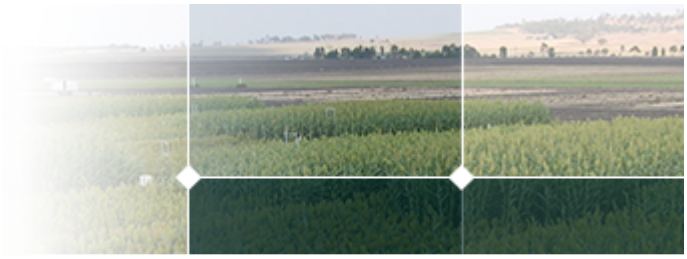


The APSIM Grapevine Model

ZHU, J., Brown, H.E., Parker, A.K.,Trought, M.C.T.

Building the model.

The APSIM Grapevine model has been adapted from the potato model 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 various plant parts
* an arbitrator to allocate resources (N, biomass) to the various plant organs

Pecularities of Grapevine

Grapevine is a challenging crop to model because it is a perennial woody plant. A budding phase was added for simulating the start of the vegetative phase. This budding phase was similar to the emergence phase for annual crop.There are an enormous number of Grapevine cultivars and trainning systems. They differ in phenology and vegetative growth.

Including a Grapevine crop in an APSIM simulation

To indluce a Grapevine crop in a simulation the grapevine plant needs to be added to the vineyard, field or zone in which it is to be grown. This can be done by right clicking on the vineyard, selecting Add from the drop down menu and then selecting Plant from the list that comes up.

Setting plant density and bud number (budNumber number and early canopy growth)

The Grapevine Management script sets the number of stems that will appear from plant.

Setting final leaf number (Crop duration)

> The duration of a Grapevine crop is determined by the number of leaves that the mainstem produces and how long it takes for these to appear and seness.

1 APSIM Description

The Agricultural Production Systems sIMulator (APSIM) is a farming systems modelling framework that is being actively developed by the APSIM Initiative.

It is comprised of

1. a set of biophysical models that capture the science and management of the system being modelled,
2. a software framework that allows these models to be coupled together to facilitate data exchange between the models,
3. a set of input models that capture soil characteristics, climate variables, genotype information, field management etc,
4. a community of developers and users who work together, to share ideas, data and source code,
5. a data platform to enable this sharing and
6. a user interface to make it accessible to a broad range of users.

The literature contains numerous papers outlining the many uses of APSIM applied to diverse problem domains. In particular, Holzworth et al., 2014; Keating et al., 2003; McCown et al., 1996; McCown et al., 1995 have described earlier versions of APSIM in detail, outlining the key APSIM crop and soil process models and presented some examples of the capabilities of APSIM.

Figure 1: This conceptual representation of an APSIM simulation shows a “top level” farm (with climate, farm management and livestock) and two fields. The farm and each field are built from a combination of models found in the toolbox. The APSIM infrastructure connects all selected model pieces together to form a coherent simulation.*

The APSIM Initiative has begun developing a next generation of APSIM (APSIM Next Generation) that is written from scratch and designed to run natively on Windows, LINUX and MAC OSX. The new framework incorporates the best of the APSIM 7.x framework with an improved supporting framework. The Plant Modelling Framework (a generic collection of plant building blocks) was ported from the existing APSIM to bring a rapid development pathway for plant models. The user interface paradigm has been kept the same as the existing APSIM version, but completely rewritten to support new application domains and the newer Plant Modelling Framework. The ability to describe experiments has been added which can also be used for rapidly building factorials of simulations. The ability to write C# scripts to control farm and paddock management has been retained. Finally, all simulation outputs are written to an SQLite database to make it easier and quicker to query, filter and graph outputs.

The model described in this documentation is for APSIM Next Generation.

APSIM is freely available for non-commercial purposes. Non-commercial use of APSIM means public-good research & development and educational activities. It includes the support of policy development and/or implementation by, or on behalf of, government bodies and industry-good work where the research outcomes are to be made publicly available. For more information visit the licensing page on the APSIM web site

2 Model description

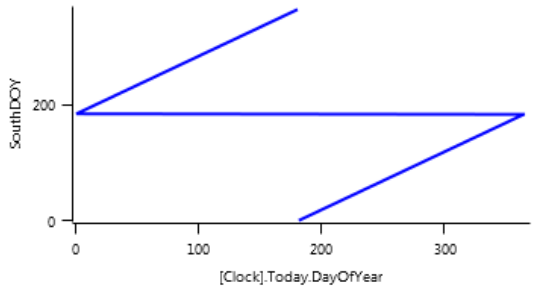
The Grapevine model is constructed from the following list of software components. Details of the exact implementation and parameterisation are provided in the following sections.

List of Plant Model Components.

Component Name	Component Type
SouthDOY	Models.PMF.Functions.LinearInterpolationFunction
Phenology	Models.PMF.Phen.Phenology
Structure	Models.PMF.Struct.Structure
Leaf	Models.PMF.Organs.Leaf
Root	Models.PMF.Organs.Root
Stem	Models.PMF.Organs.GenericOrgan
Cordon	Models.PMF.Organs.GenericOrgan
Trunk	Models.PMF.Organs.GenericOrgan
Berry	Models.PMF.Organs.ReproductiveOrgan
Arbitrator	Models.PMF.OrganArbitrator

2.1 SouthDOY

SouthDOY is calculated as a function of *[Clock].Today.DayOfYear*



X	Y
182	1
365	184
366	184
1	185
181	365

2.2 SauvignonBlanc

Cultivar class for holding cultivar overrides.

2.3 Phenology

This model simulates the development of the crop through successive developmental *phases*. Each phase is bound by distinct growth *stages*. Phases often require a target to be reached to signal movement to the next phase. Differences between cultivars are specified by changing the values of the default parameters shown below.

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

Stage Number	Stage Name	Phase Name
1	StartDormancy	
		EndoDormancy
2	EndDormancy	
		Budding
3	BudBurst	
		Flowering
4	StartFlowering	
		FruitSet
5	FruitSet	
		BerryDevelopment
6	Veraison	
		LeafDeathPhase
7	LeafFall	
		GotoPhase
8	Unused	

2.3.1 Phenological Phases

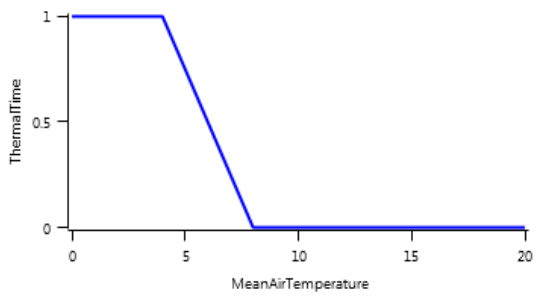
2.3.1.1 EndoDormancy Phase

This phase goes from StartDormancy to EndDormancy.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 13.5 (days)

ThermalTime is calculated from the mean of 3-hourly estimates of air temperature based on daily max and min temperatures.



X	Y
0	1
4	1
8	0
20	0

2.3.1.2 Budding Phase

This phase goes from EndDormancy to BudBurst.

has all the functionality of generic phase, but used to set the emerging date of pereniel crops

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.

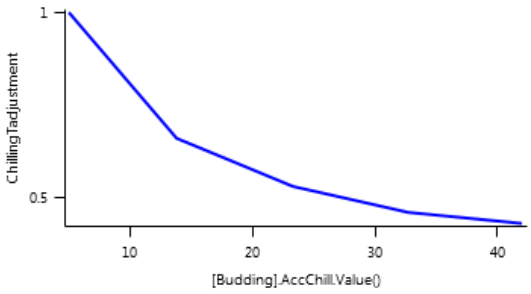
ThermalTime = [Phenology].ThermalTime

Target = Base x ChillingTadjustment

Where:

Base = 51

ChillingTadjustment is calculated as a function of *[Budding].AccChill*



X	Y
5	1
13.8	0.66
23.3	0.53
32.7	0.46
42	0.43

AccChill is a daily accumulation of the values of functions listed below between the StartDormancy and BudBurst stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].EndoDormancy.ThermalTime

2.3.1.3 Flowering Phase

This phase goes from BudBurst to StartFlowering.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 29

ThermalTime = [Phenology].ThermalTime

2.3.1.4 FruitSet Phase

This phase goes from StartFlowering to FruitSet.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 7 (d)

ThermalTime = [Phenology].ThermalTime

2.3.1.5 BerryDevelopment Phase

This phase goes from FruitSet to Veraison.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 20

ThermalTime = [Phenology].ThermalTime

2.3.1.6 LeafDeathPhase Phase

This phase goes from Veraison to LeafFall.

It proceeds until the last leaf on the main-stem has fully senessced. Therefore its duration depends on the number of main-stem leaves that are produced and the rate at which they seness following final leaf appearance.

ThermalTime = [Phenology].ThermalTime

2.3.1.7 GotoPhase Phase

This phase goes from LeafFall to Unused.

A special phase that jumps to another phase.

2.3.2 ThermalTime

ThermalTime =

2.3.3 BudBurstDOY

A function is used to provide flowering date as days after sowing(DAS).

Before BudBurst

PreEventValue = 0

On BudBurst the value is set to:

PostEventValue = [Clock].Today.DayOfYear

2.3.4 FloweringDOY

A function is used to provide flowering date as days after sowing(DAS).

Before StartFlowering

PreEventValue = 0

On StartFlowering the value is set to:

PostEventValue = [Clock].Today.DayOfYear

2.3.5 VeraisonDOY

A function is used to provide flowering date as days after sowing(DAS).

Before Veraison

PreEventValue = 0

On Veraison the value is set to:

PostEventValue = [Clock].Today.DayOfYear

2.3.6 CurrentSeason

A function is used to capture the year of the current season as in the south hemisphere the growing season extends to the second year.

Before BudBurst

PreEventValue = 0

On BudBurst the value is set to:

PostEventValue = [Clock].Today.Year

2.4 Structure

The structure model simulates morphological development of the plant to inform the Leaf class when and how many leaves appear and to provides a hight estimate for use in calculating potential transpiration.

2.4.1 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 and the main-stem population (*MainStemPopn*) for the crop is calculated as: *MainStemPopn* = *Plant.Population* x *PrimaryBudNumber* Primary bud number is > 1 for crops like potato and grape vine where there are more than one main-stem per plant

2.4.2 Main-Stem leaf appearance

Each day the number of main-stem leaf tips appeared (*LeafTipsAppeared*) is calculated as: *LeafTipsAppeared* += *DeltaTips* Where *DeltaTips* is calculated as: *DeltaTips* = *ThermalTime/Phyllochron* Where *Phyllochron* is the thermal time duration between the appearance of leaf tipx given by:

2.4.2.1 Phyllochron

0 between BudBurst and Veraison and a value of zero outside of this period

and *ThermalTime* is given by:

2.4.2.2 ThermalTime

ThermalTime = RefTem x TempRes

Where:

RefTem = 25

TempRes =

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

2.4.2.3 FinalLeafNumber

FinalLeafNumber = 24 (number)

2.4.3 Branching and Branch Mortality

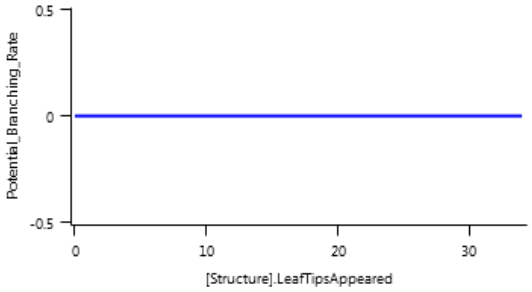
The total population of stems (*TotalStemPopn*) is calculated as: *TotalStemPopn* = *MainStemPopn* + *NewBranches* - *NewlyDeadBranches* Where *NewBranches* = *MainStemPopn* x *BranchingRate* and *BranchingRate* is given by:

2.4.3.1 BranchingRate

BranchingRate = Potential_Branching_Rate x LinearInterpolationFunction

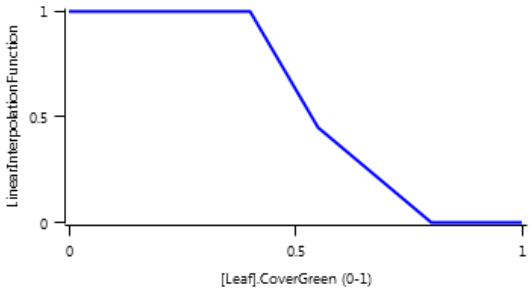
Where:

Potential_Branching_Rate is calculated as a function of *[Structure].LeafTipsAppeared*



X	Y
0	0
7	0
8	0
13	0
14	0
34	0

LinearInterpolationFunction is calculated as a function of *[Leaf].CoverGreen*



X	Y
0	1
0.4	1
0.55	0.45
0.8	0
1	0

NewlyDeadBranches is calcualted as: *NewlyDeadBranches* = (*TotalStemPopn* - *MainStemPopn*) x *BranchMortality* where *BranchMortality* is given by:

2.4.3.2 BranchMortality

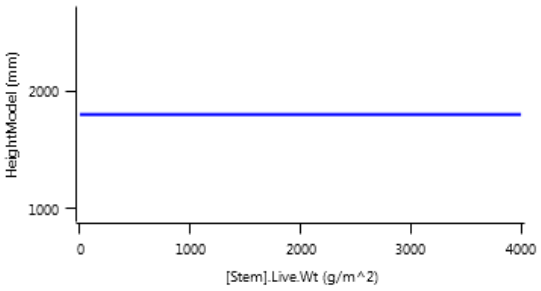
BranchMortality = 0 (/d)

2.4.4 Height

The Height of the crop is calculated by the *HeightModel*:

2.4.4.1 HeightModel

HeightModel is calculated as a function of *[Stem].Live.Wt*



X	Y
0	1800
200	1800
1200	1800
4000	1800

2.4.5 MainStemPrimordiaInitiationRate

0 between BudBurst and Veraison and a value of zero outside of this period

2.4.6 DroughtInducedBranchMortality

DroughtInducedBranchMortality = 0

2.4.7 StemSenescenceAge

StemSenescenceAge = 0

2.5 Leaf

The leaves are modeled as a set of leaf cohorts and the properties of each of these cohorts are summed to give overall values for the leaf organ. A cohort represents all the leaves of a given main stem node position including all of the branch leaves appearing at the same time as the given main-stem leaf (Lawless et al., 2005). The number of leaves in each cohort is the product of the number of plants per m^2 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.

2.5.1 Dry Matter Fixation

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

2.5.1.1 ExtinctionCoeff

ExtinctionCoeff = Constant

Where:

Constant = 0.5

2.5.1.2 Photosynthesis

Biomass accumulation is modeled 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. RUE is calculated from a potential value which is discounted using stress factors that 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.

2.5.1.2.1 RUE

The value of RUE from BudBurst to FruitSet is calculated as follows:

Constant = 1.2 (g/MJ)

The value of RUE from FruitSet to Veraison is calculated as follows:

Constant = 1.05 (g/MJ)

The value of RUE from Veraison to LeafFall is calculated as follows:

Constant = 1.5 (g/MJ)

RUE has a value of zero for phases not specified above

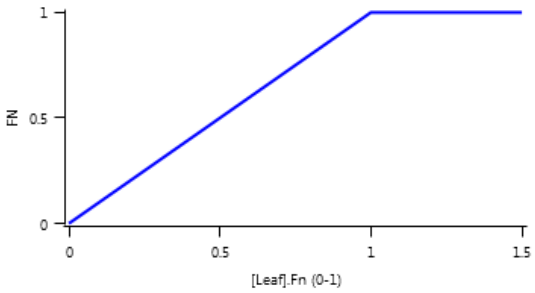
RUE1 = 1 (g/MJ)

2.5.1.2.2 FCO2

This model calculates CO2 Impact on RUE using the approach of Reyenga et al., 1999.

2.5.1.2.3 FN

FN is calculated as a function of *[Leaf].Fn*

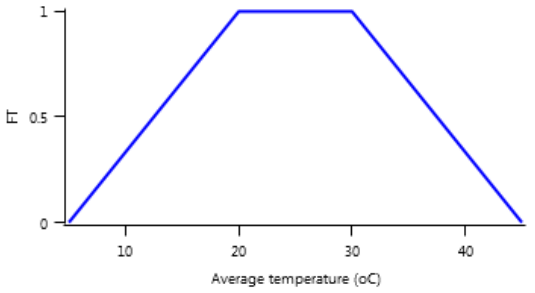


X	Y
0	0
1	1
1.5	1

2.5.1.2.4 FT

FT is calculated as a function of average daily temperature weighted toward max temperature according to the specified MaximumTemperatureWeighting factor.

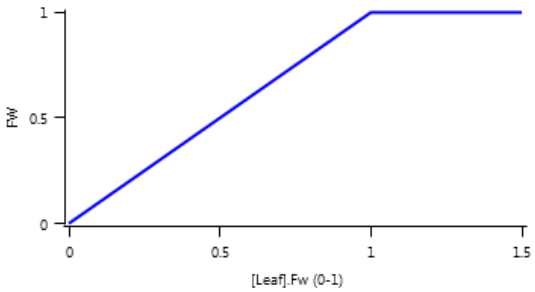
MaximumTemperatureWeighting = 0.75



X	Y
5	0
20	1
30	1
45	0

2.5.1.2.5 FW

FW is calculated as a function of *[Leaf].Fw*

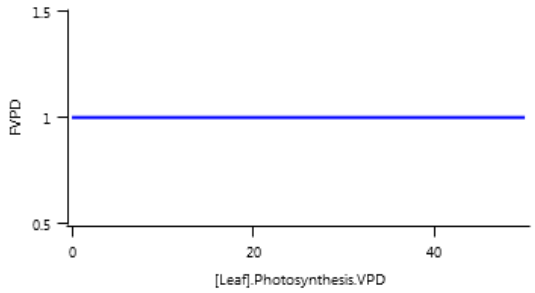


X	Y
0	0
1	1
1.5	1

2.5.1.2.6 FVPD

FVPD is calculated as a function of *[Leaf].Photosynthesis.VPD*

X	Y
0	1
10	1
50	1



RadnInt = [Leaf].RadIntTot

ThermalTime = [Structure].ThermalTime

Area = 1000

Area = 0

Area = 0

2.5.2 CohortParameters

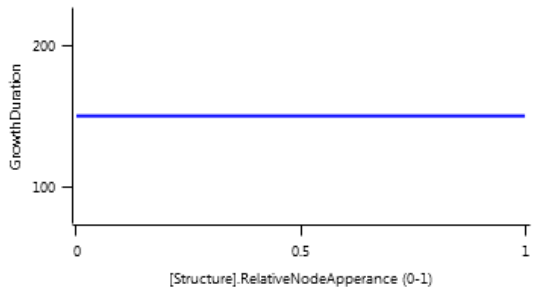
2.5.2.1 Potential Leaf Area index

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 certian number of degree days defined by the *GrowthDuration*

2.5.2.1.1 GrowthDuration

GrowthDuration is calculated as a function of *[Structure].RelativeNodeApperance*

X	Y
0	150
0.3	150
0.4	150
1	150

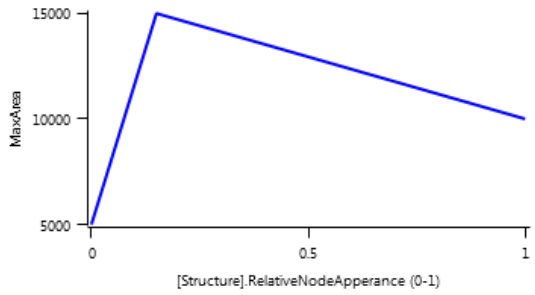


If no stress occurs the leaves will reach a Maximum area (*MaxArea*) at the end of the *GrowthDuration*. The *MaxArea* is defined by:

2.5.2.1.2 MaxArea

MaxArea is calculated as a function of *[Structure].RelativeNodeApperance*

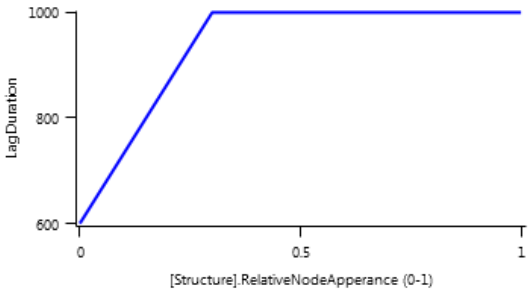
X	Y
0	5000
0.15	15000
1	10000



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*

2.5.2.1.3 LagDuration

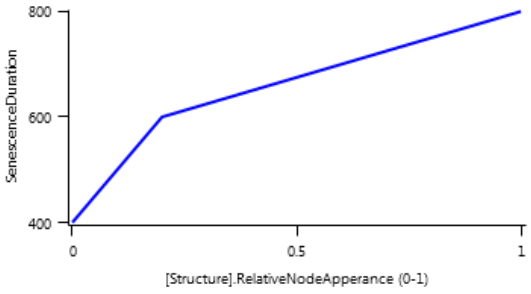
LagDuration is calculated as a function of *[Structure].RelativeNodeApperance*



X	Y
0	600
0.15	800
0.3	1000
0.4	1000
0.73	1000
1	1000

2.5.2.1.4 SenescenceDuration

SenescenceDuration is calculated as a function of *[Structure].RelativeNodeApperance*

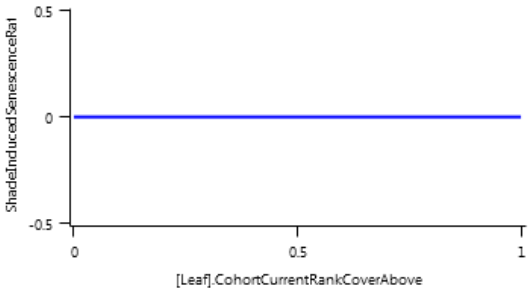


X	Y
0	400
0.2	600
1	800

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:

2.5.2.1.5 ShadeInducedSenescenceRate

ShadeInducedSenescenceRate is calculated as a function of *[Leaf].CohortCurrentRankCoverAbove*



X	Y
0	0
0.5	0
0.8	0
1	0

2.5.2.2 Stress effects on Leaf Area Index

Stress reduces leaf area in a number of ways. Firstly, stress occuring 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:

2.5.2.2.1 CellDivisionStress

CellDivisionStress = 1 (0-1)

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

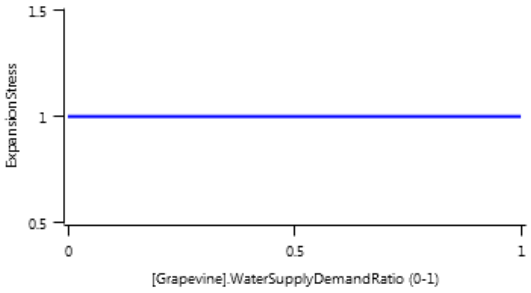
Leaf.FW quantifies water stress and is calculated as *Leaf.Transpiration*/*Leaf.WaterDemand*, where *Leaf.Transpiration* is the minimum of *Leaf.WaterDemand* and *Root.WaterUptake*

Stress during the *GrowthDuration* of the cohort reduces the size increase of the cohort by multiplying the potential increase by a *ExpansionStress* factor:

2.5.2.2.2 ExpansionStress

ExpansionStress is calculated as a function of *[Grapevine].WaterSupplyDemandRatio*

X	Y
0	1
1	1



Stresses can also acellerate 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:

2.5.2.2.3 DroughtInducedLagAcceleration

DroughtInducedLagAcceleration = 1

2.5.2.2.4 DroughtInducedSenAcceleration

DroughtInducedSenAcceleration = 1

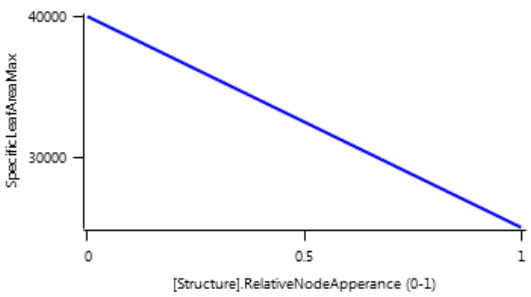
2.5.2.3 Dry matter Demand

Leaf calculates the DM demand from each cohort as a function of the potential size increment (DeltaPotentialArea) an 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*.

2.5.2.3.1 SpecificLeafAreaMax

SpecificLeafAreaMax is calculated as a function of *[Structure].RelativeNodeApperance*

X	Y
0	40000
1	25000



2.5.2.3.2 SpecificLeafAreaMin

SpecificLeafAreaMin = 20000 (mm2/g)

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

2.5.2.3.3 StructuralFraction

StructuralFraction = 0.8 (percentage)

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

Unknown child name: NonStructuralFraction

2.5.2.4 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 = *PotentialStructuralDMAllocation* * *MinimumNConc* where:

2.5.2.4.1 MinimumNConc

MinimumNConc = 0.01 (g/g)

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

2.5.2.4.2 CriticalNConc

CriticalNConc = 0.03 (g/g)

Storage N demand is calculated as the sum of metabolic and structural wt (including todays demands) multiplied by *LuxaryNconc* (*MaximumNConc* - *CriticalNConc*) less the amount of storage N already present. *MaximumNConc* is given by:

2.5.2.4.3 MaximumNConc

MaximumNConc = 0.05 (g/g)

2.5.2.5 Drymatter supply

In additon to photosynthesis, the leaf can also supply DM by reallocation of senescing DM and retranslocation of storgage 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:

2.5.2.5.1 DMReallocationFactor

DMReallocationFactor = 0 (percentage)

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

2.5.2.5.2 DMRetranslocationFactor

DMRetranslocationFactor = 0 (percentage)

2.5.2.6 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:

2.5.2.6.1 NReallocationFactor

NReallocationFactor = 0 (percentage)

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

2.5.2.6.2 NRetranslocationFactor

NRetranslocationFactor = 0 (percentage)

2.5.2.7 DetachmentLagDuration

DetachmentLagDuration = 1000000 (degreeDay)

2.5.2.8 DetachmentDuration

DetachmentDuration = 1000000 (degreeDay)

2.5.2.9 InitialNConc

InitialNConc = 0.04 (g/g)

2.5.2.10 StorageFraction

StorageFraction = 0.2 (percentage)

2.5.2.11 SenessingLeafRelativeSize

SenessingLeafRelativeSize = 1 (0-1)

2.5.2.12 LeafSizeShapeParameter

LeafSizeShapeParameter = 0.01 (parameter controls the logistic growth of the leaf in growth stage)

2.5.2.13 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 0 (g/g)

2.5.2.14 LagDurationAgeMultiplier

LagDurationAgeMultiplier = 1 1 1

2.5.2.15 SenescenceDurationAgeMultiplier

SenescenceDurationAgeMultiplier = 1 1 1

2.5.2.16 LeafSizeAgeMultiplier

LeafSizeAgeMultiplier = 1 1 1 1 1 1 1 1 1 1 1

2.5.2.17 RemobilisationCost

RemobilisationCost = 0

2.5.3 FRGRFunction

FRGRFunction = minimum (RUE_FT, Others)

Where:

RUE_FT = [Leaf].Photosynthesis.FT

Others = minimum (RUE_FN, RUE_FVPD)

Where:

RUE_FN = [Leaf].Photosynthesis.FN

RUE_FVPD = [Leaf].Photosynthesis.FVPD

2.5.4 Total Biomass

This is a composite biomass class. *i.e.* a biomass made up of 1 or more biomass objects.

Total is a composite of the following biomass objects:

- [Leaf].Live
- [Leaf].Dead

2.5.5 CohortArrayLive

This class encapsulates an array of biomass objects

2.5.6 CohortArrayDead

This class encapsulates an array of biomass objects

2.5.7 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	20	0	0	100
Cut	0	0	20	30
Prune	0	0	50	50
Graze	60	0	10	0

2.5.8 StructuralFraction

StructuralFraction = 0.8 (percentage)

2.5.9 FrostFraction

0 between Veraison and LeafFall and a value of zero outside of this period

2.5.10 DMConversionEfficiency

DMConversionEfficiency = 1

2.5.11 ApexStandard

Calculate cohort population using stem population.

2.5.12 RemobilisationCost

RemobilisationCost = 0

2.6 Root

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

Root 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 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'.

Dry Matter Demands

A daily DM demand is provided to the organ abitrator 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.

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 is the respective factors are set to values other then zero.

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 = $NO3_i$ x KNO3 x $NO3_{ppm,\ i}$ x NUptakeSWFactor

NH4 uptake = $NH4_i$ x KNH4 x $NH4_{ppm,\ i}$ x NUptakeSWFactor

Nitrogen uptake demand is limited to the maximum daily potential uptake (MaxDailyNUptake) and the plants N demand. 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.

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) x KL_i x KLModifier

2.6.1 NitrogenDemandSwitch

A value of 1 is returned if phenology is between BudBurst and LeafFall phases, otherwise a value of 0 is returned.

2.6.2 MinimumNConc

MinimumNConc = 0.01 (g/g)

2.6.3 MaximumNConc

MaximumNConc = 0.01 (g/g)

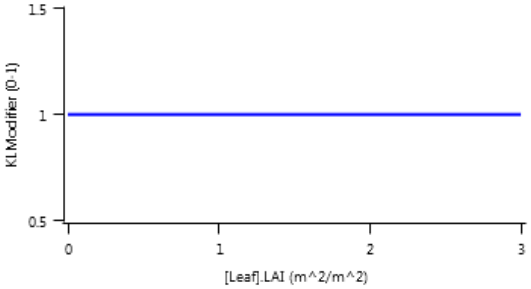
2.6.4 MaximumRootDepth

MaximumRootDepth = 1000000 (mm)

2.6.5 KLModifier

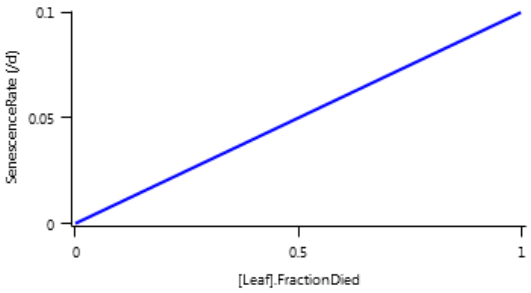
KLModifier is calculated as a function of *[Leaf].LAI*

X	Y
0	1
3	1



2.6.6 SenescenceRate

SenescenceRate is calculated as a function of *[Leaf].FractionDied*



X	Y
0	0
1	0.1

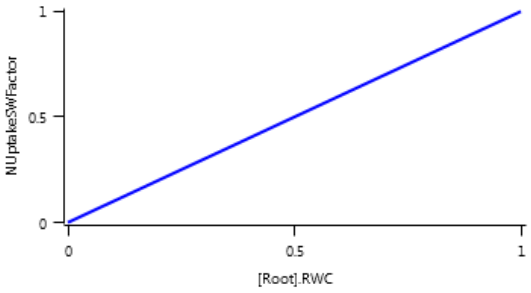
2.6.7 BiomassRemovalDefaults

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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

2.6.8 NUptakeSWFactor

NUptakeSWFactor is calculated as a function of *[Root].RWC*



X	Y
0	0
1	1

2.6.9 InitialDM

InitialDM = 5000 (g/plant)

2.6.10 SpecificRootLength

SpecificRootLength = 40 (m/g)

2.6.11 RootFrontVelocity

RootFrontVelocity = 10 (mm/d)

2.6.12 KNO3

KNO3 = 0.02

2.6.13 KNH4

KNH4 = 0.003

2.6.14 MaxDailyNUptake

MaxDailyNUptake = 6 (kg N/ha)

2.6.15 DMDemandFunction

This is the Partition Fraction Demand Function which returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs

PartitionFraction = 0.2

2.6.16 DMConversionEfficiency

DMConversionEfficiency = 1

2.6.17 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 1 (/d)

2.6.18 RemobilisationCost

RemobilisationCost = 0

2.7 Stem

This organ is simulated using a generic organ type.

2.7.1 Dry Matter Demand

Total Dry matter demand is calculated by the DMDemandFunction

This is the Partition Fraction Demand Function which returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs

PartitionFraction = 0.2

All demand is structural and this organ has no Non-structural demand

2.7.2 Nitrogen Demand

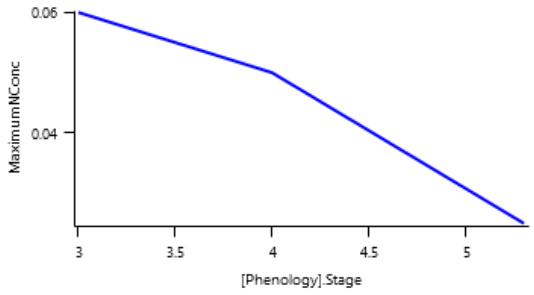
The daily structural N demand is the product of Total DM demand and a Minimum N concentration

MinimumNConc = 0.006 (g/g)

The daily Storage N demand is the product of Total DM demand and a Maximum N concentration

MaximumNConc is calculated as a function of *[Phenology].Stage*

X	Y
3	0.06
4	0.05
5.3	0.025



The demand for N is reduced by a factor specified by the NitrogenDemandFactor

A value of 1 is returned if phenology is between BudBurst and LeafFall phases, otherwise a value of 0 is returned.

2.7.3 Dry Matter Supply

Stem does not reallocate DM when senescence of the organ occurs

Stem does not retranslocate non-structural DM

2.7.4 Nitrogen Supply

Stem does not reallocate N when senescence of the organ occurs

Stem will retranslocate 5% of non-structural N each day

2.7.5 Senescence and Detachment

Stem has senescence parameterised to zero so all biomss in this organ will remain live

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

2.7.6 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

2.8 Cordon

This organ is simulated using a generic organ type.

2.8.1 Dry Matter Demand

Total Dry matter demand is calculated by the DMDemandFunction

This is the Partition Fraction Demand Function which returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs

PartitionFraction = 0.01

All demand is structural and this organ has no Non-structural demand

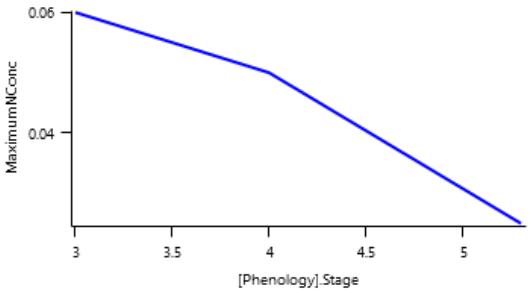
2.8.2 Nitrogen Demand

The daily structural N demand is the product of Total DM demand and a Minimum N concentration

MinimumNConc = 0.006 (g/g)

The daily Storage N demand is the product of Total DM demand and a Maximum N concentration

MaximumNConc is calculated as a function of *[Phenology].Stage*



X	Y
3	0.06
4	0.05
5.3	0.025

The demand for N is reduced by a factor specified by the NitrogenDemandFactor

A value of 1 is returned if phenology is between BudBurst and LeafFall phases, otherwise a value of 0 is returned.

2.8.3 Dry Matter Supply

Cordon does not reallocate DM when senescence of the organ occurs

Cordon does not retranslocate non-structural DM

2.8.4 Nitrogen Supply

Cordon does not reallocate N when senescence of the organ occurs

Cordon will retranslocate 5% of non-structural N each day

2.8.5 Senescence and Detachment

Cordon has senescence parameterised to zero so all biomss in this organ will remain live

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

2.8.6 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

2.9 Trunk

This organ is simulated using a generic organ type.

2.9.1 Dry Matter Demand

Total Dry matter demand is calculated by the DMDemandFunction

This is the Partition Fraction Demand Function which returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs

PartitionFraction = 0.05

All demand is structural and this organ has no Non-structural demand

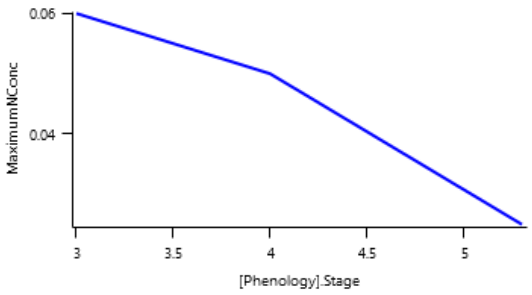
2.9.2 Nitrogen Demand

The daily structural N demand is the product of Total DM demand and a Minimum N concentration

MinimumNConc = 0.006

The daily Storage N demand is the product of Total DM demand and a Maximum N concentration

MaximumNConc is calculated as a function of *[Phenology].Stage*



X	Y
3	0.06
4	0.05
5.3	0.025

The demand for N is reduced by a factor specified by the NitrogenDemandFactor

A value of 1 is returned if phenology is between BudBurst and LeafFall phases, otherwise a value of 0 is returned.

2.9.3 Dry Matter Supply

Trunk does not reallocate DM when senescence of the organ occurs

Trunk does not retranslocate non-structural DM

2.9.4 Nitrogen Supply

Trunk does not reallocate N when senescence of the organ occurs

Trunk will retranslocate 5% of non-structural N each day

2.9.5 Senescence and Detachment

Trunk has senescence parameterised to zero so all biomss in this organ will remain live

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

2.9.6 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

2.10 Berry

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).

2.10.1 MaxNConcDailyGrowth

MaxNConcDailyGrowth = 0.01 (g/g)

2.10.2 NFillingRate

NFillingRate = 0 (g/m2/d)

2.10.3 MinimumNConc

MinimumNConc = 0 (g/g)

2.10.4 MaximumNConc

MaximumNConc = 0.0001 (g/g)

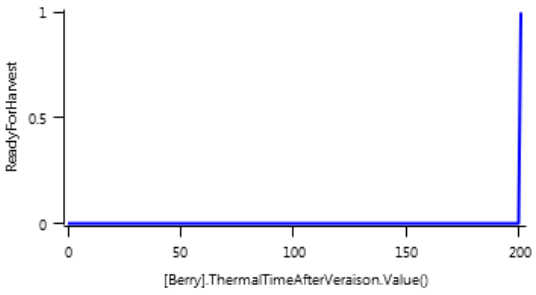
2.10.5 ThermalTimeAfterVeraison

ThermalTimeAfterVeraison is a daily accumulation of the values of functions listed below between the Veraison and LeafFall stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].ThermalTime

2.10.6 ReadyForHarvest

ReadyForHarvest is calculated as a function of *[Berry].ThermalTimeAfterVeraison*



X	Y
0	0
200	0
201	1

2.10.7 ThermalTimeAfterFlowering

ThermalTimeAfterFlowering is a daily accumulation of the values of functions listed below between the FruitSet and LeafFall stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].ThermalTime

2.10.8 SolubleSolidsConc

An exponential function

ThermalTime = [Berry].DaysAfterVeraison

2.10.9 NumberFunction

NumberFunction = [Structure].MainStemPopn x BunchesPerShoot x PotentialBerriesPerBunch

Where:

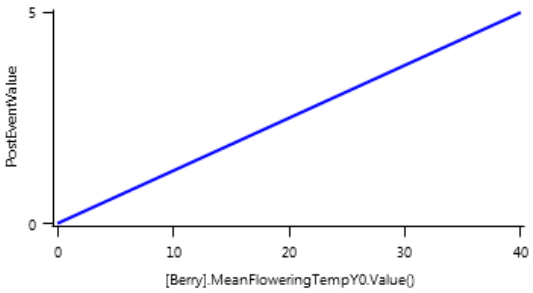
Before BudBurst

PreEventValue = 0

On BudBurst the value is set to:

PostEventValue is calculated as a function of *[Berry].MeanFloweringTempY0*

X	Y
0	0
40	5



PotentialBerriesPerBunch = MinNum + BerriesPerBunch

Where:

MinNum = 66

a sigmoid function of the form y = Xmax * 1 / 1 + $e^{-(XValue - Xo) / b}$

XValue = [Berry].MeanFloweringTemp

Ymax = 44

Xo = 16.3

b = 0.7

2.10.10 MeanFloweringTempY0

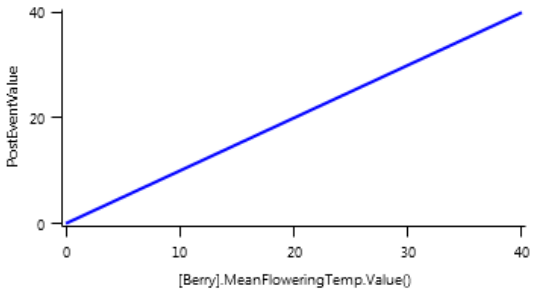
Before FruitSet

PreEventValue = 13

On FruitSet the value is set to:

PostEventValue is calculated as a function of *[Berry].MeanFloweringTemp*

X	Y
0	0
40	40



2.10.11 MeanFloweringTemp

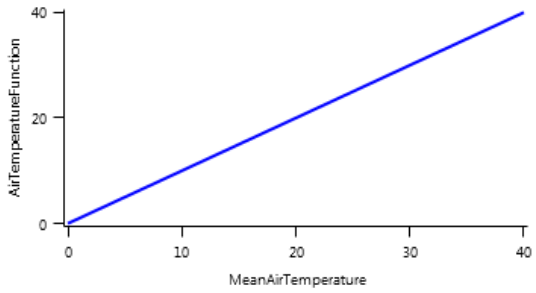
MeanFloweringTemp = Temperature / Time

Where:

Temperature is a daily accumulation of the values of functions listed below between the StartFlowering and FruitSet stages. Function values added to the accumulate total each day are:

AirTemperatureFunction is calculated from the mean of 3-hourly estimates of air temperature based on daily max and min temperatures.

X	Y
0	0
40	40



Time is a daily accumulation of the values of functions listed below between the StartFlowering and FruitSet stages. Function values added to the accumulate total each day are:

Days = 1 (days/day)

2.10.12 DMDemandFunction

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

TT = [Structure].ThermalTime

2.10.13 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

2.10.14 MaximumPotentialGrainSize

MaximumPotentialGrainSize = 2 (g)

2.10.15 WaterContent

WaterContent = Constant - PhaseLookupValue

Where:

Constant = 0.903

0 between Veraison and LeafFall and a value of zero outside of this period

2.10.16 DMConversionEfficiency

DMConversionEfficiency = 1

2.10.17 RemobilisationCost

RemobilisationCost = 0

2.11 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 differnt pools of biomass:

- **Structural biomass** which remains within an organ once it is partitioned there
- **Metabolic biomass** which generally remains within an organ but is able to be re-allocated when the organ senesses and may be re-translocated 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 re-translocation to other organs whenever supply from uptake, fixation and re-allocation is lower than demand .

The process followed for biomass arbitration is shown in Figure 1. Arbitration responds to events broadcast daily by the central APSIM infrastructure:

1. **doPotentialPlantGrowth**. When this event is broadcast 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:
2. **Fixation supply**. From photosynthesis (DM) or symbiotic fixation (N)
3. **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).
4. **Retranslocation supply**. Storage biomass that may be moved from organs to meet demands of other organs.
5. **Reallocation supply**. Biomass that can be moved from senescing organs to meet the demands of other organs.
6. **doPotentialPlantPartitioning.** On this event the Arbitrator first executes the DoDMSetup() method to establish the DM supplies and demands from each organ. 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() establish N supplies and Demands 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 the plants demand for N uptake from the soil (N Uptake Demand).
7. **doNutrientArbitration.** When this event is broadcast by the model framework 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 nuch 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.
8. **doActualPlantPartitioning.** On this event the arbitrator call DoNRetranslocation() and DoNFixation() to satisify 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 conentration and final allocations are sent to organs.

Figure 2: Schematic showing procedure for arbitration of biomass partitioning. Pink boxes are events that are broadcast each day by the model infrastructure and their numbering shows the order of procedure. Blue boxes are methods that are called when these events are broadcast. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.

2.11.1 NArbitrator

Relative allocation rules used to determine partitioning

Arbitration is performed in two passes for each of the supply sources. On the first pass, biomass or nutrient supply is allocated to structural and metabolic pools of each organ based on their demand relative to the demand from all organs. On the second pass any remaining supply is allocated to non-structural pool based on the organ's relative demand.

2.11.2 DMArbitrator

Relative allocation rules used to determine partitioning

Arbitration is performed in two passes for each of the supply sources. On the first pass, biomass or nutrient supply is allocated to structural and metabolic pools of each organ based on their demand relative to the demand from all organs. On the second pass any remaining supply is allocated to non-structural pool based on the organ's relative demand.

3 Replacements

A replacements model

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

List of Plant Model Components.

Component Name	Component Type
SouthDOY	Models.PMF.Functions.LinearInterpolationFunction
Phenology	Models.PMF.Phen.Phenology
Structure	Models.PMF.Struct.Structure
Leaf	Models.PMF.Organs.Leaf
Root	Models.PMF.Organs.Root
Stem	Models.PMF.Organs.GenericOrgan
Cordon	Models.PMF.Organs.GenericOrgan
Trunk	Models.PMF.Organs.GenericOrgan
Berry	Models.PMF.Organs.ReproductiveOrgan
Arbitrator	Models.PMF.OrganArbitrator

3.1 SauvignonBlanc

Cultivar class for holding cultivar overrides.

3.2 Phenology

This model simulates the development of the crop through successive developmental *phases*. Each phase is bound by distinct growth *stages*. Phases often require a target to be reached to signal movement to the next phase. Differences between cultivars are specified by changing the values of the default parameters shown below.

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

Stage Number	Stage Name	Phase Name
1	StartDormancy	
		EndoDormancy
2	EndDormancy	
		Budding
3	BudBurst	
		Flowering
4	StartFlowering	
		FruitSet
5	FruitSet	
		BerryDevelopment
6	Veraison	

Stage Number	Stage Name	Phase Name
		LeafDeathPhase
7	LeafFall	
		GotoPhase
8	Unused	

3.2.1 Phenological Phases

3.2.1.1 EndoDormancy Phase

This phase goes from StartDormancy to EndDormancy.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 13.5 (days)

3.2.1.2 Budding Phase

This phase goes from EndDormancy to BudBurst.

has all the functionality of generic phase, but used to set the emerging date of pereniel crops

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.

ThermalTime = [Phenology].ThermalTime

Target = Base x ChillingTadjustment

Where:

Base = 51

AccChill is a daily accumulation of the values of functions listed below between the StartDormancy and BudBurst stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].EndoDormancy.ThermalTime

3.2.1.3 Flowering Phase

This phase goes from BudBurst to StartFlowering.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 29

ThermalTime = [Phenology].ThermalTime

3.2.1.4 FruitSet Phase

This phase goes from StartFlowering to FruitSet.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 7 (d)

ThermalTime = [Phenology].ThermalTime

3.2.1.5 BerryDevelopment Phase

This phase goes from FruitSet to Veraison.

It uses a *ThermalTime Target* to determine the duration between development *Stages*. *ThermalTime* is accumulated until the *Target* is met and remaining *ThermalTime* is forwarded to the next phase.

Target = 20

ThermalTime = [Phenology].ThermalTime

3.2.1.6 LeafDeathPhase Phase

This phase goes from Veraison to LeafFall.

It proceeds until the last leaf on the main-stem has fully senessced. Therefore its duration depends on the number of main-stem leaves that are produced and the rate at which they seness following final leaf appearance.

ThermalTime = [Phenology].ThermalTime

3.2.1.7 GotoPhase Phase

This phase goes from LeafFall to Unused.

A special phase that jumps to another phase.

3.2.2 ThermalTime

ThermalTime =

3.2.3 BudBurstDOY

A function is used to provide flowering date as days after sowing(DAS).

3.2.4 FloweringDOY

A function is used to provide flowering date as days after sowing(DAS).

3.2.5 VeraisonDOY

A function is used to provide flowering date as days after sowing(DAS).

3.2.6 CurrentSeason

A function is used to capture the year of the current season as in the south hemisphere the growing season extends to the second year.

3.3 Structure

The structure model simulates morphological development of the plant to inform the Leaf class when and how many leaves appear and to provides a hight estimate for use in calculating potential transpiration.

3.3.1 Plant and Main-Stem Population

The *Plant.Population* is set at sowing with information sent from a manager script in the Sow method. The *PrimaryBudNumber* is also sent with the Sow method and the main-stem population (*MainStemPopn*) for the crop is calculated as: *MainStemPopn = Plant.Population* x *PrimaryBudNumber* Primary bud number is > 1 for crops like potato and grape vine where there are more than one main-stem per plant

3.3.2 Main-Stem leaf appearance

Each day the number of main-stem leaf tips appeared (*LeafTipsAppeared*) is calculated as: *LeafTipsAppeared* += *DeltaTips* Where *DeltaTips* is calculated as: *DeltaTips = ThermalTime/Phyllochron* Where *Phyllochron* is the thermal time duration between the appearance of leaf tipx given by:

3.3.2.1 Phyllochron

0 between BudBurst and Veraison and a value of zero outside of this period

and *ThermalTime* is given by:

3.3.2.2 ThermalTime

ThermalTime = RefTem x TempRes

Where:

RefTem = 25

TempRes =

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

3.3.2.3 FinalLeafNumber

FinalLeafNumber = 24 (number)

3.3.3 Branching and Branch Mortality

The total population of stems (*TotalStemPopn*) is calculated as: *TotalStemPopn* = *MainStemPopn* + *NewBranches* - *NewlyDeadBranches* Where *NewBranches* = *MainStemPopn* x *BranchingRate* and *BranchingRate* is given by:

3.3.3.1 BranchingRate

BranchingRate = Potential_Branching_Rate x LinearInterpolationFunction

Where:

NewlyDeadBranches is calcualted as: *NewlyDeadBranches* = (*TotalStemPopn* - *MainStemPopn*) x *BranchMortality* where *BranchMortality* is given by:

3.3.3.2 BranchMortality

BranchMortality = 0 (/d)

3.3.4 Height

The Height of the crop is calculated by the *HeightModel*:

3.3.4.1 HeightModel

3.3.5 MainStemPrimordialnitiationRate

0 between BudBurst and Veraison and a value of zero outside of this period

3.3.6 DroughtInducedBranchMortality

DroughtInducedBranchMortality = 0

3.3.7 StemSenescenceAge

StemSenescenceAge = 0

3.4 Leaf

The leaves are modeled as a set of leaf cohorts and the properties of each of these cohorts are summed to give overall values for the leaf organ. A cohort represents all the leaves of a given main stem node position including all of the branch leaves appearing at the same time as the given main-stem leaf (Lawless et al., 2005). The number of leaves in each cohort is the product of the number of plants per m^2 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.

3.4.1 Dry Matter Fixation

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

3.4.1.1 ExtinctionCoeff

ExtinctionCoeff = Constant

Where:

Constant = 0.5

3.4.1.2 Photosynthesis

Biomass accumulation is modeled 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. RUE is calculated from a potential value which is discounted using stress factors that 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.

3.4.1.2.1 RUE

The value of RUE from BudBurst to FruitSet is calculated as follows:

Constant = 1.2 (g/MJ)

The value of RUE from FruitSet to Veraison is calculated as follows:

Constant = 1.05 (g/MJ)

The value of RUE from Veraison to LeafFall is calculated as follows:

Constant = 1.5 (g/MJ)

RUE has a value of zero for phases not specified above

RUE1 = 1 (g/MJ)

3.4.1.2.2 FCO2

This model calculates CO2 Impact on RUE using the approach of Reyenga et al., 1999.

3.4.1.2.3 FVPD

RadnInt = [Leaf].RadIntTot

ThermalTime = [Structure].ThermalTime

Area = 1000

Area = 0

Area = 0

3.4.2 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 certian number of degree days defined by the *GrowthDuration*

3.4.2.1 GrowthDuration

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

3.4.2.2 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*

3.4.2.3 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:

3.4.2.4 ShadeInducedSenescenceRate

3.4.3 Stress effects on Leaf Area Index

Stress reduces leaf area in a number of ways. Firstly, stress occuring 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:

3.4.3.1 CellDivisionStress

CellDivisionStress = 1 (0-1)

Leaf.FN quantifys the N stress status of the plant and represents the concentration of metabolic N relative the maximum potentil 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 *GrowthDuration* of the cohort reduces the size increase of the cohort by multiplying the potential increase by a *ExpansionStress* factor:

3.4.3.2 ExpansionStress

Stresses can also acellerate 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:

3.4.3.3 DroughtInducedLagAcceleration

DroughtInducedLagAcceleration = 1

3.4.3.4 DroughtInducedSenAcceleration

DroughtInducedSenAcceleration = 1

3.4.4 Dry matter Demand

Leaf calculates the DM demand from each cohort as a function of the potential size increment (DeltaPotentialArea) an 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*.

3.4.4.1 SpecificLeafAreaMin

SpecificLeafAreaMin = 20000 (mm2/g)

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

3.4.4.2 StructuralFraction

StructuralFraction = 0.8 (percentage)

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

Unknown child name: NonStructuralFraction

3.4.5 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 = *PotentialStructuralDMAllocation* * *MinimumNConc* where:

3.4.5.1 MinimumNConc

MinimumNConc = 0.01 (g/g)

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

3.4.5.2 CriticalNConc

CriticalNConc = 0.03 (g/g)

Storage N demand is calculated as the sum of metabolic and structural wt (including todays demands) multiplied by *LuxaryNconc* (*MaximumNConc* - *CriticalNConc*) less the amount of storage N already present. *MaximumNConc* is given by:

3.4.5.3 MaximumNConc

MaximumNConc = 0.05 (g/g)

3.4.6 Drymatter supply

In additon to photosynthesis, the leaf can also supply DM by reallocation of senescing DM and retranslocation of storgage 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:

3.4.6.1 DMReallocationFactor

DMReallocationFactor = 0 (percentage)

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

3.4.6.2 DMRetranslocationFactor

DMRetranslocationFactor = 0 (percentage)

3.4.7 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:

3.4.7.1 NReallocationFactor

NReallocationFactor = 0 (percentage)

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

3.4.7.2 NRetranslocationFactor

NRetranslocationFactor = 0 (percentage)

3.4.7.1 DetachmentLagDuration

DetachmentLagDuration = 1000000 (degreeDay)

3.4.7.2 DetachmentDuration

DetachmentDuration = 1000000 (degreeDay)

3.4.7.3 InitialNConc

InitialNConc = 0.04 (g/g)

3.4.7.4 StorageFraction

StorageFraction = 0.2 (percentage)

3.4.7.5 SenessingLeafRelativeSize

SenessingLeafRelativeSize = 1 (0-1)

3.4.7.6 LeafSizeShapeParameter

LeafSizeShapeParameter = 0.01 (parameter controls the logistic growth of the leaf in growth stage)

3.4.7.7 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 0 (g/g)

3.4.7.8 LagDurationAgeMultiplier

LagDurationAgeMultiplier = 1 1 1

3.4.7.9 SenescenceDurationAgeMultiplier

SenescenceDurationAgeMultiplier = 1 1 1

3.4.7.10 LeafSizeAgeMultiplier

LeafSizeAgeMultiplier = 1 1 1 1 1 1 1 1 1 1 1

3.4.7.11 RemobilisationCost

RemobilisationCost = 0

3.4.8 FRGRFunction

FRGRFunction = minimum (RUE_FT, Others)

Where:

RUE_FT = [Leaf].Photosynthesis.FT

Others = minimum (RUE_FN, RUE_FVPD)

Where:

RUE_FN = [Leaf].Photosynthesis.FN

RUE_FVPD = [Leaf].Photosynthesis.FVPD

3.4.9 Total Biomass

This is a composite biomass class. *i.e.* a biomass made up of 1 or more biomass objects.

Total is a composite of the following biomass objects:

- [Leaf].Live
- [Leaf].Dead

3.4.10 CohortArrayLive

This class encapsulates an array of biomass objects

3.4.11 CohortArrayDead

This class encapsulates an array of biomass objects

3.4.12 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. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

Method	% Live Removed	% Dead Removed	% Live To Residue	% Dead To Residue
Harvest	20	0	0	100
Cut	0	0	20	30
Prune	0	0	50	50
Graze	60	0	10	0

3.4.13 StructuralFraction

StructuralFraction = 0.8 (percentage)

3.4.14 FrostFraction

0 between Veraison and LeafFall and a value of zero outside of this period

3.4.15 DMConversionEfficiency

DMConversionEfficiency = 1

3.4.16 ApexStandard

Calculate cohort population using stem population.

3.4.17 RemobilisationCost

RemobilisationCost = 0

3.5 Root

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

Root 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 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'.

Dry Matter Demands

A daily DM demand is provided to the organ abitrator 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.

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 is the respective factors are set to values other then zero.

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.

$$NO3\ uptake = NO3_i \times KNO3 \times NO3_{ppm,\ i} \times NUptakeSWFactor$$

$$NH4\ uptake = NH4_i \times KNH4 \times NH4_{ppm,\ i} \times NUptakeSWFactor$$

Nitrogen uptake demand is limited to the maximum daily potential uptake (MaxDailyNUptake) and the plants N demand. 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.

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) x KL_i x KLModifier

3.5.1 NitrogenDemandSwitch

A value of 1 is returned if phenology is between BudBurst and LeafFall phases, otherwise a value of 0 is returned.

3.5.2 MinimumNConc

MinimumNConc = 0.01 (g/g)

3.5.3 MaximumNConc

MaximumNConc = 0.01 (g/g)

3.5.4 MaximumRootDepth

MaximumRootDepth = 1000000 (mm)

3.5.5 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

3.5.6 InitialDM

InitialDM = 5000 (g/plant)

3.5.7 SpecificRootLength

SpecificRootLength = 40 (m/g)

3.5.8 RootFrontVelocity

RootFrontVelocity = 10 (mm/d)

3.5.9 KNO3

KNO3 = 0.02

3.5.10 KNH4

KNH4 = 0.003

3.5.11 MaxDailyNUptake

MaxDailyNUptake = 6 (kg N/ha)

3.5.12 DMDemandFunction

This is the Partition Fraction Demand Function which returns the product of its PartitionFraction and the total DM supplied to the arbitrator by all organs

PartitionFraction = 0.2

3.5.13 DMConversionEfficiency

DMConversionEfficiency = 1

3.5.14 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 1 (/d)

3.5.15 RemobilisationCost

RemobilisationCost = 0

3.6 Berry

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).

3.6.1 MaxNConcDailyGrowth

MaxNConcDailyGrowth = 0.01 (g/g)

3.6.2 NFillingRate

NFillingRate = 0 (g/m2/d)

3.6.3 MinimumNConc

MinimumNConc = 0 (g/g)

3.6.4 MaximumNConc

MaximumNConc = 0.0001 (g/g)

3.6.5 ThermalTimeAfterVeraison

ThermalTimeAfterVeraison is a daily accumulation of the values of functions listed below between the Veraison and LeafFall stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].ThermalTime

3.6.6 ThermalTimeAfterFlowering

ThermalTimeAfterFlowering is a daily accumulation of the values of functions listed below between the FruitSet and LeafFall stages. Function values added to the accumulate total each day are:

ThermalTime = [Phenology].ThermalTime

3.6.7 SolubleSolidsConc

An exponential function

ThermalTime = [Berry].DaysAfterVeraison

3.6.8 NumberFunction

NumberFunction = [Structure].MainStemPopn x BunchesPerShoot x PotentialBerriesPerBunch

Where:

PotentialBerriesPerBunch = MinNum + BerriesPerBunch

Where:

MinNum = 66

a sigmoid function of the form y = Xmax * 1 / 1 + $e^{-(XValue - Xo) / b}$

XValue = [Berry].MeanFloweringTemp

Ymax = 44

Xo = 16.3

b = 0.7

3.6.9 MeanFloweringTemp

MeanFloweringTemp = Temperature / Time

Where:

Temperature is a daily accumulation of the values of functions listed below between the StartFlowering and FruitSet stages. Function values added to the accumulate total each day are:

Time is a daily accumulation of the values of functions listed below between the StartFlowering and FruitSet stages. Function values added to the accumulate total each day are:

Days = 1 (days/day)

3.6.10 DMDemandFunction

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

TT = [Structure].ThermalTime

3.6.11 BiomassRemovalDefaults

This organ will respond to certain management actions by either removing some of its biomass from the system or transferring some of its biomass to the soil. The following table describes the proportions of live and dead biomass that are transferred for a range of management actions.

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

3.6.12 MaximumPotentialGrainSize

MaximumPotentialGrainSize = 2 (g)

3.6.13 WaterContent

WaterContent = Constant - PhaseLookupValue

Where:

Constant = 0.903

0 between Veraison and LeafFall and a value of zero outside of this period

3.6.14 DMConversionEfficiency

DMConversionEfficiency = 1

3.6.15 RemobilisationCost

RemobilisationCost = 0

3.7 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 differnt pools of biomass:

- **Structural biomass** which remains within an organ once it is partitioned there
- **Metabolic biomass** which generally remains within an organ but is able to be re-allocated when the organ senesses and may be re-translocated 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 re-translocation to other organs whenever supply from uptake, fixation and re-allocation is lower than demand .

The process followed for biomass arbitration is shown in Figure 1. Arbitration responds to events broadcast daily by the central APSIM infrastructure:

1. **doPotentialPlantGrowth**. When this event is broadcast 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:
2. **Fixation supply**. From photosynthesis (DM) or symbiotic fixation (N)
3. **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).
4. **Retranslocation supply**. Storage biomass that may be moved from organs to meet demands of other organs.
5. **Reallocation supply**. Biomass that can be moved from senescing organs to meet the demands of other organs.
6. **doPotentialPlantPartitioning.** On this event the Arbitrator first executes the DoDMSetup() method to establish the DM supplies and demands from each organ. 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() establish N supplies and Demands 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 the plants demand for N uptake from the soil (N Uptake Demand).
7. **doNutrientArbitration.** When this event is broadcast by the model framework 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 nuch 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.
8. **doActualPlantPartitioning.** On this event the arbitrator call DoNRetranslocation() and DoNFixation() to satisify 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 conentration and final allocations are sent to organs.

Figure 3: Schematic showing procedure for arbitration of biomass partitioning. Pink boxes are events that are broadcast each day by the model infrastructure and their numbering shows the order of procedure. Blue boxes are methods that are called when these events are broadcast. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.

3.7.1 NArbitrator

Relative allocation rules used to determine partitioning

Arbitration is performed in two passes for each of the supply sources. On the first pass, biomass or nutrient supply is allocated to structural and metabolic pools of each organ based on their demand relative to the demand from all organs. On the second pass any remaining supply is allocated to non-structural pool based on the organ's relative demand.

3.7.2 DMArbitrator

Relative allocation rules used to determine partitioning

Arbitration is performed in two passes for each of the supply sources. On the first pass, biomass or nutrient supply is allocated to structural and metabolic pools of each organ based on their demand relative to the demand from all organs. On the second pass any remaining supply is allocated to non-structural pool based on the organ's relative demand.

3.1 Report

A report class for writing output to the data store.

4 DataStore

A storage service for reading and writing to/from a SQLITE database.

4.1 PhenoObs

Reads the contents of a specific sheet from an EXCEL file and stores into the DataStore.

4.2 PhenoPredObs

Reads the contents of a file (in apsim format) and stores into the DataStore. If the file has a column name of 'SimulationName' then this model will only input data for those rows where the data in column 'SimulationName' matches the name of the simulation under which this input model sits. If the file does NOT have a 'SimulationName' column then all data will be input.

4.2.1 PhenoPredObs

Variable	n	Slope	Intercept	R2	RMSE	NSE	ME	MAE
Grapevine.Phenology.BudBurstDOY.Value()	28	0.894	24.558	0.475	7.912	-0.501	-5.036	6.464
Grapevine.Phenology.FloweringDOY.Value()	28	0.723	82.549	0.594	13.955	-3.893	-13.321	13.321
Grapevine.Phenology.VeraisonDOY.Value()	28	-4.950	280.944	0.126	112.956	-205.017	-0.607	70.250

5 BaseSim

A simulation model

5.1 Vineyard

A generic system that can have children

5.1.1 Soil

The soil class encapsulates a soil characterisation and 0 or more soil samples. the methods in this class that return double[] always return using the "Standard layer structure" i.e. the layer structure as defined by the Water child object. method. Mapping will occur to achieve this if necessary. To obtain the "raw", unmapped, values use the child classes e.g. SoilWater, Analysis and Sample.

5.1.1.1 Water

A model for capturing water parameters

5.1.1.1.1 GrapevineSoil

A soil crop parameterization class.

5.1.1.2 SoilWater

.NET port of the Fortran SoilWat model Ported by Shaun Verrall Mar 2011 Extended by Eric Zurcher Mar 2012

5.1.1.3 SoilNitrogen

Computes the soil C and N processes

5.1.1.4 SoilOrganicMatter

A model for capturing soil organic matter properties

5.1.1.5 Analysis

This class captures data from a soil analysis

5.1.1.6 Initial water

Represents the simulation initial water status. There are multiple ways of specifying the starting water; 1) by a fraction of a full profile, 2) by depth of wet soil or 3) a single value of plant available water.

5.1.1.7 Sample

The class represents a soil sample.

5.1.1.8 CERESSoilTemperature

Calculates the average soil temperature at the centre of each layer, based on the soil temperature model of EPIC (Williams et al 1984) This code was separated from old SoilN - tidied up but not updated (RCichota, sep/2012)

5.1.2 SurfaceOrganicMatter

The surface organic matter model, ported by Ben Jolley from the Fortran version. Tidied up (somewhat) for ApsimX by Eric Zurcher, August 2014.

5.1.3 MicroClimate

The module MICROMET, described here, has been developed to allow the calculation of potential transpiration for multiple competing canopies that can be either layered or intermingled.

5.1.4 Irrigation

This model controls the irrigation events, which can be triggered using the Apply() method.

5.1.5 Fertiliser

The fertiliser model

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

List of Plant Model Components.

Component Name	Component Type

5.1.6 Operations

This class encapsulates an operations schedule.

5.1.7 Report

A report class for writing output to the data store.

5.1.8 Pruning

The manager model

5.1.9 SoluteManager

Manages access to solutes.

5.2 Weather

Reads in weather data and makes it available to other models.

5.3 Clock

The clock model

5.4 Summary

This model collects the simulation initial conditions and stores into the DataStore. It also provides an API for writing messages to the DataStore.

5.5 Soil Arbitrator

The APSIM farming systems model has a long history of use for simulating mixed or intercropped systems. Doing this requires methods for simulating the competition of above and below ground resources. Above ground competition for light has been calculated within APSIM assuming a mixed turbid medium using the Beer-

Lambert analogue as described by [Keating1993Intercropping]. The MicroClimate Snow et al., 2004 model now used within APSIM builds upon this by also calculating the impact of mutual shading on canopy conductance and partitions aerodynamic conductance to individual species in applying the Penman-Monteith model for calculating potential crop water use. The arbitration of below ground resources of water and nitrogen is calculated by this model.

Traditionally, below ground competition has been arbitrated using two approaches. Firstly, the early approaches Adiku et al., 1995; Carberry et al., 1996 used an alternating order of uptake calculation each day to ensure that different crops within a simulation did not benefit from precedence in daily orders of calculations. Soil water simulations using the SWIM3 model Huth et al., 2012 arbitrate individual crop uptakes as part of the simulataneous solutions of various soil water fluxes as part of its solution of the Richards' equation Richards, 1931.

The soil arbitrator operates via a simple integration of daily fluxes into crop root systems via a Runge-Kutta calculation.

If Y is any soil resource, such as water or N, and U is the uptake of that resource by one or more plant root systems, then

$Y_{t+1} = Y_t - U$

Because U will change through the time period in complex manners depending on the number and nature of demands for that resource, we use Runge-Kutta to integrate through that time period using

$Y_{t+1} = Y_t + 1/6 \times (U_1 + 2xU_2 + 2xU_3 + U_4)$

Where U_1,U_2,U_3 and U_4 are 4 estimates of the Uptake rates calculated by the crop models given a range of soil resource conditions, as follows:

$U_1 = f(Y_t)$,

$U_2 = f(Y_t - 0.5xU_1)$,

$U_3 = f(Y_t - 0.5xU_2)$,

$U_4 = f(Y_t - U_3)$.

So U_1 is the estimate based on the uptake rates at the beginning of the time interval, similar to a simple Euler method. U_2 and U_3 are estimates based on the rates somewhere near the midpoint of the time interval. U_4 is the estimate based on the rates toward the end of the time interval.

The iterative procedure allows crops to influence the uptake of other crops via various feedback mechanisms. For example, crops rapidly extracting water from near the surface will dry the soil in those layers, which will force deeper rooted crops to potentially extract water from lower layers. Uptakes can notionally be of either sign, and so trees providing hydraulic lift of water from water tables could potentially make this water available for uptake by mutplie understory species within the timestep. Crops are responsible for meeting resource demand by whatever means they prefer. And so, leguminous crops may start by taking up mineral N at the start of the day but rely on fixation later in a time period if N becomes limiting. This will reduce competition from others and change the balance dynamically throughout the integration period.

The design has been chosen to provide the following benefits:

1) The approach is numerically simple and pure.

2) The approach does not require the use of any particular uptake equation. The uptake equation is embodied within the crop model as designed by the crop model developer and tester.

3) The approach will allow any number of plant species to interact.

4) The approach will allow for arbitration between species in any zone, but also competition between species that may demand resources from multiple zones within the simulation.

5) The approach will automatically arbitrate supply of N between zones, layers, and types (nitrate vs ammonium) with the preferences of all derived by the plant model code.

6 PhenoTest

6.1 PhenoValidation

7 References

Adiku, S. G. K., Carberry, P. S., Rose, C. W., McCown, R. L., Braddock, R., 1995. A maize (zea-mays) - cowpea (vigna-unguiculata) intercrop model. Ecophysiology of Tropical Intercropping, Inst Natl Recherche Agronomique, Paris, 397-406.

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.

Carberry, P.S., McCown, R. L., Muchow, R. C., Dimes, J. P., Probert, M. E., 1996. Simulation of a legume ley farming system in northern Australia using the Agricultural Production Systems Simulator. Australian Journal of Experimental Agriculture 36 (8), 1037-1048.

Holzworth, Dean P., Huth, Neil I., deVoil, Peter G., Zurcher, Eric J., Herrmann, Neville I., McLean, Greg, Chenu, Karine, van Oosterom, Erik J., Snow, Val, Murphy, Chris, Moore, Andrew D., Brown, Hamish, Whish, Jeremy P. M., Verrall, Shaun, Fainges, Justin, Bell, Lindsay W., Peake, Allan S., Poulton, Perry L., Hochman, Zvi, Thorburn, Peter J., Gaydon, Donald S., Dalgliesh, Neal P., Rodriguez, Daniel, Cox, Howard, Chapman, Scott, Doherty, Alastair, Teixeira, Edmar, Sharp, Joanna, Cichota, Rogerio, Vogeler, Iris, Li, Frank Y., Wang, Enli, Hammer, Graeme L., Robertson, Michael J., Dimes, John P., Whitbread, Anthony M., Hunt, James, van Rees, Harm, McClelland, Tim, Carberry, Peter S., Hargreaves, John N. G., MacLeod, Neil, McDonald, Cam, Harsdorf, Justin, Wedgwood, Sara, Keating, Brian A., 2014. APSIM – Evolution towards a new generation of agricultural systems simulation. Environmental Modelling and Software 62, 327-350.

Huth, N.I., Bristow, K.L., Verburg, K., 2012. SWIM3: Model use, calibration, and validation. Transactions of the ASABE 55 (4), 1303-1313.

Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N. G., Meinke, H., Hochman, Z., McLean, G., Verburg, K., Snow, V., Dimes, J. P., Silburn, M., Wang, E., Brown, S., Bristow, K. L., Asseng, S., Chapman, S., McCown, R. L., Freebairn, D. M., Smith, C. J., 2003. An overview of APSIM, a model designed for farming systems simulation. European Journal of Agronomy 18 (3-4), 267-288.

Lawless, Conor, Semenov, MA, Jamieson, PD, 2005. A wheat canopy model linking leaf area and phenology. European Journal of Agronomy 22 (1), 19-32.

McCown, R. L., Hammer, G. L., Hargreaves, J. N. G., Holzworth, D., Huth, N. I., 1995. APSIM: an agricultural production system simulation model for operational research. Mathematics and Computers in Simulation 39 (3-4), 225-231.

McCown, R. L., Hammer, G. L., Hargreaves, J. N. G., Holzworth, D. P., Freebairn, D. M., 1996. APSIM: a Novel Software System for Model Development, Model Testing and Simulation in Agricultural Systems Research. Agricultural Systems 50 (3), 255-271.

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.

Richards, Lorenzo Adolph, 1931. Capillary conduction of liquids through porous mediums. Journal of Applied Physics 1 (5), 318-333.

Snow, V. O., Huth, N. I., 2004. The APSIM MICROMET Module. HortResearch Internal Report, HortResearch, Auckland.