1 The APSIM Maize Model

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

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

The model consists of:

- a phenology model to simulate development between growth phases
- a 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

2 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.
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.

3 Model description

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

List of Plant Model Components.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Component Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arbitrator</td>
<td>Models.PMF.OrganArbitrator</td>
</tr>
<tr>
<td>Phenology</td>
<td>Models.PMF.Phen.Phenology</td>
</tr>
<tr>
<td>Structure</td>
<td>Models.PMF.Struct.Structure</td>
</tr>
<tr>
<td>Grain</td>
<td>Models.PMF.Organs.ReproductiveOrgan</td>
</tr>
<tr>
<td>Root</td>
<td>Models.PMF.Organs.Root</td>
</tr>
<tr>
<td>Leaf</td>
<td>Models.PMF.Organs.Leaf</td>
</tr>
<tr>
<td>Husk</td>
<td>Models.PMF.Organs.GenericOrgan</td>
</tr>
</tbody>
</table>
### 3.1 Arbitrator

The Arbitrator class determines the allocation of dry matter (DM) and Nitrogen between each of the organs in the crop model. Each organ can have up to three different pools of biomass:

- **Structural biomass** which is essential for growth and remains within the organ once it is allocated there.
- **Metabolic biomass** which generally remains within an organ but is able to be re-allocated when the organ senesces and may be retranslocated when demand is high relative to supply.
- **Storage biomass** which is partitioned to organs when supply is high relative to demand and is available for retranslocation to other organs whenever supply from uptake, fixation, or re-allocation is lower than demand.

The process followed for biomass arbitration is shown in Figure 3. Arbitration calculations are triggered by a series of events (shown below) that are raised every day. For these calculations, at each step the Arbitrator exchange information with each organ, so the basic computations of demand and supply are done at the organ level, using their specific parameters.

1. **doPotentialPlantGrowth.** When this event occurs, each organ class executes code to determine their potential growth, biomass supplies and demands. In addition to demands for structural, non-structural and metabolic biomass (DM and N) each organ may have the following biomass supplies:
   - **Fixation supply.** From photosynthesis (DM) or symbiotic fixation (N)
   - **Uptake supply.** Typically uptake of N from the soil by the roots but could also be uptake by other organs (eg foliage application of N).
   - **Retranslocation supply.** Storage biomass that may be moved from organs to meet demands of other organs.
   - **Reallocation supply.** Biomass that can be moved from senescing organs to meet the demands of other organs.
2. **doPotentialPlantPartitioning.** On this event the Arbitrator first executes the DoDMSetup() method to gather the DM supplies and demands from each organ, these values are computed at the organ level. It then executes the DoPotentialDMAlocation() method which works out how much biomass each organ would be allocated assuming N supply is not limiting and sends these allocations to the organs. Each organ then uses their potential DM allocation to determine their N demand (how much N is needed to produce that much DM) and the arbitrator calls DoNSetup() to gather the N supplies and demands from each organ and begin N arbitration. Firstly DoNReallocation() is called to redistribute N that the plant has available from senescing organs. After this step any unmet N demand is considered as plant demand for N uptake from the soil (N Uptake Demand).
3. **doNutrientArbitration.** When this event occurs, the soil arbitrator gets the N uptake demands from each plant (where multiple plants are growing in competition) and their potential uptake from the soil and determines how much of their demand that the soil is able to provide. This value is then passed back to each plant instance as their Nuptake and doNUptakeAllocation() is called to distribute this N between organs.
4. **doActualPlantPartitioning.** On this event the arbitrator call DoNretranslocation() and DoNfixation() to satisfy any unmet N demands from these sources. Finally, DoActualDMAllocation is called where DM allocations to each organ are reduced if the N allocation is insufficient to achieve the organs minimum N concentration and final allocations are sent to organs.
Figure 3: Schematic showing the procedure for arbitration of biomass partitioning. Pink boxes represent events that occur every day and their numbering shows the order of calculations. Blue boxes represent the methods that are called when these events occur. Orange boxes contain properties that make up the organ/arbitrator interface. Green boxes are organ specific properties.
### 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

<table>
<thead>
<tr>
<th>Phase Number</th>
<th>Phase Name</th>
<th>Initial Stage</th>
<th>Final Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Germinating</td>
<td>Sowing</td>
<td>Germination</td>
</tr>
<tr>
<td>2</td>
<td>Emerging</td>
<td>Germination</td>
<td>Emergence</td>
</tr>
<tr>
<td>3</td>
<td>Juvenile</td>
<td>Emergence</td>
<td>EndJuvenile</td>
</tr>
<tr>
<td>4</td>
<td>Photosensitive</td>
<td>EndJuvenile</td>
<td>FloralInitiation</td>
</tr>
<tr>
<td>5</td>
<td>LeafAppearance</td>
<td>FloralInitiation</td>
<td>FlagLeaf</td>
</tr>
<tr>
<td>6</td>
<td>FlagLeafToFlowering</td>
<td>FlagLeaf</td>
<td>Flowering</td>
</tr>
<tr>
<td>7</td>
<td>FloweringToGrainFilling</td>
<td>Flowering</td>
<td>StartGrainFill</td>
</tr>
<tr>
<td>8</td>
<td>GrainFilling</td>
<td>StartGrainFill</td>
<td>EndGrainFill</td>
</tr>
<tr>
<td>9</td>
<td>Maturing</td>
<td>EndGrainFill</td>
<td>Maturity</td>
</tr>
<tr>
<td>10</td>
<td>MaturityToHarvestRipe</td>
<td>Maturity</td>
<td>HarvestRipe</td>
</tr>
<tr>
<td>11</td>
<td>ReadyForHarvesting</td>
<td>HarvestRipe</td>
<td>Unused</td>
</tr>
</tbody>
</table>

#### 3.2.1 Phenological Phases

##### 3.2.1.1 Germinating Phase

The model assumes that germination will be completed on the day after sowing, provided that the extractable soil water is greater than zero.

##### 3.2.1.2 Emerging Phase

This phase goes from Germination to Emergence and simulates time to emergence as a function of sowing depth. The *ThermalTime Target* for ending this phase is given by:

\[
\text{Target} = \text{SowingDepth} \times \text{ShootRate} + \text{ShootLag}
\]

Where:
- \(\text{ShootRate} = 0.6 \text{ (deg day/mm)}\),
- \(\text{ShootLag} = 55 \text{ (deg day)}\),
and \(\text{SowingDepth} \text{ (mm)}\) is sent from the manager with the sowing event.

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

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

##### 3.2.1.3 Juvenile Phase

This phase goes from Emergence to EndJuvenile. It uses a *Target* to determine the duration between development *Stages*. Daily progress is accumulated until the *Target* is met and remaining fraction of the day is forwarded to the next phase.

The *Target* and the daily *Progression* toward EndJuvenile are described as follow:

##### 3.2.1.3.1 Target

\[
\text{Target} = 200 \text{ (deg day)}
\]

\[
\text{Progression} = [\text{Phenology}].\text{ThermalTime}
\]
3.2.1.4 Photosensitive Phase

This phase goes from EndJuvenile to FloralInitiation. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward FloralInitiation are described as follow:

\[ \text{Progression} = [\text{Phenology}] \cdot \text{ThermalTime} \]

3.2.1.4.1 Target

The photoperiod is likely to vary with individual maize cultivars. Therefore, the response shown here is indicative only. See the descriptions of individual cultivars for more detail on photoperiod responses used by the maize model.

Target is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
\hline
0 & 0 \\
12.5 & 0 \\
24 & 10 \\
\end{array}
\]

\[ \text{XValue} = [\text{Phenology}] \cdot \text{Photoperiod} \]

3.2.1.5 LeafAppearance Phase

This phase goes from FloralInitiation to FlagLeaf. It continues until the final main-stem leaf has finished expansion. The duration of this phase is determined by leaf appearance rate (Structure.Phyllochron) and the number of leaves produced on the mainstem (Structure.FinalLeafNumber). As such, the model parameterisation of leaf appearance and final leaf number (set in the Structure model) are important for predicting the duration of the crop correctly.

3.2.1.6 FlagLeafToFlowering Phase

This phase goes from FlagLeaf to Flowering. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward Flowering are described as follow:

3.2.1.6.1 Target

Target = 50 (deg day)

\[ \text{Progression} = [\text{Phenology}] \cdot \text{ThermalTime} \]

3.2.1.7 FloweringToGrainFilling Phase

This phase goes from Flowering to StartGrainFill. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward StartGrainFill are described as follow:

3.2.1.7.1 Target

Target = 120 (deg day)

\[ \text{Progression} = [\text{Phenology}] \cdot \text{ThermalTime} \]
3.2.1.8 GrainFilling Phase

This phase goes from StartGrainFill to EndGrainFill. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward EndGrainFill are described as follow:

3.2.1.8.1 Target

Target = 550 (deg day)

Progression = [Phenology].ThermalTime

3.2.1.9 Maturing Phase

This phase goes from EndGrainFill to Maturity. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward Maturity are described as follow:

3.2.1.9.1 Target

Target = 10 (deg day)

Progression = [Phenology].ThermalTime

3.2.1.10 MaturityToHarvestRipe Phase

This phase goes from Maturity to HarvestRipe. It uses a Target to determine the duration between development Stages. Daily progress is accumulated until the Target is met and remaining fraction of the day is forwarded to the next phase.

The Target and the daily Progression toward HarvestRipe are described as follow:

3.2.1.10.1 Target

Target = 10 (deg day)

Progression = [Phenology].ThermalTime

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

ThermalTime = [Phenology].ThermalTime

3.2.2 ThermalTime

ThermalTime = BaseThermalTime

Where:

3.2.3 Photoperiod

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

Twilight = -6 (degrees)

3.2.4 FloweringDAS

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

Before Flowering

PreEventValue = 0

On Flowering the value is set to:

PostEventValue = [Plant].DaysAfterSowing
3.2.5 MaturityDAS

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

Before Maturity

\[ PreEventValue = 0 \]

On Maturity the value is set to:

\[ PostEventValue = [Plant].DaysAfterSowing \]

3.2.6 BBCH

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

<table>
<thead>
<tr>
<th>BeginStage</th>
<th>Growth Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Germinating</td>
<td>BBCH = 5 x FractionThroughPhase</td>
</tr>
<tr>
<td>2</td>
<td>Emerging</td>
<td>BBCH = 5 + 5 x FractionThroughPhase</td>
</tr>
<tr>
<td>3</td>
<td>Juvenile</td>
<td>BBCH = 10 + (Leaf.AppearedCohortNo - 1)</td>
</tr>
<tr>
<td>4</td>
<td>PhotoSensitive</td>
<td>BBCH = 10 + (Leaf.AppearedCohortNo - 1)</td>
</tr>
<tr>
<td>5</td>
<td>LeafAppearance</td>
<td>BBCH = 30 + LeavesAppearedInPhase</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Stage</th>
<th>APSIM Name</th>
<th>BBCH translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>FlagLeafFullyExpanded</td>
<td>55 - Mid Tassel Emergence</td>
</tr>
<tr>
<td>7.0</td>
<td>Flowering</td>
<td>65 - Mid flowering</td>
</tr>
<tr>
<td>8.0</td>
<td>StartGrainFill</td>
<td>70 - Beginning of Grain development</td>
</tr>
<tr>
<td>9.0</td>
<td>EndGrainFill</td>
<td>87 - Physiological maturity</td>
</tr>
<tr>
<td>10</td>
<td>Maturity</td>
<td>99 - Harvest Product</td>
</tr>
</tbody>
</table>

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 provide a height 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

3.3.2.1.1 Phyllochron

The value of Phyllochron from Emergence to FlagLeaf is calculated as follows:

3.3.2.1.1.1 Phyllochron

Phyllochron is calculated using linear interpolation

\[
\begin{array}{|c|c|}
\hline
X & Y \\
1 & 30 \\
3.4 & 30 \\
4 & 42 \\
\hline
\end{array}
\]

\[XValue = [Structure].LeafTipsAppeared\]

Phyllochron has a value of zero for phases not specified above and ThermalTime is given by:

\[ThermalTime = [Phenology].ThermalTime\]

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

3.3.2.2 FinalLeafNumber

FinalLeafNumber is the same as ValueToHold until it reaches FloralInitiation stage when it fixes its value

3.3.2.2.1 ValueToHold

ValueToHold = PrimordiaAtEmergence + PrimordiaDuringJuvenilePhase

Where:

PrimordiaAtEmergence = 6

3.3.2.2.1.1 PrimordiaDuringJuvenilePhase

PrimordiaDuringJuvenilePhase = AddFunction / MeanPlastochron

Where:


The value of MeanPlastochron from Germination to Emergence is calculated as follows:

\[Constant = 15\]

The value of MeanPlastochron from Emergence to FlagLeaf is calculated as follows:

MeanPlastochron = AccumulatedPlastochron / Days

Where:

AccumulatedPlastochron is a daily accumulation of the values of
functions listed below between the Emergence and FloralInitiation stages. Function values added to the accumulate total each day are:

\[ \text{Plastochron} = \text{[Structure].Phyllochron} \times \text{LeafTipsPerPrimordia} \]

Where:

\[ \text{LeafTipsPerPrimordia} = 0.5 \]

Days is a daily accumulation of the values of functions listed below between the Emergence and FloralInitiation stages. Function values added to the accumulate total each day are:

\[ \text{Constant} = 1 \]

MeanPlastochron has a value of zero for phases not specified above

### 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 = 0

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

#### 3.3.3.2 BranchMortality

BranchMortality = ShadeInducedBranchMortality + DroughtInducedBranchMortality

Where:

\[ \text{ShadeInducedBranchMortality} = 0 \]

\[ \text{DroughtInducedBranchMortality} = 0 \]

### 3.3.4 Height

The Height of the crop is calculated by the HeightModel:

#### 3.3.4.1 HeightModel

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

#### 3.3.4.1.1 PotentialHeight

PotentialHeight is calculated using linear interpolation

\[ \begin{array}{c|c}
 X & Y \\
 3 & 20 \\
 4 & 400 \\
 5 & 450 \\
 6 & 1700 \\
\end{array} \]

\[ XValue = [\text{Phenology}.\text{Stage}] \]
3.3.4.1.2 WaterStress

WaterStress is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 0 \\
0.6 & 1 \\
\end{array}
\]

XValue = [Maize].Leaf.Fw

Though tillering is common in some maize varieties under certain agronomic conditions, no tillering is accounted for within this model. Therefore all branching has been parameterised out of the current maize model.

3.3.5 StemSenescenceAge

StemSenescenceAge = 0

3.3.6 zFinalLeafNumber

zFinalLeafNumber = 15

3.4 Grain

This organ uses a generic model for plant reproductive components. Yield is calculated from its components in terms of organ number and size (for example, grain number and grain size).

3.4.1 DMConversionEfficiency

DMConversionEfficiency = 1

3.4.2 RemobilisationCost

RemobilisationCost = 0

3.4.3 InitialGrainProportion

InitialGrainProportion = 0.05

3.4.4 MaximumPotentialGrainSize

MaximumPotentialGrainSize = 0.3 (g)

3.4.5 MaximumGrainsPerCob

MaximumGrainsPerCob = 700 (number)

3.4.6 DMDemandFunction

3.4.6.1 InitialPhase

The value of DMDemandFunction from Flowering to StartGrainFill is calculated as follows:

3.4.6.2 FillingRateFunction

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

NumberFunction = [Grain].NumberFunction

FillingDuration = [Phenology].FloweringToGrainFilling.Target
ThermalTime = [Phenology].ThermalTime

3.4.6.2.1 PotentialSizeIncrement

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

3.4.6.3 LinearPhase

The value of DMDemandFunction from StartGrainFill to EndGrainFill is calculated as follows:

3.4.6.4 FillingRateFunction

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

\[ \text{NumberFunction} = \text{[Grain].NumberFunction} \]
\[ \text{FillingDuration} = \text{[Phenology].GrainFilling.Target} \]
\[ \text{ThermalTime} = \text{[Phenology].ThermalTime} \]

3.4.6.4.1 PotentialSizeIncrement

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

Where:

3.4.6.4.1.1 ProportionLinearPhase

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

DMDemandFunction has a value of zero for phases not specified above

3.4.7 MinimumNConc

MinimumNConc = 0.008 (g/g)

3.4.8 MaximumNConc

3.4.8.1 InitialPhase

The value of MaximumNConc from Flowering to StartGrainFill is calculated as follows:

\[ \text{InitialNconc} = 0.05 \]

3.4.8.2 LinearPhase

The value of MaximumNConc from StartGrainFill to EndGrainFill is calculated as follows:

\[ \text{FinalNconc} = 0.013 \]

MaximumNConc has a value of zero for phases not specified above

3.4.9 WaterContent

WaterContent = 0.12 (g/g)

Water content used to calculate a fresh weight.

3.4.10 NFillingRate

3.4.10.1 GrainFilling

The value of NFillingRate from Flowering to EndGrainFill is calculated as follows:

3.4.10.2 FillingRateFunction

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

\[ \text{NumberFunction} = \text{[Grain].NumberFunction} \]
3.4.10.2.1 FillingDuration

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

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

3.4.10.2.2 PotentialSizeIncrement

\[ \text{PotentialSizeIncrement} = [\text{Grain}].\text{MaximumNConcLinearPhase}.\text{FinalNconc} \times [\text{Grain}].\text{MaximumPotentialGrainSize} \]

NFillingRate has a value of zero for phases not specified above

3.4.11 NumberFunction

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

Where:

3.4.11.1 GrowthRateFactor

\[ \text{GrowthRateFactor} \text{ is calculated using linear interpolation} \]

\[
\begin{array}{c|c}
X & Y \\
0.55 & 0 \\
3.3 & 0.6 \\
6.6 & 1 \\
10 & 1 \\
\end{array}
\]

\[ XValue = [\text{GrowthRateGrainDevelopment}] \]

3.4.12 YieldBuPerAcre

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

3.4.13 GrowthGrainDevelopment

\[ \text{GrowthGrainDevelopment} \text{ is a daily accumulation of the values of functions listed below between the FlagLeaf and StartGrainFill stages. Function values added to the accumulate total each day are:} \]

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

3.4.14 DaysGrainDevelopment

\[ \text{DaysGrainDevelopment} \text{ is a daily accumulation of the values of functions listed below between the FlagLeaf and StartGrainFill stages. Function values added to the accumulate total each day are:} \]

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

3.4.15 GrowthRateGrainDevelopment

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

3.4.16 BiomassRemovalDefaults

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

\[ \text{Carbon Concentration} = 0.4 \]

### 3.5 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. 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 arbitrator and a DM supply returned. By default, 100% of the dry matter (DM) demanded from the root is structural. The daily loss of roots is calculated using a SenescenceRate function. All senesced material is automatically detached and added to the soil FOM.

#### Nitrogen Demands

The daily structural N demand from root is the product of total DM demand and the minimum N concentration. Any N above this is considered Storage and can be used for retranslocation and/or reallocation as the respective factors are set to values other then zero.

#### 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 \(k\text{NO}_3\) or \(k\text{NH}_4\), the concentration of N form (ppm), and a soil moisture factor \(\text{NUptakeSWFactor}\) which typically decreases as the soil dries.

\[
\text{NO}_3\text{ uptake} = \text{NO}_3_i \times k\text{NO}_3 \times \text{NO}_3_{\text{ppm},i} \times \text{NUptakeSWFactor}
\]

\[
\text{NH}_4\text{ uptake} = \text{NH}_4_i \times k\text{NH}_4 \times \text{NH}_4_{\text{ppm},i} \times \text{NUptakeSWFactor}
\]

Nitrogen uptake demand is limited to the maximum daily potential uptake \(\text{MaxDailyNUptake}\) 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 \(KL\text{Modifier}\) function.

\[
\text{SW uptake} = (\text{SW}_i - \text{LL}_i) \times KL_i \times KL\text{Modifier}
\]
3.5.1 KLModifier

The parameter KL is described by Meinke et al (1993) to combine elements of soil conductivity (K) with root length (L) in determining root water uptake. Peake et al (2013) demonstrated that in maize KL varied with plant population through impacts on root length density. Here we adjust KL linearly with plant population.

KLModifier is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 1 \\
10 & 1 \\
\end{array}
\]

\[
X_{\text{Value}} = [\text{Maize}.\text{Population}]
\]

3.5.2 DMConversionEfficiency

DMConversionEfficiency = 1

3.5.3 RemobilisationCost

RemobilisationCost = 0

3.5.4 SoilWaterEffect

SoilWaterEffect = 1

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

3.5.5 MaxDailyNUptake

MaxDailyNUptake = 20 (g/m2/d)

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

3.5.6 SenescenceRate

SenescenceRate = 0.005 (/d)

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

3.5.7 MaximumRootDepth

MaximumRootDepth = 100000

3.5.8 MaximumNConc

MaximumNConc = 0.01 (g/g)

3.5.9 MinimumNConc

MinimumNConc = 0.01 (g/g)

3.5.10 RootFrontVelocity

RootFrontVelocity = PotentialRootFrontVelocity × TemperatureFactor × WaterFactor

Where:
3.5.10.1 PotentialRootFrontVelocity

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

3.5.10.1.1 PreFlowering

The value of PotentialRootFrontVelocity from Germination to Flowering is calculated as follows:

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

3.5.10.1.2 PostFlowering

The value of PotentialRootFrontVelocity from Flowering to Maturity is calculated as follows:

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

PotentialRootFrontVelocity has a value of zero for phases not specified above.

3.5.10.2 TemperatureFactor

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

MaximumTemperatureWeighting = 0.5

![TemperatureFactor graph]

3.5.10.3 WaterFactor

WaterFactor is calculated using linear interpolation.

![WaterFactor graph]

\[ XValue = [\text{SoilWaterScale}] \]

3.5.11 NitrogenDemandSwitch

1 between Germination and Maturity and a value of zero outside of this period.

3.5.12 BiomassRemovalDefaults

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

<table>
<thead>
<tr>
<th>Method</th>
<th>% Live Removed</th>
<th>% Dead Removed</th>
<th>% Live To Residue</th>
<th>% Dead To Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harvest</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Cut</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>Prune</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Graze</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

### 3.5.13 SpecificRootLength

\[
SpecificRootLength = 40 \text{ (m/g)}
\]

### 3.5.14 KNO3

\[
KNO3 = 0.03
\]

### 3.5.15 KNH4

\[
KNH4 = 0.02
\]

### 3.5.16 NUptakeSWFactor

\[
NUptakeSWFactor \text{ is calculated using linear interpolation}
\]

\[
\begin{array}{c|c}
X & Y \\
0 & 0 \\
1 & 1 \\
\end{array}
\]

\[
XValue = \text{[Root].RWC}
\]

### 3.5.17 MaintenanceRespirationFunction

\[
MaintenanceRespirationFunction = 0
\]

### 3.5.18 CarbonConcentration

\[
CarbonConcentration = 0.4
\]

### 3.5.19 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

#### 3.5.19.1 Structural

\[
\text{Structural} = DMDemandFunction \times \text{StructuralFraction}
\]

Where:
3.5.19.1.1 DM Demand Function

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

Where:

3.5.19.1.1.1 Partition Fraction

\[ PartitionFraction \] is calculated using linear interpolation

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.99</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>0.2</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ XValue = [Phenology].Stage \]

\[ StructuralFraction = 1 \]

3.5.19.2 Metabolic

\[ Metabolic = 0 \text{ (g/m2)} \]

3.5.19.3 Storage

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

3.5.19.3.1 Storage Fraction

\[ StorageFraction = 1 - [Root].DMDemands.Structural.StructuralFraction \]

3.5.20 N Demands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.5.20.1 Structural

\[ Structural = [Root].minimumNconc \times [Root].potentialDMAllocation.Structural \]

3.5.20.2 Metabolic

\[ Metabolic = MetabolicNconc \times [Root].potentialDMAllocation.Structural \]

Where:

3.5.20.2.1 Metabolic N Conc

\[ MetabolicNconc = [Root].criticalNConc - [Root].minimumNconc \]

3.5.20.3 Storage

The partitioning of daily growth to storage biomass attempts to bring the organ’s N content to the maximum concentration.

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

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

\[ CriticalNConc = [Root].MinimumNConc \]
3.5.21 InitialWt

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.5.21.1 Structural

Structural = 0.2 (g/plant)

3.5.21.2 Metabolic

Metabolic = 0 (g/m²)

3.5.21.3 Storage

Storage = 0 (g/m²)

3.5.22 RootDepthStressFactor

RootDepthStressFactor = 1

3.6 Leaf

The leaves are modelled as a set of leaf cohorts and the properties of each of these cohorts are summed to give overall values for the leaf organ. A cohort represents all the leaves of a given main-stem node position including all of the branch leaves appearing at the same time as the given main-stem leaf (Lawless et al., 2005). The number of leaves in each cohort is the product of the number of plants per m² and the number of branches per plant. The Structure class models the appearance of main-stem leaves and branches. Once cohorts are initiated the Leaf class models the area and biomass dynamics of each. It is assumed all the leaves in each cohort have the same size and biomass properties. The modelling of the status and function of individual cohorts is delegated to LeafCohort classes.

3.6.1 Dry Matter Fixation

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

3.6.1.1 ExtinctionCoeff

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

ExtinctionCoeff = PotentialExtinctionCoeff × WaterStressImpact

Where:

3.6.1.1.1 PotentialExtinctionCoeff

PotentialExtinctionCoeff is calculated using linear interpolation

\[
\begin{array}{c|c|c|c}
X & Y \\
0.2 & 0.7 \\
0.5 & 0.5 \\
1 & 0.4 \\
\end{array}
\]

\[
XValue = [Maize].SowingData.RowSpacing
\]

3.6.1.1.2 WaterStressImpact

WaterStressImpact is calculated using linear interpolation
3.6.1.2 Photosynthesis

Biomass fixation is modelled as the product of intercepted radiation and its conversion efficiency, the radiation use efficiency (RUE) (Monteith et al., 1977). This approach simulates net photosynthesis rather than providing separate estimates of growth and respiration. The potential photosynthesis calculated using RUE is then adjusted according to stress factors, these account for plant nutrition (FN), air temperature (FT), vapour pressure deficit (FVPD), water supply (FW) and atmospheric CO$_2$ concentration (FCO2). NOTE: RUE in this model is expressed as g/MJ for a whole plant basis, including both above and below ground growth.

\[ RUE = 2 \]

3.6.1.2.1 FT

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

\[ MaximumTemperatureWeighting = 0.75 \]

3.6.1.2.2 FVPD

\[ FVPD \] is calculated using linear interpolation

\[ XValue = [Leaf].Photosynthesis.VPD \]
3.6.1.2.3 FN

FN is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 0.3 \\
1 & 1 \\
\end{array}
\]

\[XValue = [Leaf].Fn\]

3.6.1.2.4 FW

FW is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 0 \\
0.3 & 1 \\
\end{array}
\]

\[XValue = [Leaf].Fw\]

3.6.1.2.5 FCO2

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

\[RadnInt = [Leaf].RadIntTot\]

3.6.2 CO2internal

\[CO2internal = (163 - [IWeather].MeanT)/(5 - 0.1x[IWeather].MeanT)\]

3.6.3 StomatalConductanceCO2Modifier

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

Where:

3.6.3.1 RelativeCO2Gradient

\[RelativeCO2Gradient = ([IWeather].CO2 - [Leaf].CO2internal)/(350 - [Leaf].CO2internal)\]

Area = 380

Area = 0

3.6.4 Potential Leaf Area index

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

GrowthDuration is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 1 \\
0.3 & 200 \\
0.5 & 250 \\
1 & 100 \\
\end{array}
\]

\[X Value = [Structure].RelativeNodeAppearance\]

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

3.6.4.2 MaxArea

Calculates the maximum leaf size (mm²/leaf) given its node position (Elings, 2000 - Agronomy Journal 92, 436-444)

Potential leaf area is calculated from leaf number using the approach of Keating and Wafula (1992). Individual leaf size is described using a bell-shaped curve, the parameters of which are derived from final leaf number.

3.6.4.2.1 LargestLeafPosition

\[\text{LargestLeafPosition} = [\text{Structure}].\text{FinalLeafNumber} \times \text{RelativePositionOfLargestLeaf}\]

Where:

\[\text{RelativePositionOfLargestLeaf} = 0.67\]

3.6.4.2.2 AreaMax

\[\text{AreaMax} \text{ is calculated using linear interpolation}\]

\[
\begin{array}{c|c}
X & Y \\
0 & 32530 \\
2 & 39259 \\
7 & 49659 \\
12 & 60000 \\
17 & 75000 \\
22 & 90000 \\
\end{array}
\]

\[X Value = [\text{Structure}].\text{FinalLeafNumber}\]

3.6.4.2.3 Breadth

\[\text{Breadth} \text{ is calculated using linear interpolation}\]
\[ X_{\text{Value}} = \text{[Structure].FinalLeafNumber} \]

Skewness = 0.00025

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

3.6.4.3 LagDuration

LagDuration is calculated using linear interpolation

\[ X \quad Y \]
\[ 0 \quad 300 \]
\[ 0.5 \quad 300 \]
\[ 0.7 \quad 700 \]
\[ 0.8 \quad 900 \]
\[ 1 \quad 1300 \]

\[ X_{\text{Value}} = \text{[Structure].RelativeNodeAppearance} \]

3.6.4.4 SenescenceDuration

SenescenceDuration is calculated using linear interpolation

\[ X \quad Y \]
\[ 0 \quad 150 \]
\[ 0.3 \quad 150 \]
\[ 0.5 \quad 350 \]

\[ X_{\text{Value}} = \text{[Structure].RelativeNodeAppearance} \]

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:

3.6.4.5 ShadeInducedSenescenceRate

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

### 3.6.5.1 CellDivisionStress

CellDivisionStress is calculated using linear interpolation

\[
\text{CellDivisionStress} = \min(\text{WaterStressEffect}, \text{TemperatureEffect}, \text{NitrogenStressEffect})
\]

Where:

### 3.6.5.2 ExpansionStress

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

Where:

### 3.6.5.2.1 WaterStressEffect

WaterStressEffect is calculated using linear interpolation

\[
\text{Leaf.Fw} = \text{[Maize].Leaf.Fw}
\]

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

Leaf.FW quantifies water stress and is calculated as \(\text{Leaf.Transpiration}/\text{Leaf.WaterDemand}\), 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.
Stresses can also accelerate the onset and rate of senescence in a number of ways. Nitrogen shortage will cause N to be retranslocated out of lower order leaves to support the expansion of higher order leaves and other organs. When this happens, the lower order cohorts will have their area reduced in proportion to the amount of N that is remobilised out of them.

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

**3.6.5.3 DroughtInducedLagAcceleration**

\[ DroughtInducedLagAcceleration = 1 + Stress \]
Where:

3.6.5.3.1 Stress

\[ Stress = StressResponseCoefficient \times StressFactor \]

Where:

\[ StressResponseCoefficient = 2 \]

3.6.5.3.1.1 StressFactor

\[ StressFactor \] is calculated using linear interpolation

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ XValue = \text{[Maize].Leaf.Fw} \]

3.6.5.4 DroughtInducedSenAcceleration

\[ DroughtInducedSenAcceleration = 1 + Stress \]

Where:

3.6.5.4.1 Stress

\[ Stress = StressResponseCoefficient \times StressFactor \]

Where:

\[ StressResponseCoefficient = 0 \]

3.6.5.4.1.1 StressFactor

\[ StressFactor \] is calculated using linear interpolation

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ XValue = \text{[Root].WaterTensionFactor} \]

3.6.6 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 \( \text{DeltaPotentialArea} \) divided by the mean of SpecificLeafAreaMax and SpecificLeafAreaMin. Under stressed conditions it is calculated as \( \text{DeltaWaterConstrainedArea} \) divided by SpecificLeafAreaMin.
3.6.6.1 SpecificLeafAreaMax

SpecificLeafAreaMax is calculated using linear interpolation

\[
X \quad Y \\
1 \quad 30000 \\
9 \quad 20000
\]

\[XValue = [Structure].LeafTipsAppeared\]

3.6.6.2 SpecificLeafAreaMin

SpecificLeafAreaMin is calculated using linear interpolation

\[
X \quad Y \\
1 \quad 25000 \\
9 \quad 15000
\]

\[XValue = [Structure].LeafTips Appeared\]

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

3.6.6.3 StructuralFraction

StructuralFraction = 0.3

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

Unknown child name: NonStructuralFraction

3.6.7 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:

3.6.7.1 MinimumNConc

MinimumNConc = 0.005 (g/g)

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

3.6.7.2 CriticalNConc

CriticalNConc = 0.01 (g/g)

Storage N demand is calculated as the sum of metabolic and structural wt (including today’s demands)
multiplied by \( \text{LuxaryNconc} \) \((\text{MaximumNConc} - \text{CriticalNConc})\) less the amount of storage N already present. \( \text{MaximumNConc} \) is given by:

### 3.6.7.3 MaximumNConc

\[
\text{MaximumNConc} = \text{Leaf}.\text{CohortParameters}.\text{CriticalNConc} + \text{StorageNConc}
\]

Where:

### 3.6.7.3.1 StorageNConc

\[
\text{StorageNConc} = \text{NonStructuralNConc} \times \text{WaterStressEffect}
\]

Where:

### 3.6.7.3.1.1 NonStructuralNConc

\( \text{NonStructuralNConc} \) is calculated using linear interpolation

\[
\begin{array}{|c|c|}
\hline
X & Y \\
1 & 0.034 \\
10 & 0.01 \\
\hline
\end{array}
\]

\( X \) Value = \([\text{Structure}].\text{LeafTipsAppeared} \)

### 3.6.7.3.1.2 WaterStressEffect

\( \text{WaterStressEffect} \) is calculated using linear interpolation

\[
\begin{array}{|c|c|}
\hline
X & Y \\
0.5 & 0 \\
0.95 & 0 \\
1 & 1 \\
\hline
\end{array}
\]

\( X \) Value = \([\text{Root}].\text{WaterTensionFactor} \)

### 3.6.8 Drymatter supply

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

#### 3.6.8.1 DMReallocationFactor

\[
\text{DMReallocationFactor} = 1
\]

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

DMRetranslocationFactor = 0.5

3.6.9 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:

3.6.9.1 NReallocationFactor

NReallocationFactor = 1

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

3.6.9.2 NRetranslocationFactor

NRetranslocationFactor = 0

3.6.9.3 RemobilisationCost

RemobilisationCost = 0

3.6.9.4 LeafSizeShapeParameter

LeafSizeShapeParameter = 0.01

3.6.9.5 MaintenanceRespirationFunction

MaintenanceRespirationFunction = 0

3.6.9.6 SenessingLeafRelativeSize

SenessingLeafRelativeSize = 1 (0-1)

3.6.9.7 DetachmentLagDuration

DetachmentLagDuration = 1000000 (deg day)

3.6.9.8 DetachmentDuration

DetachmentDuration = 1000000 (deg day)

3.6.9.9 StorageFraction

StorageFraction = 0.1

3.6.9.10 InitialNConc

InitialNConc = 0 (g/g)

3.6.9.11 LagDurationAgeMultiplier

LagDurationAgeMultiplier = 1 1 1

3.6.9.12 SenescenceDurationAgeMultiplier

SenescenceDurationAgeMultiplier = 1 1 1

3.6.9.13 LeafSizeAgeMultiplier

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

3.6.10 FrostFraction

FrostFraction is calculated using linear interpolation
\[ XValue = \text{[Weather].MinT} \]

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

### 3.6.11 RemobilisationCost

\[ \text{RemobilisationCost} = 0 \]

### 3.6.12 FRGRFunction

\[ \text{FRGRFunction} = \text{minimum}(FT, FN, FVPD) \]

Where:

\[ FT = \text{[Leaf].Photosynthesis.FT} \]

\[ FN = \text{[Leaf].Photosynthesis.FN} \]

\[ FVPD = \text{[Leaf].Photosynthesis.FVPD} \]

### 3.6.13 StructuralFraction

\[ \text{StructuralFraction} = 0.7 \]

### 3.6.14 DMConversionEfficiency

\[ \text{DMConversionEfficiency} = 1 \text{ (0-1)} \]

### 3.6.15 BiomassRemovalDefaults

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

<table>
<thead>
<tr>
<th>Method</th>
<th>% Live Removed</th>
<th>% Dead Removed</th>
<th>% Live To Residue</th>
<th>% Dead To Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harvest</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>Cut</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prune</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>Graze</td>
<td>60</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

### 3.6.16 CarbonConcentration

\[ \text{CarbonConcentration} = 0.4 \]
3.6.17 WidthFunction

\[ \text{WidthFunction} = 0 \]

\[ \text{DepthFunction} = \text{[Leaf].Height} \]

3.7 Husk

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

3.7.1 Dry Matter Demand

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

3.7.1.1 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.7.1.1.1 Structural

\[ \text{Structural} = \text{DMDemandFunction} \times \text{StructuralFraction} \]

Where:

3.7.1.2 DMDemandFunction

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

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

\[ \text{StartStage} = 5.5 \]

Stage when organ growth starts

\[ \text{GrowthDuration} = 300 \, (^{\circ} \text{Cd}) \]

ThermalTime duration of organ growth

\[ \text{MaximumOrganWt} = 25 \, (g) \]

Size individual organs will grow to when fully supplied with DM

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

\[ \text{SenescenceRate} = 0 \, (/\text{d}) \]

3.7.1.2.1 ExpansionStress2

\[ \text{ExpansionStress2} \] is calculated using linear interpolation

\[
\begin{array}{c|c}
        X & Y \\
\hline
        0  & 0 \\
        0.35 & 1 \\
\end{array}
\]

\[ X_{\text{Value}} = \text{[Maize].Leaf.Fw} \]
3.7.1.2 ExpansionStress

ExpansionStress = minimum (WaterStressEffect, TemperatureEffect, NitrogenStressEffect)

Where:

3.7.1.2.1 WaterStressEffect

WaterStressEffect is calculated using linear interpolation

\[
\begin{array}{c|c|c}
X & Y \\
0.1 & 0 \\
0.7 & 1 \\
1.3 & 1 \\
\end{array}
\]

\[XValue = [Root].WaterTensionFactor\]

3.7.1.2.2 TemperatureEffect

TemperatureEffect is calculated using linear interpolation

\[
\begin{array}{c|c|c}
X & Y \\
0 & 0 \\
12 & 1 \\
14 & 1 \\
\end{array}
\]

\[XValue = [Weather].MeanT\]

3.7.1.2.3 NitrogenStressEffect

NitrogenStressEffect is calculated using linear interpolation

\[
\begin{array}{c|c|c}
X & Y \\
0 & 0.1 \\
0.5 & 0.1 \\
1 & 1 \\
\end{array}
\]

\[XValue = [Maize].Leaf.Fn\]

3.7.1.2.4 StructuralFraction

The value of StructuralFraction from Emergence to StartGrainFill is calculated as follows:
StemFraction = 0.5

The value of StructuralFraction from StartGrainFill to HarvestRipe is calculated as follows:

\[ \text{StemFraction} = 1 \]

StructuralFraction has a value of zero for phases not specified above

3.7.1.2.3 Metabolic

\[ \text{Metabolic} = 0 \ (\text{g/m}^2) \]

3.7.1.2.4 Storage

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

3.7.1.2.4.1 StorageFraction

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

3.7.2 Nitrogen Demand

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

3.7.2.1 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.7.2.1.1 Structural

\[ \text{Structural} = [\text{Husk}].\text{minimumNconc} \times [\text{Husk}].\text{potentialDMAllocation}.\text{Structural} \]

3.7.2.1.2 Metabolic

\[ \text{Metabolic} = \text{MetabolicNconc} \times [\text{Husk}].\text{potentialDMAllocation}.\text{Structural} \]

Where:

3.7.2.1.2.1 MetabolicNconc

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

3.7.2.1.3 Storage

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

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

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

3.7.2.2 MinimumNConc

\[ \text{MinimumNConc} = 0.005 \ (\text{g/g}) \]

\[ \text{CriticalNConc} = [\text{Husk}].\text{MinimumNConc} \]

3.7.2.3 MaximumNConc

\[ \text{MaximumNConc} = 0.007 \ (\text{g/g}) \]

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

3.7.2.4 NitrogenDemandSwitch

1 between Emergence and StartGrainFill and a value of zero outside of this period

3.7.3 Dry Matter Supply

Husk does not reallocate DM when senescence of the organ occurs.
Husk will retranslocate 10% of non-structural DM each day.

3.7.4 Nitrogen Supply

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

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

3.7.5 Senescence and Detachment

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

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

3.7.6 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 surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Live Removed</th>
<th>% Dead Removed</th>
<th>% Live To Residue</th>
<th>% Dead To Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harvest</td>
<td>50</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Cut</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prune</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>Graze</td>
<td>60</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

3.8 Rachis

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

3.8.1 Dry Matter Demand

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

3.8.1.1 DMDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.8.1.1.1 Structural

\[ \text{Structural} = \text{DMDemandFunction} \times \text{StructuralFraction} \]

Where:

3.8.1.2 DMDemandFunction

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

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

\[ \text{StartStage} = 5.5 \]

Stage when organ growth starts

\[ \text{GrowthDuration} = 300 \text{ (°Cd)} \]

ThermalTime duration of organ growth
Maximum Organ Wt = 23 (g)

Size individual organs will grow to when fully supplied with DM

Organ Population = [Rachis].CobNumber

3.8.1.2 Expansion Stress

Expansion Stress = minimum (Water Stress Effect, Temperature Effect, Nitrogen Stress Effect)

Where:

3.8.1.2.1 Water Stress Effect

Water Stress Effect is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0.1 & 0 \\
0.7 & 1 \\
1.3 & 1 \\
\end{array}
\]

\[X \text{Value} = [\text{Root}].\text{Water Tension Factor}\]

3.8.1.2.2 Temperature Effect

Temperature Effect is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 0 \\
12 & 1 \\
14 & 1 \\
\end{array}
\]

\[X \text{Value} = [\text{Weather}].\text{Mean T}\]

3.8.1.2.3 Nitrogen Stress Effect

Nitrogen Stress Effect is calculated using linear interpolation

\[
\begin{array}{c|c}
X & Y \\
0 & 0.1 \\
0.5 & 0.1 \\
1 & 1 \\
\end{array}
\]
3.8.1.2.2 Metabolic

Metabolic = 0 (g/m2)

3.8.1.2.3 Storage

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

3.8.1.2.3.1 StorageFraction

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

3.8.2 Nitrogen Demand

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

3.8.2.1 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.8.2.1.1 Structural

Structural = [Rachis].minimumNconc × [Rachis].potentialDMAllocation.Structural

3.8.2.1.2 Metabolic

Metabolic = MetabolicNconc × [Rachis].potentialDMAllocation.Structural

Where:

3.8.2.1.2.1 MetabolicNconc

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

3.8.2.1.3 Storage

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

Storage = [Rachis].maximumNconc × ([Rachis].Live.Wt + potentialAllocationWt) - [Rachis].Live.N

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

3.8.2.2 MinimumNConc

MinimumNConc = 0.003 (g/g)

CriticalNConc = [Rachis].MinimumNConc

3.8.2.3 MaximumNConc

MaximumNConc = 0.008 (g/g)

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

3.8.2.4 NitrogenDemandSwitch

1 between Emergence and StartGrainFill and a value of zero outside of this period

3.8.3 Dry Matter Supply

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

Rachis does not retranslocate non-structural DM.
3.8.4 Nitrogen Supply
Rachis does not reallocate N when senescence of the organ occurs.
Rachis will retranslocate 10% of non-structural N each day.

3.8.5 Senescence and Detachment
Rachis has senescence parameterised to zero so all biomass in this organ will remain alive.
Rachis has detachment parameterised to zero so all biomass in this organ will remain with the plant until a
defoliation or harvest event occurs.

3.8.6 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 surface residues. The following table describes the default
proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil
surface residue using "To Residue" for a range of management actions. The total percentage removed for live
do not exceed 100%. The difference between the total and 100% gives the biomass remaining on the
plant. These can be changed during a simulation using a manager script.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Live Removed</th>
<th>% Dead Removed</th>
<th>% Live To Residue</th>
<th>% Dead To Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harvest</td>
<td>50</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Cut</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prune</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>Graze</td>
<td>60</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

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

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

3.9.1.1 DMDemands
This is the collection of functions for calculating the demands for each of the biomass pools (Structural,
Metabolic, and Storage).

3.9.1.1.1 Structural
\[ Structural = DMDemandFunction \times StructuralFraction \]

Where:

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

Where:

PartitionFraction is calculated using linear interpolation
XValue = [Phenology].Stage

3.9.1.1.2 StructuralFraction

The value of StructuralFraction from Emergence to StartGrainFill is calculated as follows:

\[ \text{StemFraction} = 0.5 \]

The value of StructuralFraction from StartGrainFill to HarvestRipe is calculated as follows:

\[ \text{StemFraction} = 1 \]

StructuralFraction has a value of zero for phases not specified above

3.9.1.1.2 Metabolic

\[ \text{Metabolic} = 0 \text{ (g/m2)} \]

3.9.1.1.3 Storage

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

3.9.1.1.3.1 StorageFraction

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

3.9.2 Nitrogen Demand

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

3.9.2.1 NDemands

This is the collection of functions for calculating the demands for each of the biomass pools (Structural, Metabolic, and Storage).

3.9.2.1.1 Structural

\[ \text{Structural} = [\text{Stem}].\text{minimumNconc} \times [\text{Stem}].\text{potentialDMAllocation}.\text{Structural} \]

3.9.2.1.2 Metabolic

\[ \text{Metabolic} = \text{MetabolicNconc} \times [\text{Stem}].\text{potentialDMAllocation}.\text{Structural} \]

Where:

3.9.2.1.2.1 MetabolicNconc

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

3.9.2.1.3 Storage

The partitioning of daily growth to storage biomass attempts to bring the organ’s N content to the maximum concentration.

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

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

MinimumNConc = 0.004 (g/g)

3.9.2.3 CriticalNConc

CriticalNConc = [Stem].MinimumNConc + MetabolicNConc

Where:

3.9.2.3.1 MetabolicNConc

MetabolicNConc = NonStructuralNConc × Proportion

Where:

3.9.2.3.1.1 NonStructuralNConc

NonStructuralNConc = [Stem].MaximumNConc - [Stem].MinimumNConc

Proportion = 0.2

3.9.2.4 MaximumNConc

MaximumNConc is calculated using linear interpolation

<table>
<thead>
<tr>
<th>XValue</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.025</td>
</tr>
<tr>
<td>0.7</td>
<td>0.025</td>
</tr>
<tr>
<td>1</td>
<td>0.005</td>
</tr>
</tbody>
</table>

XValue = [Structure].RelativeNodeAppearance

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

3.9.2.5 NitrogenDemandSwitch

1 between Emergence and StartGrainFill and a value of zero outside of this period

3.9.3 Dry Matter Supply

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

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

3.9.4 Nitrogen Supply

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

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

3.9.4.1 NRetranslocationFactor

3.9.4.1.1 PreStemElongation

The value of NRetranslocationFactor from Emergence to FlagLeaf is calculated as follows:

StemFraction = 0

3.9.4.1.2 StemElongation

The value of NRetranslocationFactor from FlagLeaf to HarvestRipe is calculated as follows:
**3.9.5 Senescence and Detachment**

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

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

**3.9.6 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 surface residues. The following table describes the default proportions of live and dead biomass that are transferred out of the simulation using "Removed" or to soil surface residue using "To Residue" for a range of management actions. The total percentage removed for live or dead must not exceed 100%. The difference between the total and 100% gives the biomass remaining on the plant. These can be changed during a simulation using a manager script.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Live Removed</th>
<th>% Dead Removed</th>
<th>% Live To Residue</th>
<th>% Dead To Residue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harvest</td>
<td>50</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Cut</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Prune</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>Graze</td>
<td>60</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

**3.10 AboveGround Biomass**

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

The AboveGround composite biomass object includes all organs occurring above the soil surface to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

AboveGround summarises the following biomass objects:

- [Leaf].Live
- [Leaf].Dead
- [Stem].Live
- [Stem].Dead
- [Husk].Live
- [Husk].Dead
- [Grain].Live
- [Grain].Dead
- [Rachis].Live
- [Rachis].Dead

**3.11 BelowGround Biomass**

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

The BelowGround composite biomass object includes all root biomass pools to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

BelowGround summarises the following biomass objects:

- [Root].Live
- [Root].Dead

**3.12 Total Biomass**

This is a composite biomass class, representing the sum of 1 or more biomass objects.
The Total composite biomass object includes all organs (above+below ground) to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

Total summarises the following biomass objects:

- [Leaf].Live
- [Leaf].Dead
- [Stem].Live
- [Stem].Dead
- [Husk].Live
- [Husk].Dead
- [Grain].Live
- [Grain].Dead
- [Rachis].Live
- [Rachis].Dead
- [Root].Live
- [Root].Dead

### 3.13 TotalLive Biomass

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

The TotalLive composite biomass object includes the Live pools of all organs (above+below ground) to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

TotalLive summarises the following biomass objects:

- [Leaf].Live
- [Stem].Live
- [Husk].Live
- [Grain].Live
- [Rachis].Live
- [Root].Live

### 3.14 TotalDead Biomass

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

The TotalDead composite biomass object includes the Dead pools of all organs (above+below ground) to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

TotalDead summarises the following biomass objects:

- [Leaf].Dead
- [Stem].Dead
- [Husk].Dead
- [Grain].Dead
- [Rachis].Dead
- [Root].Dead

### 3.15 EarLive Biomass

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

The EarLive composite biomass object includes the Live pools of all organs within the Ear to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

EarLive summarises the following biomass objects:

- [Husk].Live
- [Grain].Live
- [Rachis].Live

### 3.16 AboveGroundLive Biomass

This is a composite biomass class, representing the sum of 1 or more biomass objects.
The AboveGroundLive composite biomass object includes the Live pools of all organs occurring above the soil surface to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

AboveGroundLive summarises the following biomass objects:
- [Leaf].Live
- [Stem].Live
- [Husk].Live
- [Grain].Live
- [Rachis].Live

3.17 AboveGroundDead Biomass

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

The AboveGroundDead composite biomass object includes the Dead pools of all organs occurring above the soil surface to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

AboveGroundDead summarises the following biomass objects:
- [Leaf].Dead
- [Stem].Dead
- [Husk].Dead
- [Grain].Dead
- [Rachis].Dead

3.18 Future Development requirements

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

3.19 CultivarFolder

Atrium, CG4141, Dekalb_XL82, GH_5009, GH_5019WX, Hycorn_40, Hycorn_53, Katumani, Laila, Makueni, malawi_local, Melkassa, mh12, mh16, mh17, mh18, mh19, NSCM_41, Pioneer_3153, Pioneer_33M54, Pioneer_34K77, Pioneer_38H20, Pioneer_39G12, Pioneer_39V43, r201, r215, sc401, sc501, sc601, sc623, sc625, sr52

3.19.1 Hycorn_53

This cultivar is defined by overriding some of the base parameters of the plant model.

Hycorn_53 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 190
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 520
[Grain].MaximumGrainsPerCob.FixedValue = 550

3.19.2 Pioneer_33M54

This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_33M54 makes the following changes:
3.19.3 Pioneer_38H20
This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_38H20 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 210
[Phenology].GrainFilling.Target.FixedValue = 550

3.19.4 Pioneer_34K77
This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_34K77 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 280
[Phenology].GrainFilling.Target.FixedValue = 550

3.19.5 Pioneer_39V43
This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_39V43 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 100
[Phenology].GrainFilling.Target.FixedValue = 500
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.Y = 26, 26, 46, 46, 70
[Structure].HeightModel.PotentialHeight.XYPairs.X = 3, 4, 5, 6
[Structure].HeightModel.PotentialHeight.XYPairs.Y = 20, 40, 50, 1300

3.19.6 Atrium
This cultivar is defined by overriding some of the base parameters of the plant model.

Atrium makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 150
[Phenology].GrainFilling.Target.FixedValue = 500
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Structure].Phyllochron.Phylochron.Phylochron.XYPairs.Y = 26, 26, 46, 46, 70
[Structure].HeightModel.PotentialHeight.XYPairs.X = 3, 4, 5, 6
[Structure].HeightModel.PotentialHeight.XYPairs.Y = 20, 40, 50, 2370

3.19.7 Laila
This cultivar is defined by overriding some of the base parameters of the plant model.

Laila makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 211
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 0
[Phenology].FlagLeafToFlowering.Target.FixedValue = 1
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 812
[Phenology].Maturing.Target.FixedValue = 1
[Phenology].MaturityToHarvestRipe.Target.FixedValue = 1
[Grain].MaximumGrainsPerCob.FixedValue = 720
3.19.8 GH_5019WX

This cultivar is defined by overriding some of the base parameters of the plant model.

GH_5019WX makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 225
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24.0
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124
[Phenology].FlagLeafToFlowering.Target.FixedValue = 101
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 694
[Grain].MaximumGrainsPerCob.FixedValue = 550

3.19.9 Hycorn_40

This cultivar is defined by overriding some of the base parameters of the plant model.

Hycorn_40 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 190
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 124
[Phenology].FlagLeafToFlowering.Target.FixedValue = 50
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 713
[Grain].MaximumGrainsPerCob.FixedValue = 550

3.19.10 GH_5009

This cultivar is defined by overriding some of the base parameters of the plant model.

GH_5009 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 190
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 300
[Phenology].FlagLeafToFlowering.Target.FixedValue = 50
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 713
[Grain].MaximumGrainsPerCob.FixedValue = 550

3.19.11 Dekalb_XL82

This cultivar is defined by overriding some of the base parameters of the plant model.

Dekalb_XL82 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 211
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 472
[Phenology].FlagLeafToFlowering.Target.FixedValue = 101
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 813
[Grain].MaximumGrainsPerCob.FixedValue = 550

3.19.12 malawi_local

This cultivar is defined by overriding some of the base parameters of the plant model.

malawi_local makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 280
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
This cultivar is defined by overriding some of the base parameters of the plant model.

mh19 makes the following changes:

- **Phenology**.Juvenile.Target.FixedValue = 290
- **Phenology**.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- **Phenology**.Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
- **Phenology**.FlagLeafToFlowering.Target.FixedValue = 10
- **Phenology**.FloweringToGrainFilling.Target.FixedValue = 170
- **Phenology**.GrainFilling.Target.FixedValue = 570
- **Grain**.MaximumGrainsPerCob.FixedValue = 504
- **Grain**.MaximumPotentialGrainSize.FixedValue = 0.361

mh17 makes the following changes:

- **Phenology**.Juvenile.Target.FixedValue = 290
- **Phenology**.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- **Phenology**.Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
- **Phenology**.FlagLeafToFlowering.Target.FixedValue = 10
- **Phenology**.FloweringToGrainFilling.Target.FixedValue = 170
- **Phenology**.GrainFilling.Target.FixedValue = 570
- **Grain**.MaximumGrainsPerCob.FixedValue = 504
- **Grain**.MaximumPotentialGrainSize.FixedValue = 0.361

mh16 makes the following changes:

- **Phenology**.Juvenile.Target.FixedValue = 290
- **Phenology**.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- **Phenology**.Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
- **Phenology**.FlagLeafToFlowering.Target.FixedValue = 10
- **Phenology**.FloweringToGrainFilling.Target.FixedValue = 170
- **Phenology**.GrainFilling.Target.FixedValue = 570
- **Grain**.MaximumGrainsPerCob.FixedValue = 504
- **Grain**.MaximumPotentialGrainSize.FixedValue = 0.361

mh12 makes the following changes:

- **Phenology**.Juvenile.Target.FixedValue = 290
- **Phenology**.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- **Phenology**.Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
- **Phenology**.FlagLeafToFlowering.Target.FixedValue = 10
- **Phenology**.FloweringToGrainFilling.Target.FixedValue = 170
- **Phenology**.GrainFilling.Target.FixedValue = 570
- **Grain**.MaximumGrainsPerCob.FixedValue = 504
- **Grain**.MaximumPotentialGrainSize.FixedValue = 0.361
This cultivar is defined by overriding some of the base parameters of the plant model.

sc623 makes the following changes:

```
[Phenology].Juvenile.Target.FixedValue = 280
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 550
[Grain].MaximumGrainsPerCob.FixedValue = 504
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361
```

This cultivar is defined by overriding some of the base parameters of the plant model.

sc625 makes the following changes:

```
[Phenology].Juvenile.Target.FixedValue = 280
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 570
[Grain].MaximumGrainsPerCob.FixedValue = 504
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361
```

This cultivar is defined by overriding some of the base parameters of the plant model.

sc601 makes the following changes:

```
[Phenology].Juvenile.Target.FixedValue = 280
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 560
[Grain].MaximumGrainsPerCob.FixedValue = 495
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361
```

This cultivar is defined by overriding some of the base parameters of the plant model.

CG4141 makes the following changes:

```
[Phenology].Juvenile.Target.FixedValue = 233
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 550
[Grain].MaximumGrainsPerCob.FixedValue = 468
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361
```

This cultivar is defined by overriding some of the base parameters of the plant model.

mh18 makes the following changes:
3.19.22 r215

This cultivar is defined by overriding some of the base parameters of the plant model.

r215 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 250
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 550
[Grain].MaximumGrainsPerCob.FixedValue = 468
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

3.19.23 Melkassa

This cultivar is defined by overriding some of the base parameters of the plant model.

Melkassa makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 230
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 160
[Phenology].GrainFilling.Target.FixedValue = 550
[Grain].MaximumGrainsPerCob.FixedValue = 387
[Grain].MaximumPotentialGrainSize.FixedValue = 0.33

3.19.24 sr52

This cultivar is defined by overriding some of the base parameters of the plant model.

sr52 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 250
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
[Phenology].GrainFilling.Target.FixedValue = 550
[Grain].MaximumGrainsPerCob.FixedValue = 468
[Grain].MaximumPotentialGrainSize.FixedValue = 0.361

3.19.25 sc501

This cultivar is defined by overriding some of the base parameters of the plant model.

sc501 makes the following changes:

[Phenology].Juvenile.Target.FixedValue = 250
[Phenology].Photosensitive.Target.XYPairs.X = 0, 12.5, 24
[Phenology].Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
[Phenology].FlagLeafToFlowering.Target.FixedValue = 10
[Phenology].FloweringToGrainFilling.Target.FixedValue = 170
This cultivar is defined by overriding some of the base parameters of the plant model.

r201 makes the following changes:

- Phenology.Juvenile.Target.FixedValue = 230
- Phenology.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- Phenology.Photosensitive.Target.XYPairs.Y = 0, 0, 264.5
- Phenology.FlagLeafToFlowering.Target.FixedValue = 10
- Phenology.FloweringToGrainFilling.Target.FixedValue = 170
- Phenology.GrainFilling.Target.FixedValue = 550
- Grain.MaximumGrainsPerCob.FixedValue = 450
- Grain.MaximumPotentialGrainSize.FixedValue = 0.375

This cultivar is defined by overriding some of the base parameters of the plant model.

sc401 makes the following changes:

- Phenology.Juvenile.Target.FixedValue = 230
- Phenology.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- Phenology.Photosensitive.Target.XYPairs.Y = 0, 0, 115
- Phenology.FlagLeafToFlowering.Target.FixedValue = 10
- Phenology.FloweringToGrainFilling.Target.FixedValue = 170
- Phenology.GrainFilling.Target.FixedValue = 550
- Grain.MaximumGrainsPerCob.FixedValue = 450
- Grain.MaximumPotentialGrainSize.FixedValue = 0.375

This cultivar is defined by overriding some of the base parameters of the plant model.

NSCM_41 makes the following changes:

- Phenology.Juvenile.Target.FixedValue = 230
- Phenology.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- Phenology.Photosensitive.Target.XYPairs.Y = 0, 0, 115
- Phenology.FlagLeafToFlowering.Target.FixedValue = 10
- Phenology.FloweringToGrainFilling.Target.FixedValue = 170
- Phenology.GrainFilling.Target.FixedValue = 550
- Grain.MaximumGrainsPerCob.FixedValue = 450
- Grain.MaximumPotentialGrainSize.FixedValue = 0.375

This cultivar is defined by overriding some of the base parameters of the plant model.

Makueni makes the following changes:

- Phenology.Juvenile.Target.FixedValue = 150
- Phenology.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- Phenology.Photosensitive.Target.XYPairs.Y = 0, 0, 115
- Phenology.FlagLeafToFlowering.Target.FixedValue = 10
- Phenology.FloweringToGrainFilling.Target.FixedValue = 120
- Phenology.GrainFilling.Target.FixedValue = 530
- Grain.MaximumGrainsPerCob.FixedValue = 432
- Grain.MaximumPotentialGrainSize.FixedValue = 0.375
3.19.30 Katumani

This cultivar is defined by overriding some of the base parameters of the plant model.

Katumani makes the following changes:

- **[Phenology]**.Juvenile.Target.FixedValue = 150
- **[Phenology]**.Photosensitive.Target.XYPairs.X = 0, 12.5, 24
- **[Phenology]**.Photosensitive.Target.XYPairs.Y = 0, 0, 115
- **[Phenology]**.FlagLeafToFlowering.Target.FixedValue = 10
- **[Phenology]**.FloweringToGrainFilling.Target.FixedValue = 120
- **[Phenology]**.GrainFilling.Target.FixedValue = 530
- **[Grain]**.MaximumGrainsPerCob.FixedValue = 405
- **[Grain]**.MaximumPotentialGrainSize.FixedValue = 0.375

3.19.31 Pioneer_3153

This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_3153 makes the following changes:

3.19.32 Pioneer_39G12

This cultivar is defined by overriding some of the base parameters of the plant model.

Pioneer_39G12 makes the following changes:

- **[Phenology]**.Juvenile.Target.FixedValue = 180
- **[Phenology]**.GrainFilling.Target.FixedValue = 500
- **[Grain]**.MaximumGrainsPerCob.FixedValue = 550
- **[Grain]**.NumberFunction.GrowthRateFactor.XYPairs.X = 0.2, 2, 4, 10
- **[Grain]**.NumberFunction.GrowthRateFactor.XYPairs.Y = 0, 0.6, 1, 1
- **[Grain]**.MaximumPotentialGrainSize.FixedValue = 0.25

3.20 Spike Biomass

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

The SpikeLive composite biomass object includes the Live pools of then non-grain organs within the reproductive spike to allow output for total dry matter and Nutrient contents, including total structural, non-structural and metabolic pools.

Spike summarises the following biomass objects:

- **[Husk]**.Live
- **[Rachis]**.Live

3.21 MortalityRate

*MortalityRate* = 0

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

A replacements model

4 Validation

4.1 Combined Results

Simulation results for the combined datasets from the various countries are shown in the following graphs. The model is able to adequately capture the influence of growing conditions (soil, climate) and management (population, Nitrogen, irrigation, sowing date).
4.2 Africa

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

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

**List of experiments.**

<table>
<thead>
<tr>
<th>Experiment Name</th>
<th>Design (Number of Treatments)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMW2Katumani</td>
<td>(10)</td>
</tr>
<tr>
<td>JMW2Kiboko</td>
<td>(10)</td>
</tr>
<tr>
<td>BMW1</td>
<td>Site (24)</td>
</tr>
</tbody>
</table>
4.3 Australia

The data for two trials from near Katherine, Northern Territory, Australia, have been included from CARBERRY et al., 1989 and CARBERRY et al., 1991. These include two planting dates for the summer of 1983/1984 (labelled as K841, K842). At each planting date, cultivar Dekalb XL82 was sown at 4 populations (3, 5, 7, 9 plants/m²) under low and high irrigation conditions. The low irrigation treatments only included preliminary applications to ensure crop establishment.

List of experiments.

<table>
<thead>
<tr>
<th>Experiment Name</th>
<th>Design (Number of Treatments)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K841</td>
<td>(8)</td>
</tr>
<tr>
<td>K842</td>
<td>(8)</td>
</tr>
<tr>
<td>DRK1</td>
<td>Sowing (7)</td>
</tr>
<tr>
<td>DRK2</td>
<td>Sowing (7)</td>
</tr>
<tr>
<td>DRK3</td>
<td>Sowing (7)</td>
</tr>
<tr>
<td>Angelo98</td>
<td>Fert (3)</td>
</tr>
<tr>
<td>Angelo99</td>
<td>Fert (3)</td>
</tr>
<tr>
<td>Angelo00</td>
<td>Fert (3)</td>
</tr>
</tbody>
</table>
4.4 New Zealand

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

5 Sensibility Tests

List of experiments.

<table>
<thead>
<tr>
<th>Experiment Name</th>
<th>Design (Number of Treatments)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugesera</td>
<td>NRate (2)</td>
</tr>
</tbody>
</table>

5.1 Bugesera

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


KEATING, BA, WA FULA, BM, WATIKI, JM, 1992. DEVELOPMENT OF A MODELING CAPABILITY FOR MAIZE IN SEMIARID EASTERN KENYA. ACIAR PROCEEDINGS SERIES, 26-33.


