

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

Peculiarities 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 training systems. They differ in phenology and vegetative growth.

Including a Grapevine crop in an APSIM simulation

To include 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 senesce.

Background: 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 community of developers and users who work together, to share ideas, data and source code,
4. a data platform to enable this sharing and
5. 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.

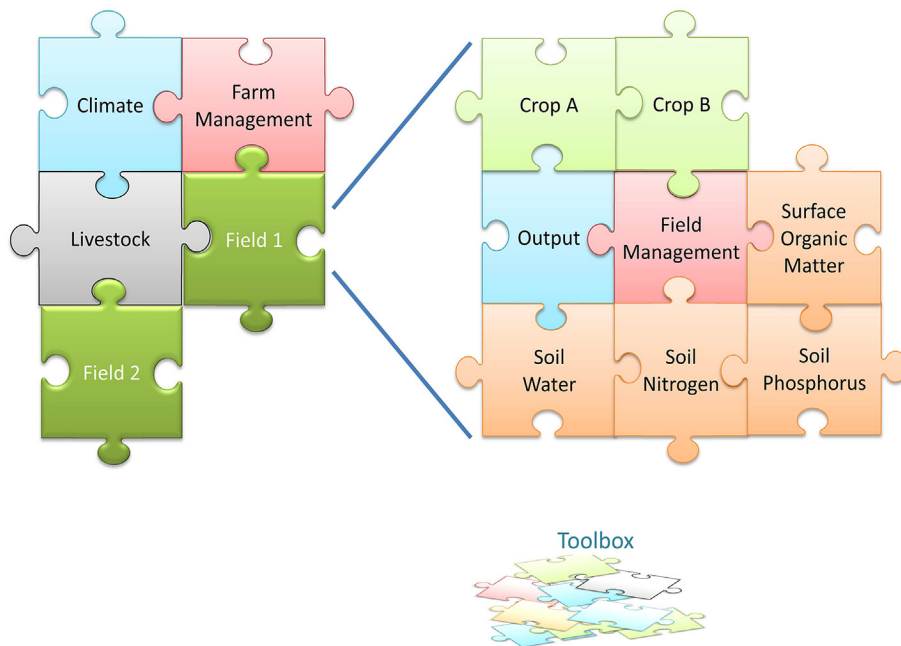


Figure: This conceptual representation of an APSIM simulation shows a “top level” farm (with climate, farm management and livestock) and two fields. The farm and each field are built from a combination of models found in the toolbox. The APSIM infrastructure connects all selected model pieces together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

1 Model description

1.1 SauvignonBlanc

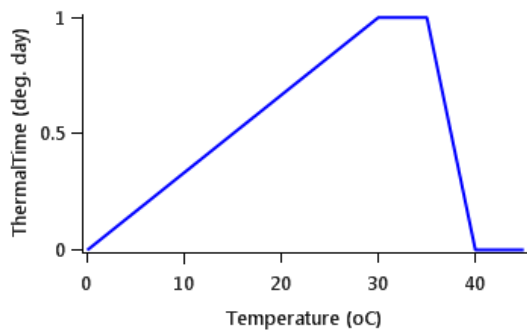
Cultivar class for holding cultivar overrides.

1.2 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.

1.2.1 ThermalTime

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



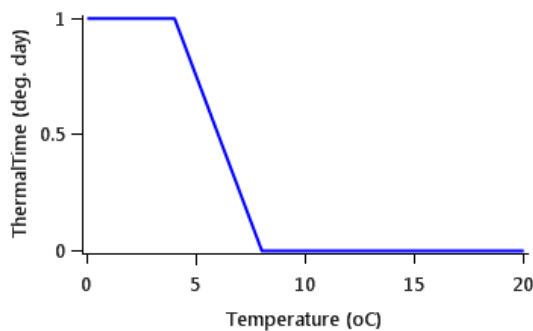
X	Y
0	0
30	1
35	1
40	0
45	0

1.2.2 Dormant Phase

This phase goes from StartDomancy to EndDomancy.

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.

A value 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

1.2.3 Budding Phase

This phase goes from EndDomancy to BudBurst.

has all the functionality of generic phase, but used to set the emerging date of perennial 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.

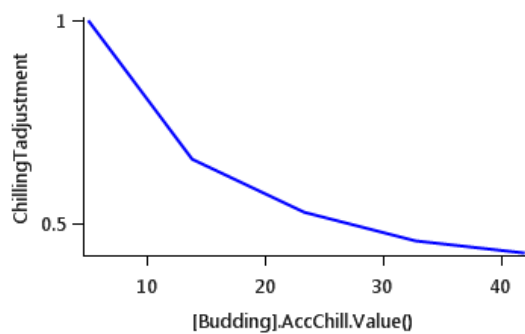
$ThermalTime = [Phenology].ThermalTime$

$Target = Base \times ChillingTadjustment$

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 StartDomancy and BudBurst stages. Function values added to the accumulate total each day are:

$$ThermalTime = [Phenology].Dormant.ThermalTime$$

1.2.4 ShootGrowth 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.

$$ThermalTime = [Phenology].ThermalTime$$

1.2.5 FlowerDevelopment 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.

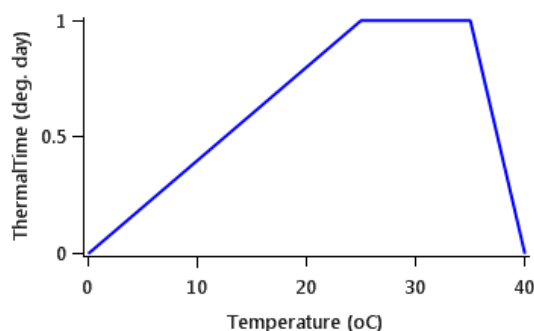
$$ThermalTime = [Phenology].ThermalTime$$

1.2.6 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.

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



X	Y
0	0
25	1
35	1
40	0

1.2.7 LeafDeathPhase Phase

This phase goes from Veraison to LeafFall.

Leaf death phenological phase

$$ThermalTime = [Phenology].ThermalTime$$

1.2.8 GotoPhase Phase

This phase goes from LeafFall to Unused.

A special phase that jumps to another phase.

1.2.9 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:

$$PostEventValue = [Clock].Today.DayOfYear$$

1.2.10 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:

$$PostEventValue = [Clock].Today.DayOfYear$$

1.2.11 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:

$$PostEventValue = [Clock].Today.DayOfYear$$

1.3 Structure

The structure model calculates structural development of the plant. This includes the number of primordia, leaves, stems and nodes, as well as overall plant height.

1.3.1 MainStemPrimordialInitiationRate

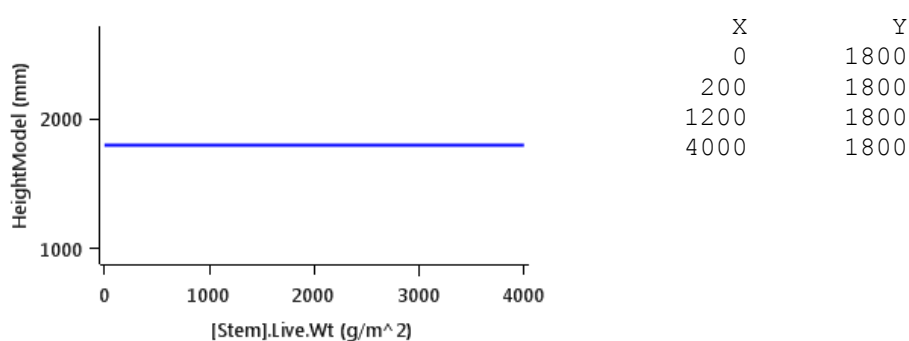
0 between BudBurst and Veraison and a value of zero outside of this period

1.3.2 MainStemNodeAppearanceRate

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

1.3.3 HeightModel

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

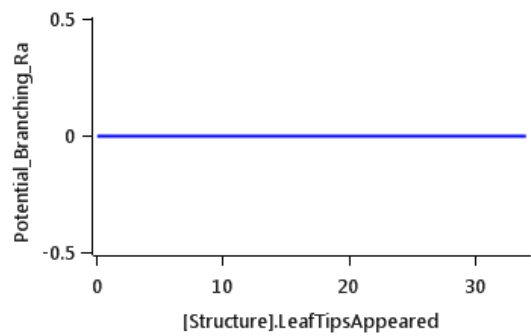


1.3.4 BranchingRate

$BranchingRate = Potential_Branching_Rate \times LinearInterpolationFunction$

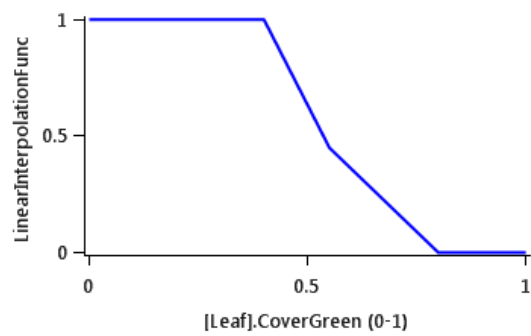
Where:

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

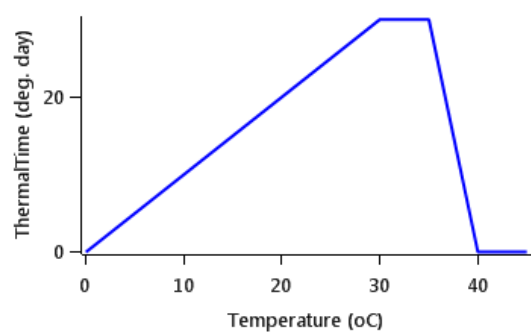
LinearInterpolationFunction is calculated as a function of *[Leaf].CoverGreen*



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

1.3.5 ThermalTime

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



X	Y
0	0
30	30
35	30
40	0
45	0

1.4 Leaf

This provides a phytomer type which predicts the area, biomass and nutrient as the tally of separate cohorts of leaves. A cohort of leaves is represented by a main-stem node position and branch leaves are kept in the same cohort as the mainstem leaf appearing at the same time (Lawless et al., 2005). The Leaf class delegates the status and function of individual cohorts into LeafCohort sub-classes.

$ThermalTime = [Structure].ThermalTime$

$Area = 1000$

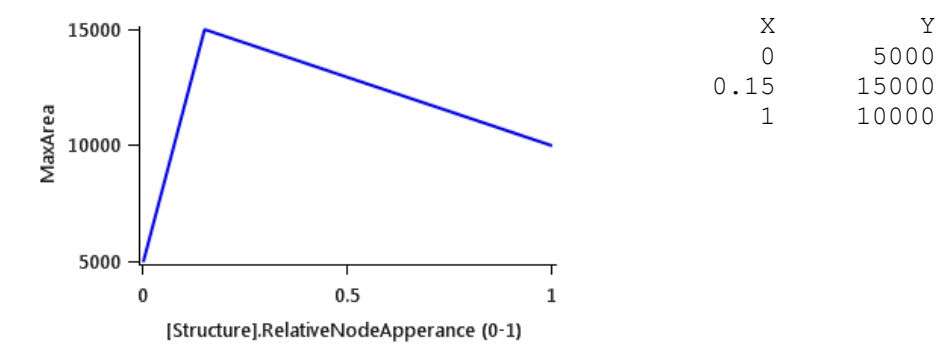
$Area = 0$

Area = 0

1.4.1 CohortParameters

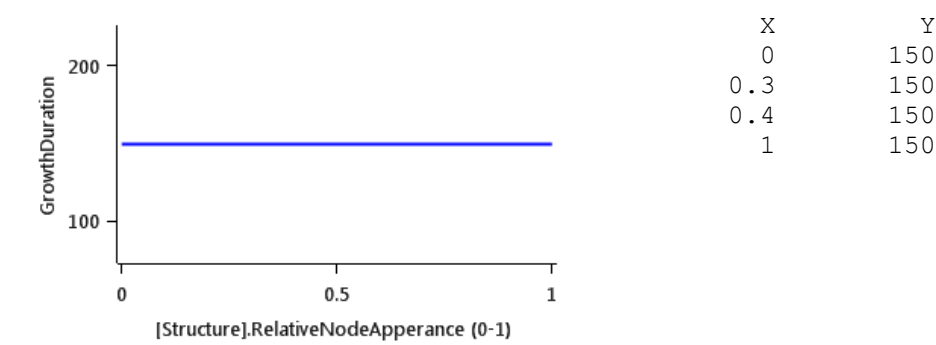
1.4.1.1 MaxArea

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



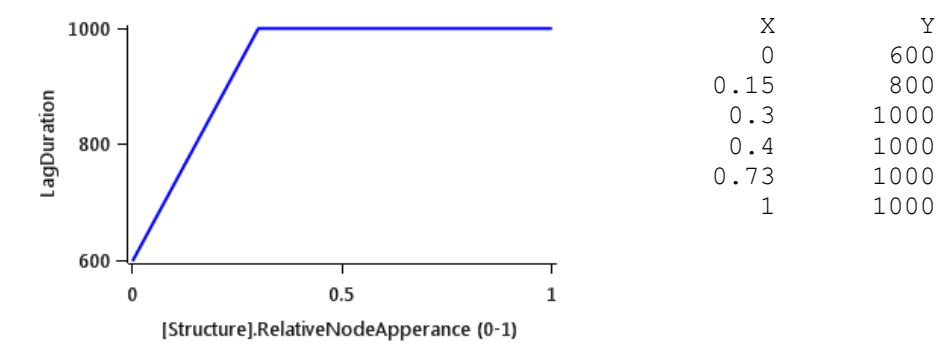
1.4.1.2 GrowthDuration

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



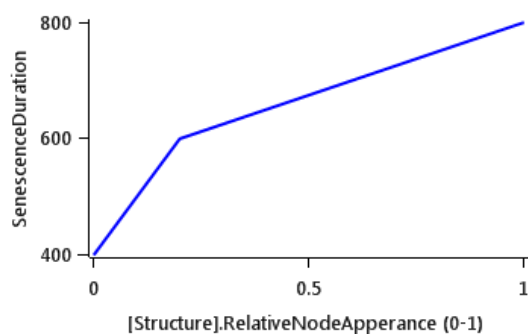
1.4.1.3 LagDuration

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



1.4.1.4 SenescenceDuration

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



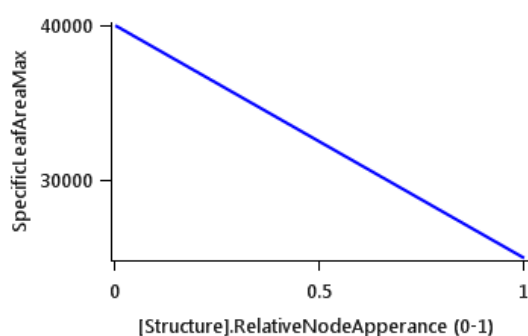
X	Y
0	400
0.2	600
1	800

1.4.1.5 DetachmentLagDuration = 1000000 (degreeDay)

1.4.1.6 DetachmentDuration = 1000000 (degreeDay)

1.4.1.7 SpecificLeafAreaMax

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



X	Y
0	40000
1	25000

1.4.1.8 SpecificLeafAreaMin = 20000 (mm2/g)

1.4.1.9 StructuralFraction = 0.99 (percentage)

1.4.1.10 MaximumNConc = 0.05 (g/g)

1.4.1.11 MinimumNConc = 0.01 (g/g)

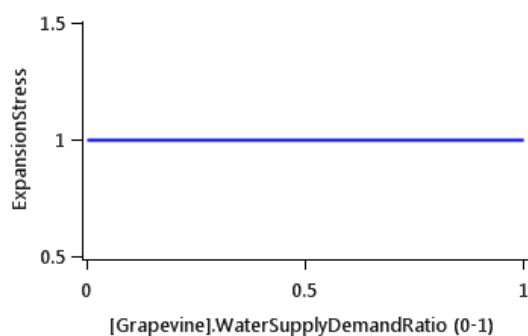
1.4.1.12 InitialNConc = 0.04 (g/g)

1.4.1.13 NReallocationFactor = 0 (percentage)

1.4.1.14 NRetranslocationFactor = 0 (percentage)

1.4.1.15 ExpansionStress

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



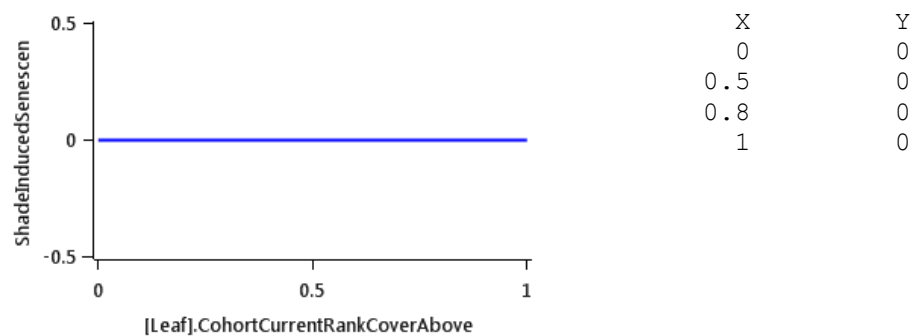
X	Y
0	1
1	1

1.4.1.16 CriticalNConc = 0.01 (g/g)

1.4.1.17 DMRetranslocationFactor = 0 (percentage)

1.4.1.18 ShadeInducedSenescenceRate

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



1.4.1.19 NonStructuralFraction = 0.2 (percentage)

1.4.1.20 SenessingLeafRelativeSize = 1 (0-1)

1.4.1.21 DMReallocationFactor = 0 (percentage)

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

1.4.1.23 MaintenanceRespirationFunction = 0 (g/g)

1.4.1.24 DroughtInducedLagAcceleration = 1

1.4.1.25 DroughtInducedSenAcceleration = 1

1.4.1.26 CellDivisionStress = 1 (0-1)

1.4.2 Photosynthesis

Biomass accumulation is the product of the amount 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 CO2 concentration (Fco2). NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

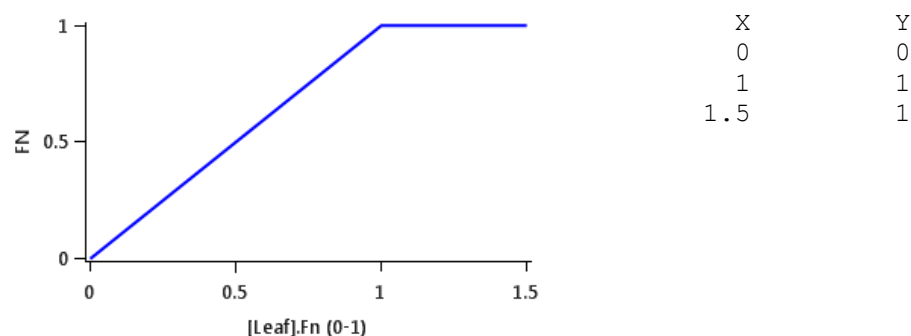
$RUE = 0.8 \text{ (g/MJ)}$

1.4.2.1 FCO2

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

1.4.2.2 FN

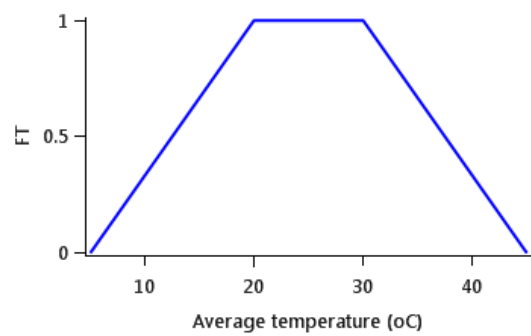
FN is calculated as a function of [Leaf].Fn



1.4.2.3 FT

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

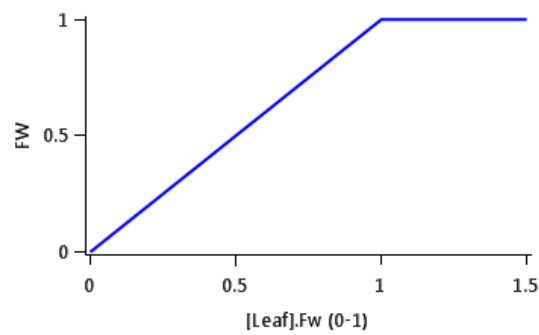
MaximumTemperatureWeighting = 0.75



X	Y
5	0
20	1
30	1
45	0

1.4.2.4 FW

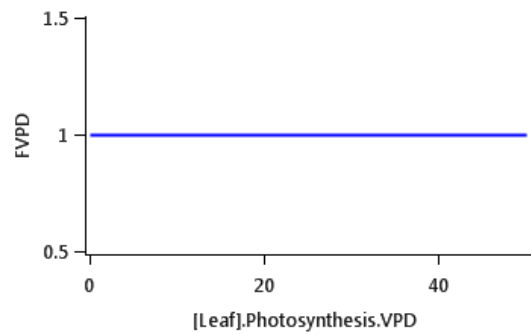
FW is calculated as a function of [Leaf].Fw



X	Y
0	0
1	1
1.5	1

1.4.2.5 FVPD

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



X	Y
0	1
10	1
50	1

RadnInt = [Leaf].RadIntTot

1.4.3 FRGRFunction

FRGRFunction = minimum (RUE_FT, Others)

Where:

RUE_FT = [Leaf].Photosynthesis.FT

Others = minimum (RUE_FN, RUE_FVPD)

Where:

$RUE_FN = [Leaf].Photosynthesis.FN$

$RUE_FVPD = [Leaf].Photosynthesis.FVPD$

1.4.4 ExtinctionCoeff

$ExtinctionCoeff = Constant$

Where:

$Constant = 0.5$

1.4.5 Live

A composite biomass i.e. a biomass made up of 1 or more biomass objects.

1.4.6 Dead

A composite biomass i.e. a biomass made up of 1 or more biomass objects.

1.4.7 Total

A composite biomass i.e. a biomass made up of 1 or more biomass objects.

1.4.8 CohortArrayLive

This class encapsulates an array of biomass objects

1.4.9 CohortArrayDead

This class encapsulates an array of biomass objects

1.4.10 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 20% of leaf biomass will be removed with none of it going to the surface organic matter pool

If a **cut** is performed and no fractions are specified then 20% of leaf biomass will be removed with all of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 50% of leaf biomass will be removed with all of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 70% of leaf biomass will be removed with 10% of it going to the surface organic matter pool

1.4.11 StructuralFraction = 0.99 (percentage)

1.4.12 FrostFraction

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

1.4.13 DMConversionEfficiencyFunction = 1

1.4.14 ApexStandard

Calculate cohort population using stem population.

1.5 Root

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

Dry Matter Demands

By default, 100% of the dry matter (DM) demanded from the root is structural, but this can be modified by using StructuralFraction different than one. The daily DM demand from root is calculated as a proportion of total DM supply using a PartitionFraction function. 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 NonStructural and can be used for retranslocation and/or reallocation if the respective factors are set to values other than zero.

Nitrogen Uptake

Potential N 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 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. 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.

1.5.1 Live

A composite biomass i.e. a biomass made up of 1 or more biomass objects.

1.5.2 Dead

A composite biomass i.e. a biomass made up of 1 or more biomass objects.

1.5.3 NitrogenDemandSwitch

Returns a value of 1 if phenology is between start and end phases and otherwise a value of 0.

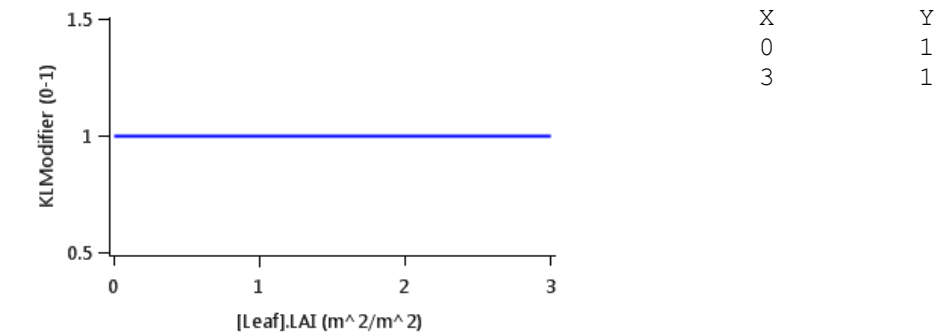
1.5.4 MinimumNConc = 0.01 (g/g)

1.5.5 MaximumNConc = 0.01 (g/g)

1.5.6 MaximumRootDepth = 1000000 (mm)

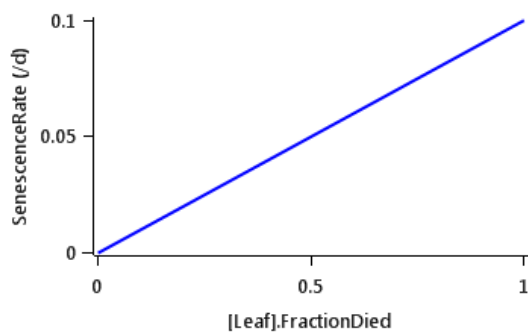
1.5.7 KLModifier

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



1.5.8 SenescenceRate

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



X	Y
0	0
1	0.1

1.5.9 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 20% of root biomass will be removed with all of it going to the surface organic matter pool

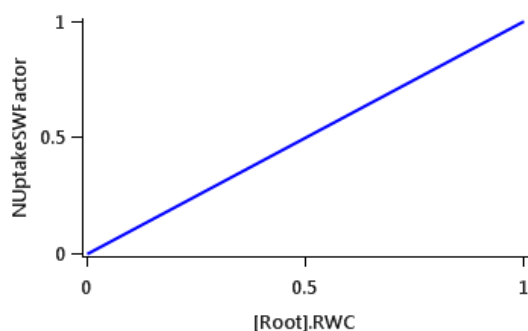
If a **cut** is performed and no fractions are specified then 30% of root biomass will be removed with all of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 10% of root biomass will be removed with all of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 15% of root biomass will be removed with all of it going to the surface organic matter pool

1.5.10 NUptakeSWFactor

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



X	Y
0	0
1	1

1.5.11 InitialDM = 5 (g/plant)

1.5.12 SpecificRootLength = 40 (m/g)

1.5.13 RootFrontVelocity = 10 (mm/d)

1.5.14 PartitionFraction = 0.2 (percentage)

1.5.15 KNO3 = 0.02

1.5.16 KNH4 = 0.003

1.5.17 MaxDailyNUptake = 6 (kg N/ha)

1.6 Stem

This organ is simulated using a generic organ type.

Dry Matter Demands

A given fraction of daily DM demand is determined to be structural and the remainder is non-structural.

Dry Matter Supplies

A given fraction of Nonstructural DM is made available to the arbitrator as `DMReTranslocationSupply`.

Nitrogen Demands

The daily nonstructural N demand is the product of Total DM demand and a Maximum N concentration less the structural N demand. The daily structural N demand is the product of Total DM demand and a Minimum N concentration. The Nitrogen demand switch is a multiplier applied to nitrogen demand so it can be turned off at certain phases.

Nitrogen Supplies

As the organ senesces a fraction of senesced N is made available to the arbitrator as `NReallocationSupply`. A fraction of nonstructural N is made available to the arbitrator as `NRetranslocationSupply`.

Biomass Senescence and Detachment

Senescence is calculated as a proportion of the live dry matter. Detachment of biomass into the surface organic matter pool is calculated daily as a proportion of the dead DM.

Canopy

The user can model the canopy by specifying either the LAI and an extinction coefficient, or by specifying the canopy cover directly. If the cover is specified, LAI is calculated using an inverted Beer-Lambert equation with the specified cover value.

The canopies values of Cover and LAI are passed to the MicroClimate module which uses the Penman Monteith equation to calculate potential evapotranspiration for each canopy and passes the value back to the crop. The effect of growth rate on transpiration is captured using the Fractional Growth Rate (FRGR) function which is parameterised as a function of temperature for the simple leaf.

1.6.1 `NReallocationFactor = 0` (percentage)

1.6.2 `NRetranslocationFactor = 0.05` (percentage)

1.6.3 `NitrogenDemandSwitch`

Returns a value of 1 if phenology is between start and end phases and otherwise a value of 0.

1.6.4 `DMRetranslocationFactor = 0` (percentage)

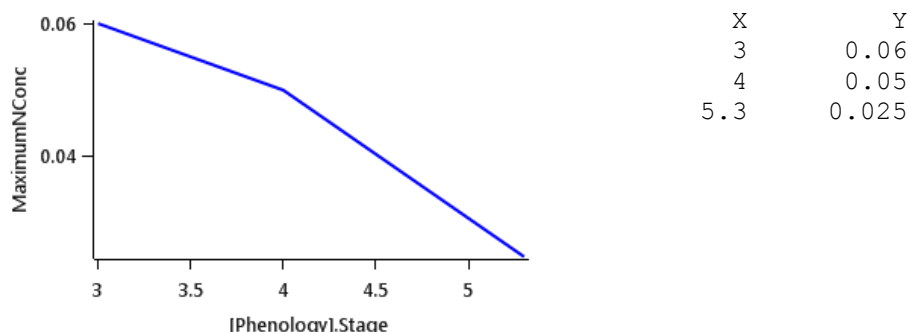
1.6.5 `DMDemandFunction`

Calculate internode demand

InterNodeWt = 0.2

1.6.6 `MaximumNConc`

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



1.6.7 `MinimumNConc = 0.006` (g/g)

1.6.8 Live

Biomass of plant organs

1.6.9 Dead

Biomass of plant organs

1.6.10 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 60% of stem biomass will be removed with 10% of it going to the surface organic matter pool

If a **cut** is performed and no fractions are specified then 80% of stem biomass will be removed with none of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 60% of stem biomass will be removed with all of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 80% of stem biomass will be removed with 20% of it going to the surface organic matter pool

1.6.11 InitialWtFunction = 0 (g/m²)

1.6.12 StructuralFraction = 1 (g/g)

1.6.13 DetachmentRateFunction = 0 (1/d)

1.6.14 SenescenceRate = 0 (/d)

1.6.15 MaintenanceRespirationFunction = 0 (0-1)

1.6.16 DMConversionEfficiency = 1 (0-1)

1.7 Cordon

This organ is simulated using a generic organ type.

Dry Matter Demands

A given fraction of daily DM demand is determined to be structural and the remainder is non-structural.

Dry Matter Supplies

A given fraction of Nonstructural DM is made available to the arbitrator as DMReTranslocationSupply.

Nitrogen Demands

The daily nonstructural N demand is the product of Total DM demand and a Maximum N concentration less the structural N demand. The daily structural N demand is the product of Total DM demand and a Minimum N concentration. The Nitrogen demand switch is a multiplier applied to nitrogen demand so it can be turned off at certain phases.

Nitrogen Supplies

As the organ senesces a fraction of senesced N is made available to the arbitrator as NReallocationSupply. A fraction of nonstructural N is made available to the arbitrator as NRetranslocationSupply

Biomass Senescence and Detachment

Senescence is calculated as a proportion of the live dry matter. Detachment of biomass into the surface organic matter pool is calculated daily as a proportion of the dead DM.

Canopy

The user can model the canopy by specifying either the LAI and an extinction coefficient, or by specifying the canopy cover directly. If the cover is specified, LAI is calculated using an inverted Beer-Lambert equation with the specified cover value.

The canopy values of Cover and LAI are passed to the MicroClimate module which uses the Penman Monteith equation to calculate potential evapotranspiration for each canopy and passes the value back to the crop. The effect of growth rate on transpiration is captured using the Fractional Growth Rate (FRGR) function which is parameterised as a function of temperature for the simple leaf.

1.7.1 NReallocationFactor = 0 (percentage)

1.7.2 NRetranslocationFactor = 0.05 (percentage)

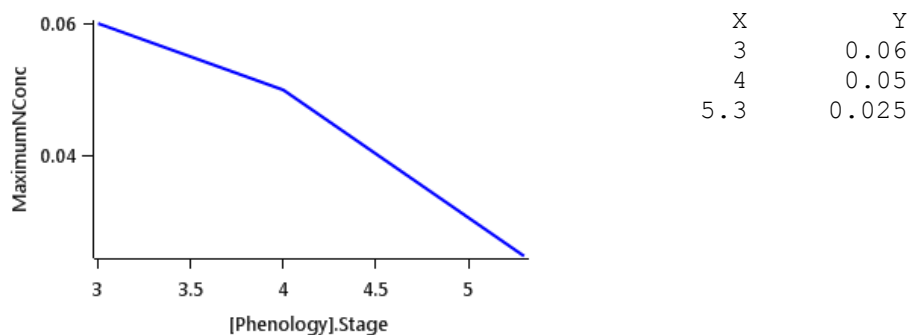
1.7.3 NitrogenDemandSwitch

Returns a value of 1 if phenology is between start and end phases and otherwise a value of 0.

1.7.4 DMRetranslocationFactor = 0 (percentage)

1.7.5 MaximumNConc

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



1.7.6 MinimumNConc = 0.006 (g/g)

1.7.7 Live

Biomass of plant organs

1.7.8 Dead

Biomass of plant organs

1.7.9 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 0% of cordon biomass will be removed with none of it going to the surface organic matter pool

If a **cut** is performed and no fractions are specified then 0% of cordon biomass will be removed with none of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 0% of cordon biomass will be removed with none of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 0% of cordon biomass will be removed with none of it going to the surface organic matter pool

1.7.10 InitialWtFunction = 200 (g/m²)

1.7.11 StructuralFraction = 1 (percentage)

1.7.12 DetachmentRateFunction = 0 (/d)

1.7.13 SenescenceRate = 0 (/d)

1.7.14 MaintenanceRespirationFunction = 0.001 (0-1)

1.7.15 DMConversionEfficiency = 1 (0-1)

1.7.16 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

1.8 Trunk

This organ is simulated using a generic organ type.

Dry Matter Demands

A given fraction of daily DM demand is determined to be structural and the remainder is non-structural.

Dry Matter Supplies

A given fraction of Nonstructural DM is made available to the arbitrator as DMReTranslocationSupply.

Nitrogen Demands

The daily nonstructural N demand is the product of Total DM demand and a Maximum N concentration less the structural N demand. The daily structural N demand is the product of Total DM demand and a Minimum N concentration. The Nitrogen demand switch is a multiplier applied to nitrogen demand so it can be turned off at certain phases.

Nitrogen Supplies

As the organ senesces a fraction of senesced N is made available to the arbitrator as NReallocationSupply. A fraction of nonstructural N is made available to the arbitrator as NRetranslocationSupply

Biomass Senescence and Detachment

Senescence is calculated as a proportion of the live dry matter. Detachment of biomass into the surface organic matter pool is calculated daily as a proportion of the dead DM.

Canopy

The user can model the canopy by specifying either the LAI and an extinction coefficient, or by specifying the canopy cover directly. If the cover is specified, LAI is calculated using an inverted Beer-Lambert equation with the specified cover value.

The canopies values of Cover and LAI are passed to the MicroClimate module which uses the Penman Monteith equation to calculate potential evapotranspiration for each canopy and passes the value back to the crop. The effect of growth rate on transpiration is captured using the Fractional Growth Rate (FRGR) function which is parameterised as a function of temperature for the simple leaf.

1.8.1 NReallocationFactor = 0 (/d)

1.8.2 NRetranslocationFactor = 0.05 (/d)

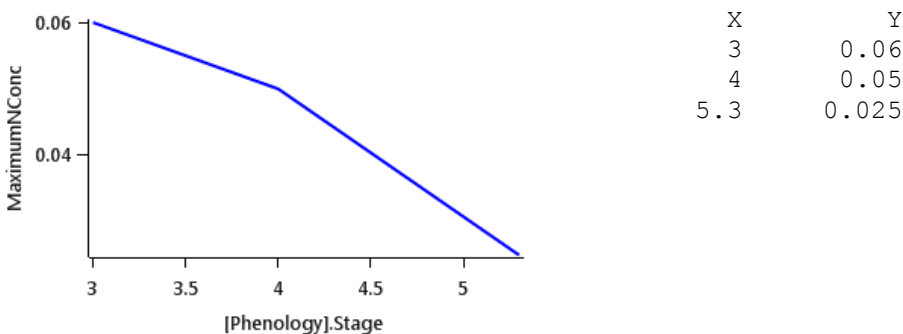
1.8.3 NitrogenDemandSwitch

Returns a value of 1 if phenology is between start and end phases and otherwise a value of 0.

1.8.4 DMRetranslocationFactor = 0 (/d)

1.8.5 MaximumNConc

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



1.8.6 MinimumNConc = 0.006

1.8.7 Live

Biomass of plant organs

1.8.8 Dead

Biomass of plant organs

1.8.9 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 0% of trunk biomass will be removed with none of it going to the surface organic matter pool

If a **cut** is performed and no fractions are specified then 0% of trunk biomass will be removed with none of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 0% of trunk biomass will be removed with none of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 0% of trunk biomass will be removed with none of it going to the surface organic matter pool

1.8.10 InitialWtFunction = 2000 (g/m²)

1.8.11 StructuralFraction = 1 (g/g)

1.8.12 DetachmentRateFunction = 0 (/d)

1.8.13 SenescenceRate = 0 (/d)

1.8.14 MaintenanceRespirationFunction = 0.001 (0-1)

1.8.15 DMConversionEfficiency = 1 (0-1)

1.8.16 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

1.9 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).

1.9.1 Live

Biomass of plant organs

1.9.2 Dead

Biomass of plant organs

1.9.3 MaxNConcDailyGrowth = 0.01 (g/g)

1.9.4 NFillingRate = 0 (g/m²/d)

1.9.5 MinimumNConc = 0 (g/g)

1.9.6 MaximumNConc = 0.0001 (g/g)

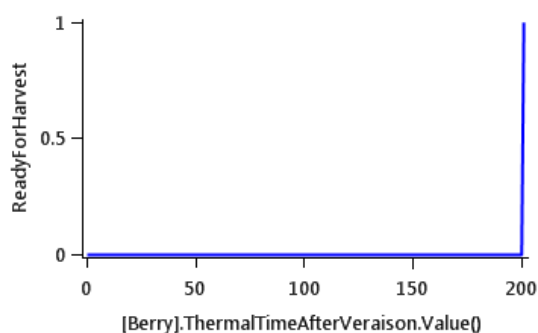
1.9.7 ThermalTimeAfterVeraison

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:

$ThermalTime = [Phenology].ThermalTime$

1.9.8 ReadyForHarvest

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



X	Y
0	0
200	0
201	1

1.9.9 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:

$ThermalTime = [Phenology].ThermalTime$

1.9.10 SolubleSolidsConc

An exponential function

$ThermalTime = [Berry].DaysAfterVeraison$

1.9.11 NumberFunction

$NumberFunction = [Structure].MainStemPopn \times BunchesPerShoot \times PotentialBerriesPerBunch$

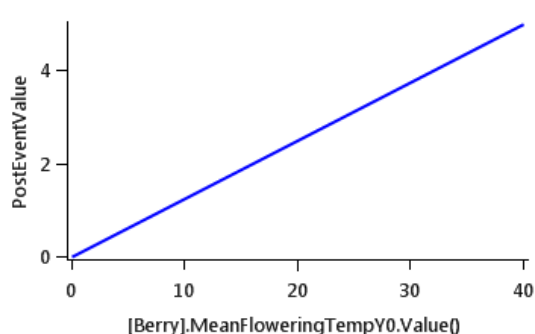
Where:

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 $y = Xmax \cdot 1 / 1 + e^{-(XValue - Xo) / b}$

$XValue = [Berry].MeanFloweringTemp$

$$Y_{max} = 44$$

$$X_0 = 16.3$$

$$b = 0.7$$

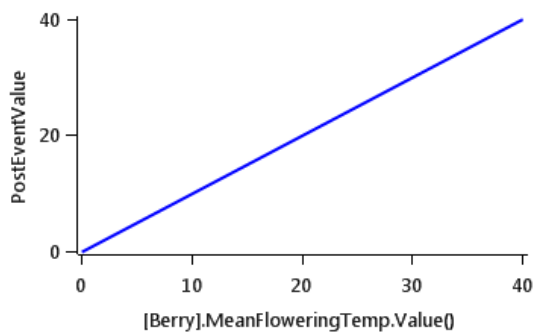
1.9.12 MeanFloweringTempY0

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

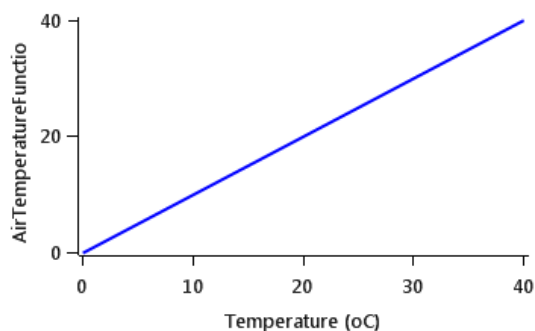
1.9.13 MeanFloweringTemp

$$MeanFloweringTemp = Temperature / Time$$

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:

A value 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:

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

1.9.14 DMDemandFunction

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

$$TT = [Structure].ThermalTime$$

1.9.15 BiomassRemovalDefaults

This class implements biomass removal from live + dead pools.

If a **harvest** is performed and no fractions are specified then 100% of berry biomass will be removed with none of it going to the surface organic matter pool

If a **cut** is performed and no fractions are specified then 100% of berry biomass will be removed with none of it going to the surface organic matter pool

If a **prune** is performed and no fractions are specified then 100% of berry biomass will be removed with all of it going to the surface organic matter pool

If a **graze** is performed and no fractions are specified then 0% of berry biomass will be removed with none of it going to the surface organic matter pool

If a **thin** is performed and no fractions are specified then 0% of berry biomass will be removed with none of it going to the surface organic matter pool

1.9.16 MaximumPotentialGrainSize = 2 (g)

1.9.17 WaterContent

WaterContent = Constant - PhaseLookupValue

Where:

Constant = 0.903

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

1.10 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 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 senesces and may be re-translocated when demand is high relative to supply.
- **Non-structural 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**. Non-structural 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 DoPotentialDMAAllocation() method which works out how much biomass each organ would be allocated assuming N supply is not limiting and sends these allocations to the organs. Each organ then uses their potential DM allocation to determine their N demand (how much N is needed to produce that much DM) and the arbitrator calls DoNSetup() 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 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.
8. **doActualPlantPartitioning**. On this event the arbitrator calls DoNRetranslocation() and DoNFixation() to satisfy any unmet N demands from these sources. Finally, DoActualDMAAllocation is called where DM allocations to each organ are reduced if the N allocation is insufficient to achieve the organs minimum N concentration and final allocations are sent to organs.

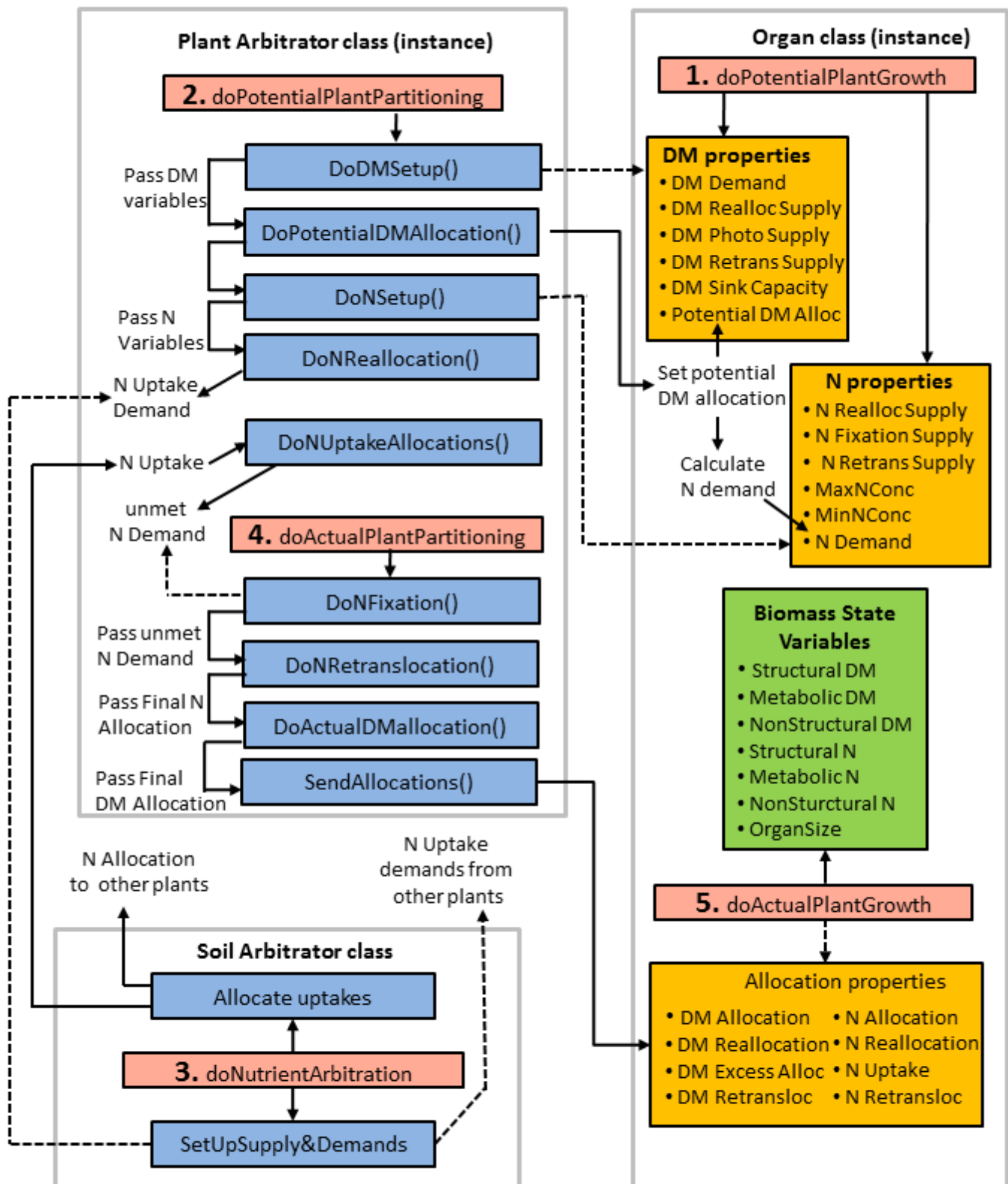


Figure 1. Schematic showing procedure for arbitration of biomass partitioning. Pink boxes are events that are broadcast each day by the model infrastructure and their numbering shows the order of procedure. Blue boxes are methods that are called when these events are broadcast. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.

1.10.1 NArbitrator

Relative allocation class

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.

1.10.2 DMArbitrator

Relative allocation class

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.

1.11 Berry

1.12 PhenoValidation

1.13 Statistics

	Variable	n	Slope	Intercept	SEslope	SEinter	R2	RMSE	NSE	ME	MAE	RSR
rapevine.Phenology.BudBurstDOY.Value()		40	0.294	175.481	0.197	54.884	0.055	22.507	-15.583	-21.125	21.675	4.021
rapevine.Phenology.FloweringDOY.Value()		42	0.423	179.337	0.149	51.376	0.168	21.131	-15.415	-20.333	20.333	4.003
rapevine.Phenology.VeraisonDOY.Value()		36	1.620	-66.501	3.781	282.897	0.005	125.994	-498.302	-20.250	100.528	22.033

2 References

- Brown, H.E, Huth, N, Holzworth, D., 2011. A potato model build using the APSIM Plant.NET framework., 961-967.
- Brown, Hamish E., Huth, Neil I., Holzworth, Dean P., Teixeira, Edmar I., Zyskowski, Rob F., Hargreaves, John N. G., Moot, Derrick J., 2014. Plant Modelling Framework: Software for building and running crop models on the APSIM platform. *Environmental Modelling and Software* 62, 385-398.
- 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.
- 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.