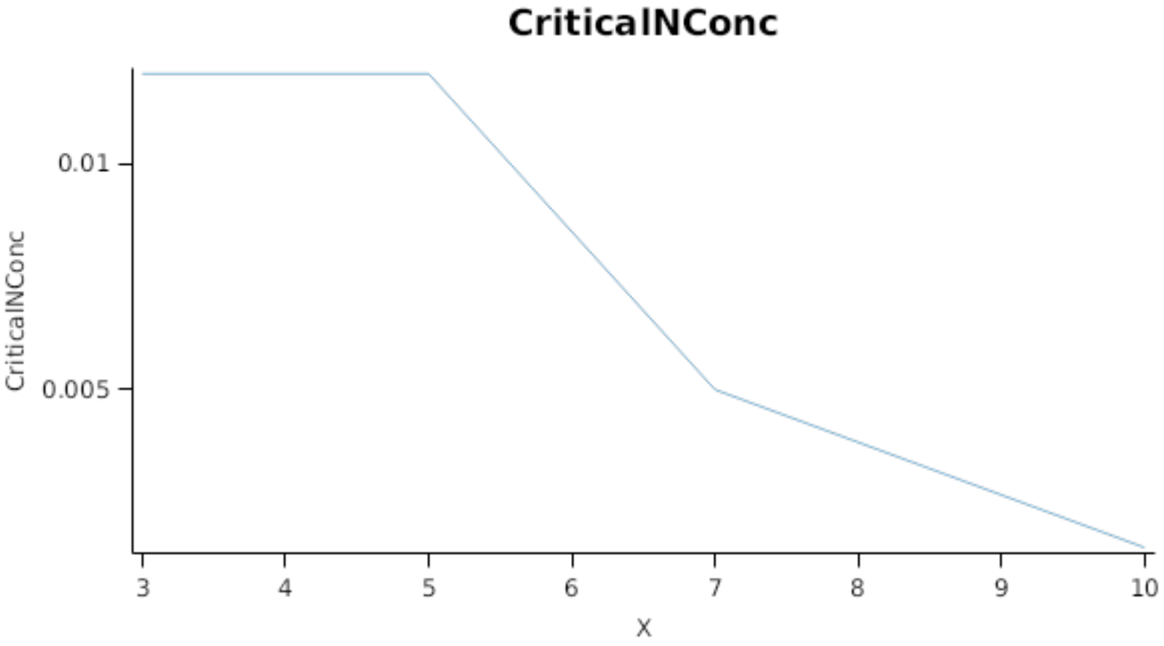
X	MinimumNconc
3.0	0.1
5.0	0.0
7.0	0.0
10.0	0.0



CriticalNConc is calculated using linear interpolation

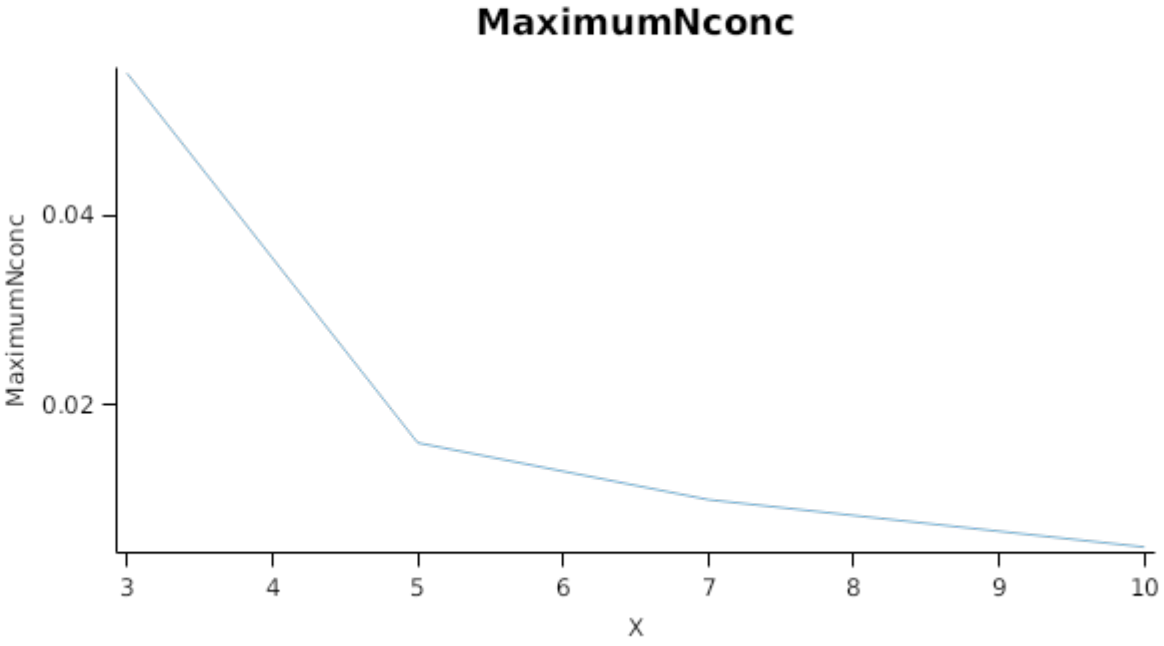
X	CriticalNConc
3.0	0.0
5.0	0.0

X	CriticalNConc
7.0	0.0
10.0	0.0



MaximumNconc is calculated using linear interpolation

X	MaximumNconc
3.0	0.1
5.0	0.0
7.0	0.0
10.0	0.0



1.8.5 Dry Matter Supply

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

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

DMRetranslocationFactor = *DMRetranslocationAmount* / [Stem].Live.Wt

DMRetranslocationAmount = Max(*StemWtAvail*, *Zero*)

Where:

StemWtAvail = *SubtractFunction* x *retransRate*

SubtractFunction = *DMToday* - *MultiplyFunction*

DMToday = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

MultiplyFunction = *PlantMinDM* x [Plant].SowingData.Population

Before Flowering

PreEventValue = 0

On Flowering the value is set to:

PostEventValue = *dmPlantStem* x *SubtractFunction*

dmPlantStem = [Stem].Live.Wt / [Plant].SowingData.Population

SubtractFunction = 1 - *translocFrac*

translocFrac = 0.2

retransRate = 0.15

Zero = 0

1.8.6 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 = *NRetranslocationAmount* / [Stem].Live.Wt

NRetranslocationAmount = Max(*MinimumFunction*, *Zero*)

Where:

MinimumFunction = Min(*availableDilnN*, *SubtractFunction*)

Where:

availableDilnN = *DivideFunction* x *DMToday*

DivideFunction = *dltStemNConc* / 100

dltStemNConc = *AddFunction* x [Phenology].ThermalTime

AddFunction = *MultiplyFunction* + *dilnNInt*

MultiplyFunction = *dilnNSlope* x *stemNConcPct*

dilnNSlope = 0.0062

stemNConcPct = *divide* x *Percentage*

divide = [Stem].Live.N / *DMToday*

DMToday = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

Percentage = 100

dilnNInt = -0.001

DMToday = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

SubtractFunction = [Stem].Live.N - *MultiplyFunction*

MultiplyFunction = *DMToday* x [Stem].MinimumNconc

DMToday = [Stem].potentialDMAllocation.Structural + [Stem].Live.Wt

Zero = 0

1.8.7 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

1.9 TotalPlantDemand

TotalPlantDemand = [Leaf].DMDemands.Structural.DMDemandFunction + [Stem].DMDemands.Structural.DMDemandFunction + [Rachis].DMDemands.Structural.DMDemandFunction + [Grain].DMDemands.Structural.DMDemandFunction

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
* Rachis
* Grain

1.11 AboveGroundLive

1.11.1 AboveGroundLive

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

AboveGroundLive summarises the following biomass objects:

* Leaf
* Stem
* Grain
* Rachis

1.12 AboveGroundDead

1.12.1 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
* Grain
* Rachis

1.13 BelowGround

1.13.1 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

1.14 Total

1.14.1 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
* Rachis
* Grain

1.15 TotalLive

1.15.1 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
* Rachis
* Grain

1.16 TotalDead

1.16.1 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
* Rachis
* Grain

1.17 Cultivars

1.17.1 Australia

1.17.1.1 Buster

Buster overrides the following properties:

[LeafCulms].aX0.FixedValue = 0.786

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aMaxIntercept.FixedValue = -321.13

1.17.1.2 early

early overrides the following properties:

[Phenology].TTEndJuvToInit.FixedValue = 114

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.3 medium

medium overrides the following properties:

[Phenology].TTEndJuvToInit.FixedValue = 157

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.4 late

late overrides the following properties:

[Phenology].TTEndJuvToInit.FixedValue = 201

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.5 ATX623xRTX430

ATX623xRTX430 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 26

[Phenology].TTEndJuvToInit.FixedValue = 202

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 450

1.17.1.6 A35xQL36

A35xQL36 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 11.8

[Phenology].PhotoModifier.XYPairs.X[2] = 14

[Phenology].PhotoModifier.XYPairs.Y[2] = 87.34

[Phenology].TTEndJuvToInit.FixedValue = 134.4

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 450

1.17.1.7 QL41xQL36

QL41xQL36 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 25

[Phenology].TTEndJuvToInit.FixedValue = 179

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 450

1.17.1.8 QL39xQL36

QL39xQL36 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 11.8

[Phenology].PhotoModifier.XYPairs.X[2] = 14

[Phenology].PhotoModifier.XYPairs.Y[2] = 87.34

[Phenology].TTEndJuvToInit.FixedValue = 114.4

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.9 M35-1

M35-1 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 122

[Phenology].TTEndJuvToInit.FixedValue = 252

[Grain].DMPerSeed.FixedValue = 0.001427844

[Leaf].PartitionRate.FixedValue = 0.018

[Arbitrator].NitrogenUptakeMethod.NUptakeCease.FixedValue = 10

[LeafCulms].leafInitRate.FixedValue = 26.1

[Leaf].TargetSLN.PreEventValue.FixedValue = 1.2

[LeafCulms].aMaxSlope.FixedValue = 8.9

[LeafCulms].aMaxIntercept.FixedValue = 245.3

1.17.1.10 CSH13R

CSH13R overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 34

[Phenology].TTEndJuvToInit.FixedValue = 228

[Grain].DMPerSeed.FixedValue = 0.0015

[Leaf].Photosynthesis.RUE.FixedValue = 1.6

[Leaf].PartitionRate.FixedValue = 0.018

[Leaf].TargetSLN.PreEventValue.FixedValue = 1.2

[LeafCulms].leafInitRate.FixedValue = 23.2

1.17.1.11 dekalb_DK55

dekalb_DK55 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 11

[Phenology].PhotoModifier.XYPairs.X[2] = 13

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 205

[Phenology].FlagLeafToFlowering.Target.FixedValue = 100

[Phenology].FloweringToGrainFilling.Target.FixedValue = 30

[Phenology].TTFlowerToMaturity.FixedValue = 695

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.12 texas_RS610

texas_RS610 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 12.3

[Phenology].PhotoModifier.XYPairs.X[2] = 14.6

[Phenology].PhotoModifier.XYPairs.Y[2] = 88.32

[Phenology].TTEndJuvToInit.FixedValue = 120

[Phenology].FlagLeafToFlowering.Target.FixedValue = 100

[Phenology].FloweringToGrainFilling.Target.FixedValue = 30

[Phenology].TTFlowerToMaturity.FixedValue = 695

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.13 pioneer_s34

pioneer_s34 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 12.3

[Phenology].PhotoModifier.XYPairs.X[2] = 14.6

[Phenology].PhotoModifier.XYPairs.Y[2] = 57.5

[Phenology].TTEndJuvToInit.FixedValue = 115

[Phenology].FlagLeafToFlowering.Target.FixedValue = 100

[Phenology].FloweringToGrainFilling.Target.FixedValue = 30

[Phenology].TTFlowerToMaturity.FixedValue = 695

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.14 texas_671

texas_671 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.X[1] = 12.3

[Phenology].PhotoModifier.XYPairs.X[2] = 14.6

[Phenology].PhotoModifier.XYPairs.Y[2] = 34.96

[Phenology].TTEndJuvToInit.FixedValue = 159

[Phenology].FlagLeafToFlowering.Target.FixedValue = 100

[Phenology].FloweringToGrainFilling.Target.FixedValue = 30

[Phenology].TTFlowerToMaturity.FixedValue = 695

[LeafCulms].aMaxIntercept.FixedValue = -321.13

[LeafCulms].aMaxSlope.FixedValue = 46.312

[LeafCulms].aX0.FixedValue = 0.786

1.17.1.15 Scorpio

Scorpio overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 130

[Phenology].FlagLeafToFlowering.Target.FixedValue = 120

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.00065

[LeafCulms].aX0.FixedValue = 0.83

[LeafCulms].aMaxSlope.FixedValue = 2.3

[LeafCulms].aMaxIntercept.FixedValue = 514

[Leaf].Photosynthesis.RUE.FixedValue = 1.75

1.17.1.16 Apollo

Apollo overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 185

[Phenology].FlagLeafToFlowering.Target.FixedValue = 130

[Phenology].TTFlowerToMaturity.FixedValue = 810

[LeafCulms].aX0.FixedValue = 0.71

[LeafCulms].aMaxSlope.FixedValue = 37

[LeafCulms].aMaxIntercept.FixedValue = -101

[Leaf].Photosynthesis.RUE.FixedValue = 1.75

1.17.1.17 Bazley

Bazley overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 185

[Phenology].FlagLeafToFlowering.Target.FixedValue = 130

[Phenology].TTFlowerToMaturity.FixedValue = 810

[LeafCulms].aX0.FixedValue = 0.71

[LeafCulms].aMaxSlope.FixedValue = 37

[LeafCulms].aMaxIntercept.FixedValue = -101

[Leaf].Photosynthesis.RUE.FixedValue = 1.75

1.17.1.18 Taurus

Taurus overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 15

[Phenology].TTEndJuvToInit.FixedValue = 118

[Phenology].FlagLeafToFlowering.Target.FixedValue = 150

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.00065

[LeafCulms].aX0.FixedValue = 0.8

[LeafCulms].aMaxSlope.FixedValue = 1

[LeafCulms].aMaxIntercept.FixedValue = 451

1.17.1.19 P85G33

P85G33 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 118

[Phenology].FlagLeafToFlowering.Target.FixedValue = 150

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0006

[LeafCulms].aX0.FixedValue = 0.86

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 542

1.17.1.20 P84G22

P84G22 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].FlagLeafToFlowering.Target.FixedValue = 140

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0006

[LeafCulms].aX0.FixedValue = 0.81

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 578

1.17.1.21 P84G99

P84G99 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 135

[Phenology].FlagLeafToFlowering.Target.FixedValue = 140

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0005

[LeafCulms].aX0.FixedValue = 0.84

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 495

1.17.1.22 P86G56

P86G56 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 20

[Phenology].TTEndJuvToInit.FixedValue = 138

[Phenology].FlagLeafToFlowering.Target.FixedValue = 150

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0005

[LeafCulms].aX0.FixedValue = 0.84

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 495

1.17.1.23 MR43

MR43 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 15

[Phenology].TTEndJuvToInit.FixedValue = 118

[Phenology].FlagLeafToFlowering.Target.FixedValue = 150

[Phenology].TTFlowerToMaturity.FixedValue = 810

[LeafCulms].aX0.FixedValue = 0.71

[LeafCulms].aMaxSlope.FixedValue = 37

[LeafCulms].aMaxIntercept.FixedValue = -101

1.17.1.24 N_14NUS01

N_14NUS01 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 180

[Phenology].FlagLeafToFlowering.Target.FixedValue = 140

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0007

[LeafCulms].aX0.FixedValue = 0.82

[LeafCulms].aMaxSlope.FixedValue = 8.5

[LeafCulms].aMaxIntercept.FixedValue = 378

1.17.1.25 N_14NUS02

N_14NUS02 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 20

[Phenology].TTEndJuvToInit.FixedValue = 174

[Phenology].FlagLeafToFlowering.Target.FixedValue = 150

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.0006

[LeafCulms].aX0.FixedValue = 0.82

[LeafCulms].aMaxSlope.FixedValue = 8.5

[LeafCulms].aMaxIntercept.FixedValue = 378

1.17.1.26 N_14NUS03

N_14NUS03 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 15

[Phenology].TTEndJuvToInit.FixedValue = 110

[Phenology].FlagLeafToFlowering.Target.FixedValue = 100

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.00045

[LeafCulms].aX0.FixedValue = 0.84

[LeafCulms].aMaxSlope.FixedValue = 9.3

[LeafCulms].aMaxIntercept.FixedValue = 384

1.17.1.27 N_14NUS04

N_14NUS04 overrides the following properties:

[Phenology].PhotoModifier.XYPairs.Y[2] = 0

[Phenology].TTEndJuvToInit.FixedValue = 120

[Phenology].FlagLeafToFlowering.Target.FixedValue = 120

[Phenology].TTFlowerToMaturity.FixedValue = 810

[Grain].DMPerSeed.FixedValue = 0.00045

[LeafCulms].aX0.FixedValue = 0.81

[LeafCulms].aMaxSlope.FixedValue = 0

[LeafCulms].aMaxIntercept.FixedValue = 445

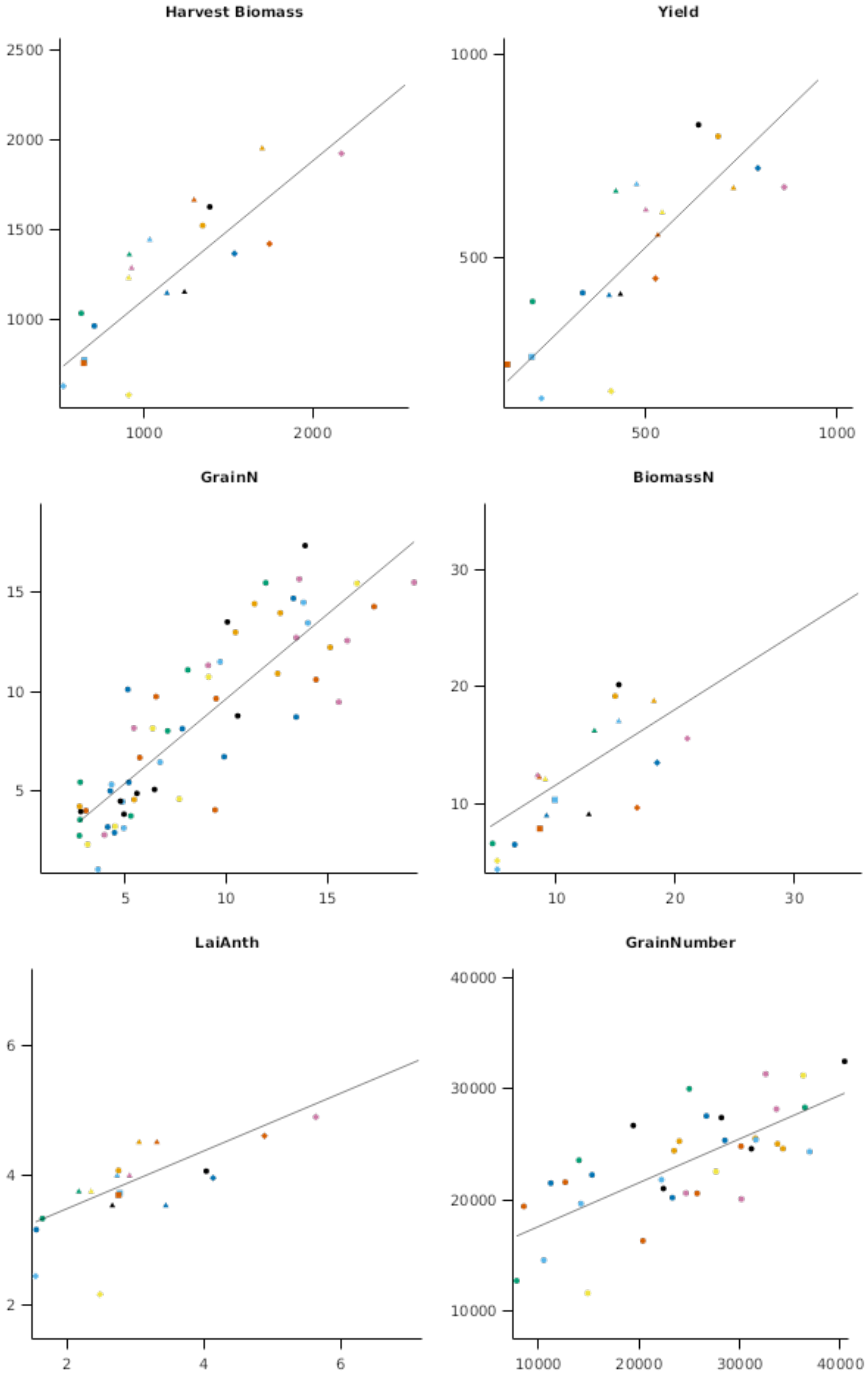
1.18 MortalityRate

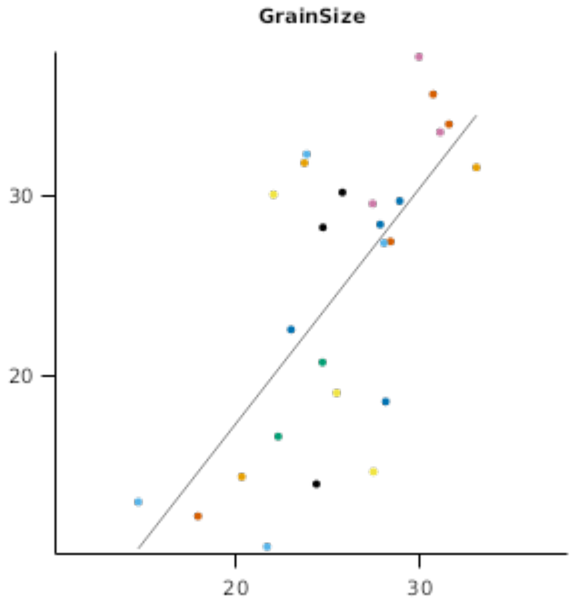
MortalityRate = 0

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).


Harvest Biomass
2500
2000
1500
1000
1000
2000
Yield
1000
500
500
1000
GrainN
15
10
5
5
10
15
BiomassN
30
20
10
10
20
30
LaiAnth
6
4
2
2
4
6
GrainNumber
40000
30000
20000
10000
10000
20000
30000
40000



2.2 Hermitage

List of experiments.

Experiment Name	Design (Number of Treatments)
HE1	Cultivar (3)
HE2	Cultivar (3)
HE3	Cultivar (3)
HE4	Cultivar (3)
HE5	Cultivar (3)
HE6	Cultivar (3)
HE7	Cultivar (3)
HE8	Cultivar (3)

2.2.1 HE1

Experiment: HE1

Description: HE1 - High N/Irrig. - 3 Indian/Aust. Cultivars

Start Date: 1/12/1996

End Date: 30/06/1997

SimulationName	Cultivar
Sorghum_HE1_T1	QL41xQL36
Sorghum_HE1_T2	Buster
Sorghum_HE1_T3	M35-1

2.2.2 HE2

Experiment: HE2

Description: HE2 - Low N/Irrig. - 3 Indian/Aust. Cultivars

Start Date: 1/12/1996

End Date: 30/06/1997

SimulationName	Cultivar
Sorghum_HE1_T1	QL41xQL36
Sorghum_HE1_T2	Buster
Sorghum_HE1_T3	M35-1

2.2.3 HE3

Experiment: HE3

Description: HE3 - High N/No Irrig. - 3 Indian/Aust. Cultivars

Start Date: 1/12/1996

End Date: 30/06/1997

SimulationName	Cultivar
Sorghum_HE3_T1	QL41xQL36
Sorghum_HE3_T2	Buster
Sorghum_HE3_T3	M35-1

2.2.4 HE4

Experiment: HE4

Description: HE4 - Low N/No Irrig. - 3 Indian/Aust. Cultivars

Start Date: 1/12/1996

End Date: 30/06/1997

SimulationName	Cultivar
Sorghum_HE4_T1	QL41xQL36
Sorghum_HE4_T2	Buster
Sorghum_HE4_T3	M35-1

2.2.5 HE5

Experiment: HE5

Description: HE5 - High N/Irrig. - 3 Indian/Aust. Cultivars

Start Date: 18/11/1997

End Date: 15/04/1998

SimulationName	Cultivar
Sorghum_HE5_T1	Buster
Sorghum_HE5_T2	M35-1
Sorghum_HE5_T3	CSH13R

2.2.6 HE6

Experiment: HE6

Description: HE6 - Low N/Irrig. - 3 Indian/Aust. Cultivars

Start Date: 18/11/1997

End Date: 15/04/1998

SimulationName	Cultivar
Sorghum_HE6_T1	Buster
Sorghum_HE6_T2	M35-1
Sorghum_HE6_T3	CSH13R

2.2.7 HE7

Experiment: HE7

Description: HE7 - High N/No Irrig. - 3 Indian/Aust. Cultivars

Start Date: 18/11/1997

End Date: 15/04/1998

SimulationName	Cultivar
Sorghum_HE7_T1	Buster
Sorghum_HE7_T2	M35-1
Sorghum_HE7_T3	CSH13R

2.2.8 HE8

Experiment: HE8

Description: HE8 - Low N/No Irrig - 3 Indian/Aust. Cultivars

Start Date: 17/11/1997

End Date: 15/04/1998

SimulationName	Cultivar
Sorghum_HE8_T1	Buster
Sorghum_HE8_T2	M35-1
Sorghum_HE8_T3	CSH13R

2.3 Icrisat

List of experiments.

Experiment Name	Design (Number of Treatments)
BW5	Cv x Fert x Irrig (12)
BW8	Cv x Fert x Irrig (12)

2.3.1 BW5

Experiment: BW5_GxE

Description: Genotype x Management Environment (Rabi Sorghum)

Start Date: 26/09/1996

End Date: 20/02/1997

Treatment	Cultivar	N Rates	Irrigation
Sorghum_BW5_GxE_T1	M35-1	240	Irrigated

Treatment	Cultivar	N Rates	Irrigation
Sorghum_BW5_GxE_T2	CSH13R	240	Irrigated
Sorghum_BW5_GxE_T3	ATx623/RTx430	240	Irrigated
Sorghum_BW5_GxE_T4	QL41/QL36	240	Irrigated
Sorghum_BW5_GxE_T5	M35-1	120	Dryland
Sorghum_BW5_GxE_T6	CSH13R	120	Dryland
Sorghum_BW5_GxE_T7	ATx623/RTx430	120	Dryland
Sorghum_BW5_GxE_T8	QL41/QL36	120	Dryland
Sorghum_BW5_GxE_T9	M35-1	10	Dryland
Sorghum_BW5_GxE_T10	CSH13R	10	Dryland
Sorghum_BW5_GxE_T11	ATx623/RTx430	10	Dryland
Sorghum_BW5_GxE_T12	QL41/QL36	10	Dryland

2.3.2 BW8

Experiment: BW8_GxE

Description: BW8 - GENO x ENVIRON (RABI SORGHUM)

Start Date: 26/09/1996

End Date: 20/02/1997

Treatment	Cultivar	N Rates	Irrigation
Sorghum_BW8_GxE_T1	M35-1	240	Irrigated
Sorghum_BW8_GxE_T2	CSH13R	240	Irrigated
Sorghum_BW8_GxE_T3	ATx623/RTx430	240	Irrigated
Sorghum_BW8_GxE_T4	QL41/QL36	240	Irrigated
Sorghum_BW8_GxE_T5	M35-1	120	Dryland
Sorghum_BW8_GxE_T6	CSH13R	120	Dryland
Sorghum_BW8_GxE_T7	ATx623/RTx430	120	Dryland
Sorghum_BW8_GxE_T8	QL41/QL36	120	Dryland
Sorghum_BW8_GxE_T9	M35-1	20	Dryland
Sorghum_BW8_GxE_T10	CSH13R	20	Dryland
Sorghum_BW8_GxE_T11	ATx623/RTx430	20	Dryland
Sorghum_BW8_GxE_T12	QL41/QL36	20	Dryland

2.4 Lawes

List of experiments.

Experiment Name	Design (Number of Treatments)
LE13	(6)
LE14	Cv (3)

Experiment Name	Design (Number of Treatments)
LE15	Cv (3)
LE17	Cv (3)
LE19	(8)
LE21	Cv x Fert (9)

2.4.1 LE13

Experiment: LE13

Description: 2 Cultivars x 2 Water regimes x 2 N Potential Yield

Start Date: 28/11/1995

End Date: 1/05/1996

SimulationName	Irrigation	N Rates	Cultivar
Lawes1995FertLowirrigOffCvBuster	Irrigation to establish & fert inco	10 N	Buster
Lawes1995FertHighIrrigOffCvBuster	Irrigation to establish & fert inco	240 N	Buster
Lawes1995FertHighIrrigOnCvBuster	Irrigated	240 N	Buster
Lawes1995FertLowIrrigOffCvM351	Irrigation to establish & fert inco	10 N	M35-1
Lawes1995FertHighIrrigOffCvM351	Irrigation to establish & fert inco	240 N	M35-1
Lawes1995FertHighIrrigOnCvM351	Irrigated	240 N	M35-1

2.4.2 LE14

Experiment: LE14

Description: Early Sow - 3 Indian/Aust Cultivars

Start Date: 5/09/1996

End Date: 19/02/1997

SimulationName	Cultivar
Lawes1996EarlyCvBuster	Buster
Lawes1996EarlyCvM351	M35-1
Lawes1996EarlyCvQL41xQL36	QL41xQL36

2.4.3 LE15

Experiment: LE15

Description: Late Sow - 3 Indian/Aust Cultivars

Start Date: 5/09/1996

End Date: 8/05/1997

SimulationName	Cultivar
Lawes1996LateCvBuster	Buster
Lawes1996LateCvM351	M35-1
Lawes1996LateCvQL41xQL36	QL41xQL36

2.4.4 LE17

Experiment: LE17

Description: LE17 Growth Analysis- Late Sow - 2 Indian/ 1 Aust. Cultivars

Start Date: 27/11/1997

End Date: 27/04/1998

SimulationName	Cultivar
Lawes1997LateCvBuster	Buster
Lawes1997LateCvM351	M35-1
Lawes1997LateCvCSH13R	CSH13R

2.4.5 LE19

Experiment: LE19

Description: LE19 - Genotypic variation in Radiation Use Efficiency

Start Date: 9/11/1998

End Date: 7/03/1999

SimulationName	N Rates	Cultivar
Lawes1998FertOffBuster	0 kg/ha	Buster
Lawes1998FertLowBuster	120 kg/ha	Buster
Lawes1998FertMedBuster	240 kg/ha	Buster
Lawes1998FertHighBuster	360 kg/ha	Buster
Lawes1998FertOffCSH13R	0 kg/ha	CSH13R
Lawes1998FertLowCSH13R	120 kg/ha	CSH13R
Lawes1998FertMedCSH13R	240 kg/ha	CSH13R
Lawes1998FertHighCSH13R	360 kg/ha	CSH13R

2.4.6 LE21

Experiment: Gatton_RUE

Description: N Rates x Genotype LE21

Start Date: 22/11/1999

End Date: 3/04/2000

SimulationName	Cultivar	N Rates
Lawes1999FertLowCvCSH13R	CSH13R	0kg/ha
Lawes1999FertLowCvA35xQL36	A35xQL36	0kg/ha
Lawes1999FertLowCvQL39xQL36	QL39xQL36	0kg/ha
Lawes1999FertMedCvCSH13R	CSH13R	45kg/ha
Lawes1999FertMedCvA35xQL36	A35xQL36	45kg/ha
Lawes1999FertMedCvQL39xQL36	QL39xQL36	45kg/ha
Lawes1999FertHighCvCSH13R	CSH13R	360kg/ha

SimulationName	Cultivar	N Rates
Lawes1999FertHighCvA35xQL36	A35xQL36	360kg/ha
Lawes1999FertHighCvQL39xQL36	QL39xQL36	360kg/ha

3 Sensibility

Need to verify that these sensibility tests are actually sensible. They were mostly adapted from the other validation files.

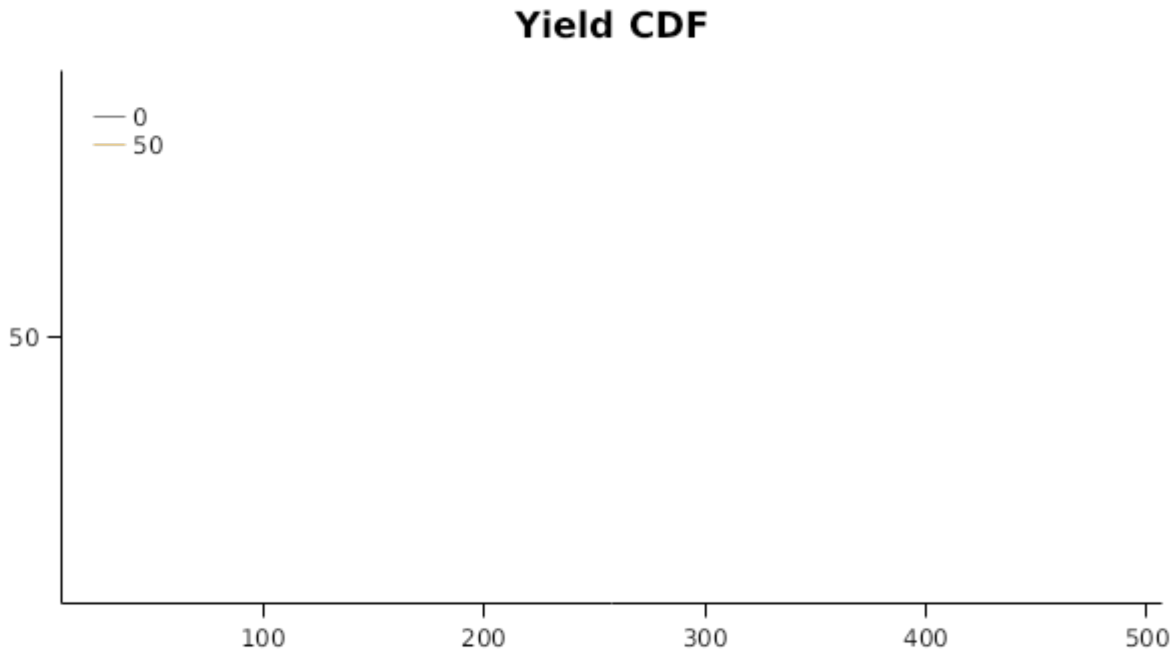
List of experiments.

Experiment Name	Design (Number of Treatments)
Bugesera	NRate (2)
WaterByNFactorial	Irrigation x Nitrogen (10)

3.1 Bugesera

3.1.1 Copied from maize validation set - this needs editing!

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.



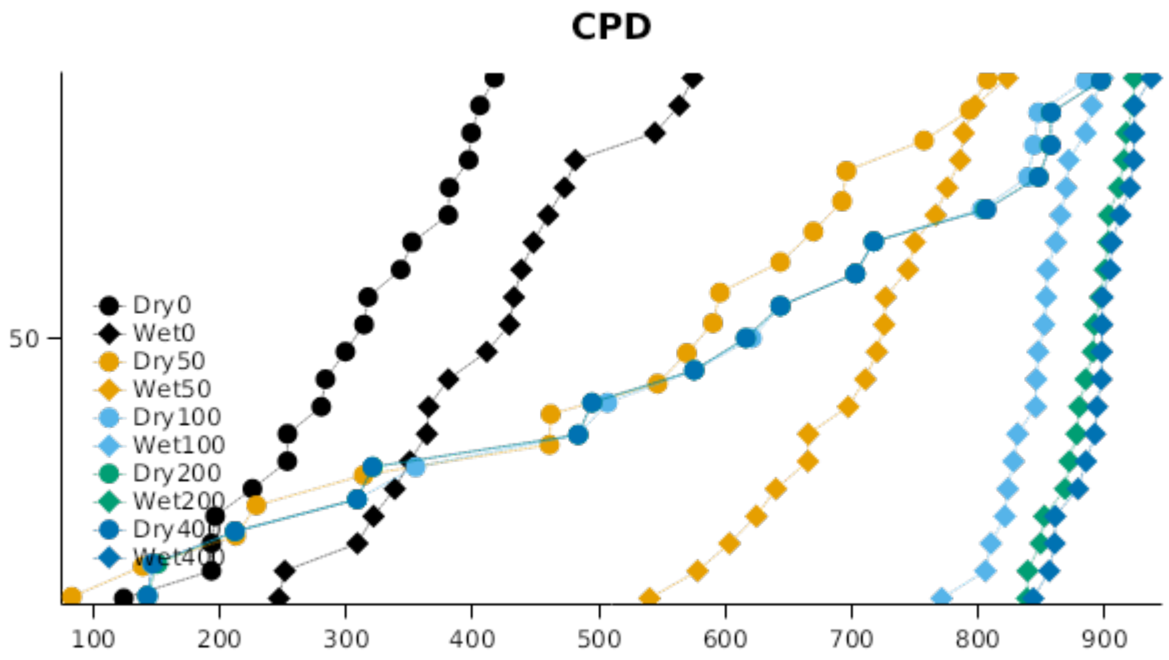
3.2 WaterByNFactorial

3.2.1 Stolen from Barley validation set - this might need edits

This is a hypothetical experiment with Barley sown on the 15th of October every year for 20 years at Lincoln, New Zealand. The treatments applied are as follows:

- Two water treatments; Dry (nil irrigation) and Wet, with irrigation applied when soil water deficit reaches 60% to return water content to 100% of capacity.
- five fertiliser N treatments; 0, 50, 100, 200 and 400 kg N/ha with half of the N applied at sowing and the other half applied at growth stage 32.

The results for irrigted, High N treatments the range of yields are inline with expectations for the location of the simulations. There is no sensitivity to irrigation with zero nitrogen as N supply is the factor limiting production. As N inputs increase the crop becomes increasingly sensitive to water application and the crop is more responsive to nitrogen with irrigation. These results show the model is giving sensible predictions



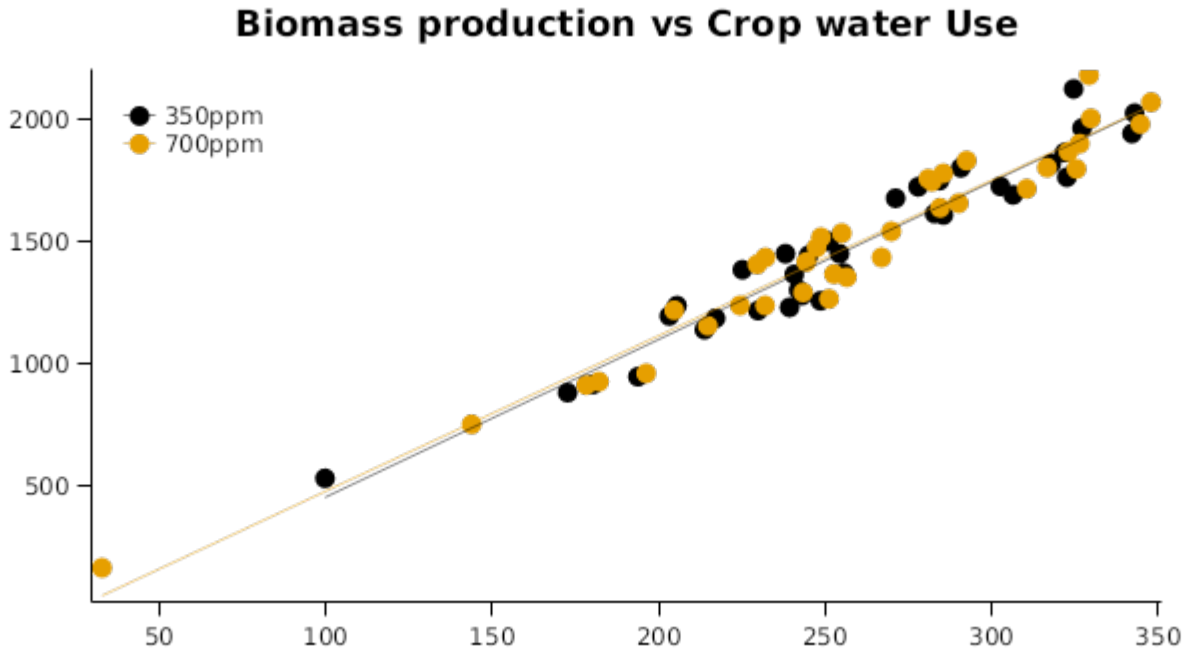
3.3 CO2AndTranspirationEfficiency

List of experiments.

Experiment Name	Design (Number of Treatments)
CO2TE	CO2 (2)

3.3.1 CO2TE

This test examines the impact of a doubling of CO2 from historical (350ppm) on Transpiration Efficiency. Reyenga et al., 1999 suggest an increase of approximately 37% in Transpiration Efficiency over this range in CO2 concentration. In this test, a series of wheat crops are simulated for Dalby, Queensland, Australia. Nitrogen limitation is removed. The slope of plots of biomass production vs crop water use is used to quantify a gross seasonal TE. The change in slope should approximate the response suggested by Reyenga et al., 1999.



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.

Jones, C.A., Kiniry, J.R., Dyke, P.T., 1986. CERES-Maize: a simulation model of maize growth and development..

Monteith, J. L., Moss, C. J., 1977. Climate and the Efficiency of Crop Production in Britain [and Discussion]. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 281 (980), 277-294.

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.