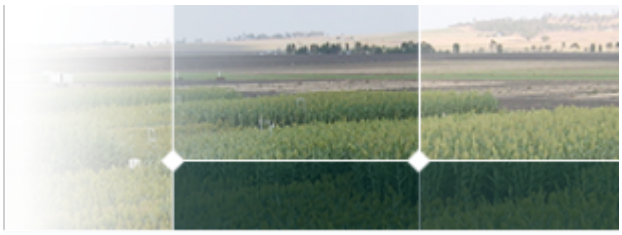


1 The APSIM Stock Model

1.1 The APSIM STOCK Model

Neville Herrmann (CSIRO)

Andrew Moore (CSIRO)

Eric Zurcher (CSIRO)

Dean Holzworth (AgResearch/CSIRO)

Mark Lieffering (AgResearch)

Val Snow (AgResearch)

1.2 Acknowledgements

CSIRO, AgResearch and the APSIM Initiative (https://www.apsim.info/) for funding the implemetation of the Stock model into APSIMX

1.3 Introduction

The Stock model is an implementation of the CSIRO's Australian Feeding Standards as expressed in the Grazplan (https://grazplan.csiro.au/) model. The technical document for Grazplan is available at available at https://grazplan.csiro.au/wp-content/uploads/2007/08/TechPaperMay12.pdf. The model has been ported into APSIM NextGeneration to allow full interaction with other models within the APSIM environment. Animals in Stock show the full range of growth, reproduction and death processes (see Grazplan) and the basic management actions (mating, weaning, moving, feeding supplements, selling and buying) have been implemented. All crop/plant models in APSIM NextGeneration have a 'damage' interface implemented and so allow grazing as soon as the Stock are moved to the same location (zone or paddock) as the crop.

Different animal species, genotypes and ages can be modelled together and various management systems implemented, including, but not limited to, forage crop grazing, rotational grazing, moving stock from paddock to feedlot and back again. Management can be simulated via Operations managers as in the simulations here or scripts. The collection of simulations in the Stock example shows examples of simulating three different management regimes (forage grazing, rotational grazing and a wheat/feedlot system) using scripts.

Management of animal groups is done by user-assigned tag values that take integer values. Tag values have two purposes:

* They can be used to manage distinct groups of animals in a common fashion. For example, all lactating ewes might be assigned the same tag value, and then all animals with this tag value might undergo the same supplementary feeding regime.
* If tag values are assigned sequentially (starting at 1), they can be used to generate summary variables. For example, WeightTag[1] gives the average live weight of all animals in groups with a tag value of 1.

Note that animal groups with different tag values are never merged, even if they are otherwise similar.

* To set the tag value of an animal group, use the Tag method.
* To determine the tag value of an animal group, use the TagNo variable.

Animal groups that become sufficiently similar are automatically merged into a single group.Animals are similar if all these criteria are the same:

* Occupy the same paddock
* Reproduction status (Castrated, Male, Empty, Early Preg, Late Preg)
* Number of foetuses
* Mating cycle (day in the mating cycle)

* Days to mating (Days left in joining period)
* Pregnancy (Days since conception)
* Lactation status (Days since parturition (if lactating)) – within 7 days
* Has (not) young
* If young exist, their reproductive status must be the same
* Implants (hormone implants)
* Mean age (if the animals are less than one year old)

These two simulations in this file are validations of the Stock model in APSIM Next Gen. A description of the validation dataset can found in the "Description" memo of the "Validation" node. Descriptions of the two validation simulations ("LUDF" and "StockSlurp") can be found in the memos of the respective simulations.

The parameters and variables that can be specified in the Stock model are in the auto-generated APSIM Next Gen documentation which can be found at https://apsimnextgeneration.netlify.app/ under the "Model documentation" link; the "Stock" model is near the bottom of the first table.

1.4 Changing Stock Parameters

1.5 Overview

The Stock model is an implementation of the CSIRO's Australian Feeding Standards as expressed in the Grazplan (https://grazplan.csiro.au/) model.

The technical paper describing Grazplan can be found at https://grazplan.csiro.au/wp-content/uploads/2007/08/TechPaperMay12.pdf

The structure of the stock parameter strings (which can be found after adding a "Genotype" model to the "Stock" node) is not user friendly but they are needed so the code stays aligned with Grazplan.

A useful resource to decipher the stock parameter strings are the Sheep and Cattle Explorer Excel spreadsheets. The Explorers are also useful to explore the effect of changing parameter values and they can be found at:

* https://grazplan.csiro.au/wp-content/uploads/2007/08/SheepExplorer.xlsx for sheep and
* https://grazplan.csiro.au/wp-content/uploads/2007/08/CattleExplorer.xlsx for cattle

1.6 Example

An example on how to change a parameter value is shown below. In the example, the Growth rate constant (CN1), which controls growth (and hence potential intake) will be changed from its default of 0.0115 to 0.015

1. in the Cattle Explorer Excel sheet, go to the "Pot.Intake" worksheet and confirm that the default Growth rate constant (CN1) is 0.0115 (cell B6). Note also any other numbers listed above or below the desired parameter - this is to double check you have the right value in the Stock GUI
2. in the genotype node (in the validation example "Friesian" under the main Stock node; this is where the animal's default parameters are defined) look for the appropriate name in the left hand column i.e. Growth
3. this name is followed by notation ("Growth C c-n-") in which the last two characters match that in the Explorer (CN)
4. ensure that the default value (0.0115) in the Cattle Explorer is found in the array in the right hand column along with the other numbers next to the default value of interest. This is just to ensure that you are dealing with the right parameter value
5. double click the array and change the default value to the new, desired value (0.015)
6. click away to save the changed value

1.7 Stock

The STOCK component encapsulates the GRAZPLAN animal biology model, as described in M Freer et al., 1997.

[The GrazPlan animal model technical description](https://grazplan.csiro.au/wp content/uploads/2007/08/TechPaperMay12.pdf)

Animals may be of different genotypes. In particular, sheep and cattle may be represented within a single STOCK instance.

Usually a single STOCK module is added to an AusFarm simulation, at the top level in the module hierarchy.

In a grazing system, however, there may be a variety of different classes of livestock. Animals may be of different genotypes (including both sheep and cattle); may be males, females or castrates; are likely to have a range of different ages; and females may be pregnant and/or lactating. The set of classes of livestock can change over time as animals enter or leave the system, are mated, give birth or are weaned. Further, animals that are otherwise similar may be placed in different paddocks, where their growth rates may differ.

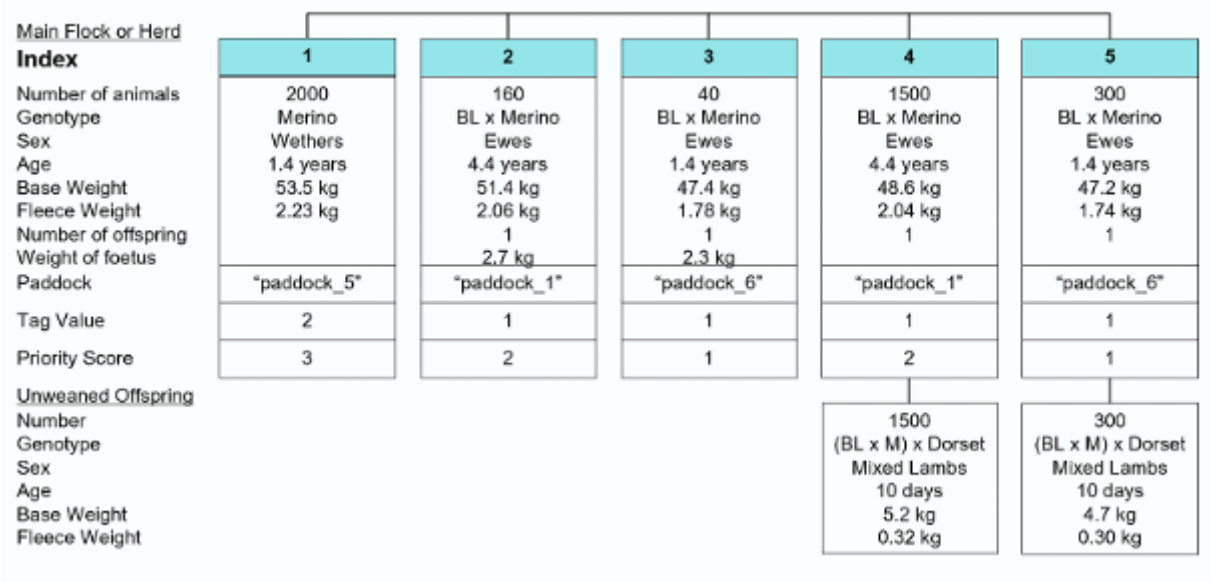


Figure 1: The list of animal groups at a particular time during a hypothetical simulation containing a STOCK module. Group 1 is distinct from the others because it has a different genotype and sex. Groups 2 and 3 are distinct because they are in different age classes (yearling vs mature). Groups 2 and 4 are distinct because they are in different reproductive states (pregnant vs lactating). Note how the unweaned lambs are associated with their mothers.

In the STOCK component, this complexity is handled by representing the set of animals in a simulated system as a list of animal groups (Figure 2.1). The members of each animal group have the same genotype and age class, but may have a range of ages (for example, an animal group containing mature animals may include four year old, five year old and six year old stock). The members of each animal group also have the same stage of pregnancy and/or lactation; the same number of suckling offspring; and occupy the same paddock.

The set of animal groups changes as animals enter and leave the simulation, and as physiological events such as maturation, mating, birth or weaning take place. Animal groups that become sufficiently similar are merged into a single group. The state of any unweaned lambs or calves is stored alongside that of their mothers; at weaning, the male and female weaners are transferred into two new animal groups within the main list.

In addition to the biological state variables that describe the animals, each animal group has four attributes that are of particular interest when writing management scripts.

Index

Each animal group has a unique, internally assigned integer index, starting at 1. Because the set of groups present in a component instance is dynamic, the index number associated with a particular group of animals can – and usually does – change over time. This dynamic numbering scheme has consequences for the way that animals of a particular kind must be located when writing management scripts.

Paddock

Each animal group is also assigned a paddock. The forage and supplementary feed available to a group of animals are determined by the paddock it occupies. Paddocks are referred to by name in the STOCK component:

* To set the paddock occupied by an animal group, use the **Move** event.
* To determine the paddock occupied by an animal group, use the **Paddock** variable.

It is the user's responsibility to ensure that paddock names correspond to PADDOCK modules or other sources of necessary driving variables.

Tag Value

Each animal group also has a user assigned tag value that takes an integer value. Tag values have two purposes:

* They can be used to manage distinct groups of animals in a common fashion. For example, all lactating ewes might be assigned the same tag value, and then all animals with this tag value might undergo the same supplementary feeding regime.
* If tag values are assigned sequentially (starting at 1), they can be used to generate summary variables. For example, **WeightTag[1]** gives the average live weight of all animals in groups with a tag value of 1.

Note that animal groups with different tag values are never merged, even if they are otherwise similar.

* To set the tag value of an animal group, use the **Tag** method.
* To determine the tag value of an animal group, use the **TagNo** variable.

Merging groups of similar animals

Animal groups that become sufficiently similar are merged into a single group. Animals are similar if all these are the same:

* Occupy the same paddock
* Reproduction status (Castrated, Male, Empty, Early Preg, Late Preg)
* Number of foetuses
* Mating cycle (day in the mating cycle)
* Days to mating (Days left in joining period)
* Pregnancy (Days since conception)
* Lactation status (Days since parturition (if lactating)) – within 7 days
* Has (not) young
* If young exist, their reproductive status must be the same
* Implants (hormone implants)
* Mean age (if the animals are less than one year old)

2 Validation

2.1 Validation Data Set Description

This vignette describes the origins and development of the Stock Model validation dataset. A high-level overview of the Stock Model itself can be found by clicking the Stock node.

The observed data set used for this validation simulation is based on publicly available data from New Zealand's Lincoln University Dairy Farm LUDF. The LUDF is a commercial demonstration dairy farm established in 2001 and operated by the South Island Dairying Development Centre (SIDDC) on behalf of Lincoln University to showcase sustainable and profitable dairy farming.

The LUDF is located at 1504 Shands Road, Lincoln (New Zealand; 43°38'S 172°26'E). The property is 186 ha of which 160 ha is the milking platform. The different soil types on the farm represent most of the common soil types in the surrounding Canterbury region. The farm operates in the top 2% of NZ dairy farms on profitability. The farm's targets and goals have varied over the years but in the 2019/20 season the target stocking rate was 3.5 cows/ha (peak milked), milk production of 1750 kg MS/ha (equivalent to 500kg MS/cow i.e. >100% liveweight in milk production), application of 160kg N/ha plus 300kg DM/cow imported supplement. Most cows are wintered off farm.

Average annual rainfall of 666 mm per annum is supplemented by an average irrigation input of 450 mm; average evapotranspiration for Lincoln is 870 mm/year. The milking platform was sown at conversion from a sheep operation (March 2001) in a mix of ryegrasses with white clovers, and a small amount of Timothy. The breed of cows at the LUDF are "KiwiCross" which was established as a separate breed in 2005 and is now New Zealand's most popular breed. In the 2019/20 season peak number of cows milked was 555 with the average days in milk of 282 days. The stocking rate of 3.5 cows/ha is equivalent to 1,665 kg liveweight/ha. In terms of feeding, in 2019/20 the cows ate 4.4 t DM/cow as pasture and 0.2 t DM/cow as supplement. Off-farm grazing was 0.7 t DM/cow giving a total feed intake of 5.4 t DM/cow.

Weekly farm data from the LUDF is available at http://www.siddc.org.nz/lu-dairy-farm/weekly-data/ as PDF reports. The data for the years 2004 to 2017 was transcribed and collated into an validation dataset.

Note that for these validation simulations, one stock parameter was changed from the original values in M Freer et al., 1997 to reflect modern dairy cow genetics. This, which can be accessed via the GUI in the "Friesian" node in the Stock model, was "Potential intake (Intake C c-i-[2]) which was increased from 0.025 to 0.04. Also note that "Dairy intake peak (c-idy-0)" was set to 1.

Acknowledgements

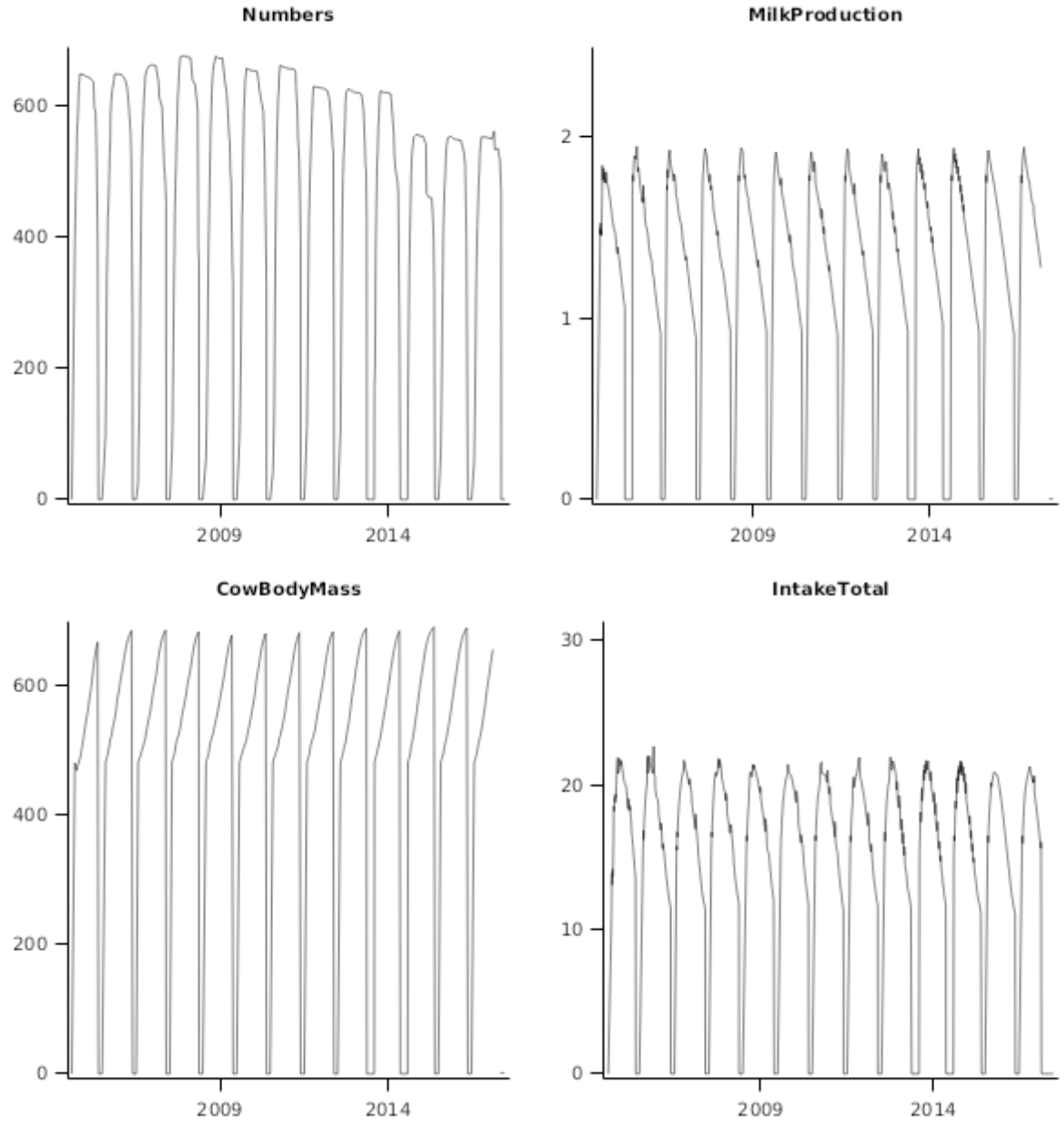
South Island Dairying Development Centre (SIDDC) and Lincoln University for making the dairy farm's production data available. Scott Rains, Samuel Dennis, Anna Taylor, Rogerio Cichota and Ronaldo Vibart for collating the LUDF data when working at AgResearch. Ron Pellow (SIDDC) for comments on the collated data set. David Pacheco and Ronaldo Vibart (AgResearch) for further discussions about the data.

2.2 LUDF

2.2.1 Feedlot simulation

The Feedlot simulation is a simple illustration of the Stock model by using animals entering a feedlot, being fed and milked and leaving the feedlot. In addition to the Stock model itself and the Supplement specification GUI, at its core is

the ChangeLactatingCowNumber GUI, the Stock operations and the Supplement operations. Both the operation files are based on stock and pasture/supplement data from the LUDF dataset (see the Validation memo for details). In the Stock operations, data on actual LUDF weekly changes in cow number have been collated from 1/08/2004 to 30/06/2017. Positive numbers denote lactating cows have been bought and moved to the feedlot while negative numbers are when cows are dried off and removed from the feedlot. It was assumed that cows were bought at weekly intervals. For the Supplement operations, six types of supplements labelled "silage_11me", "pasture_11.5me", "pasture_12me", "pasture_12.5me", "pasture_13me", and "pasture_13.5me" are bought. An excess amount of each supplement is bought (enough to last longer than the simulation) and their characteristics must be specified in the Supplement node. The LUDF dataseet is based on weekly reports - the daily pasture and supplements supplied (and their characteristics such as ME content) listed in the Supplement operations was extrapolated from the weekly data to the intervening days.



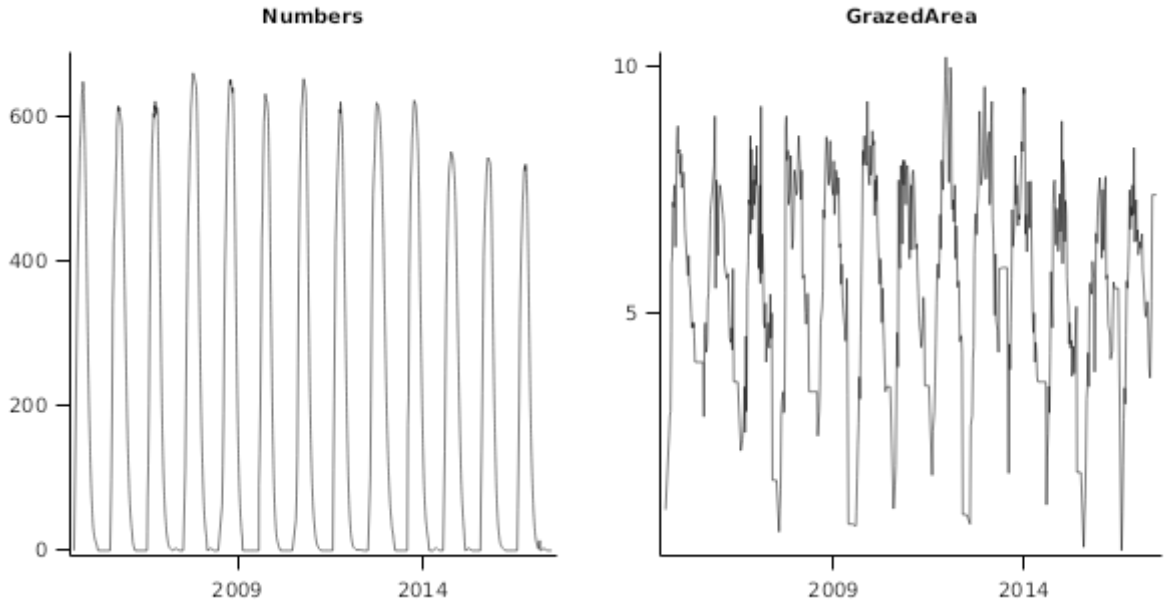
2.3 StockSlurp

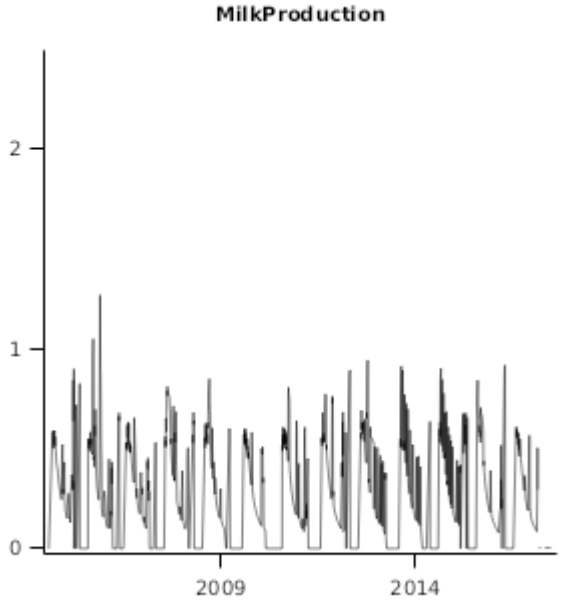
#SLURP simulation

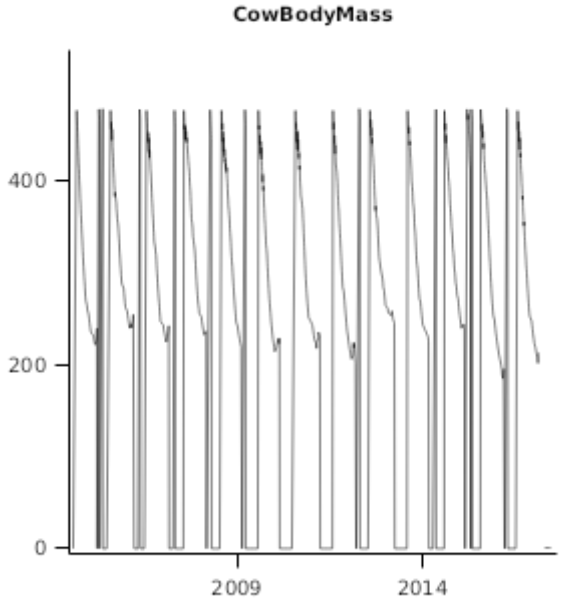
Ideally, to test the Stock model's estimates of production we would have a simulation of the animals rotationally grazing paddocks in which an appropriate pasture was growing. That would be a good reflection of reality but would also add errors in the modelling of plant growth and rotation rules that might obscure the testing of the Stock model.

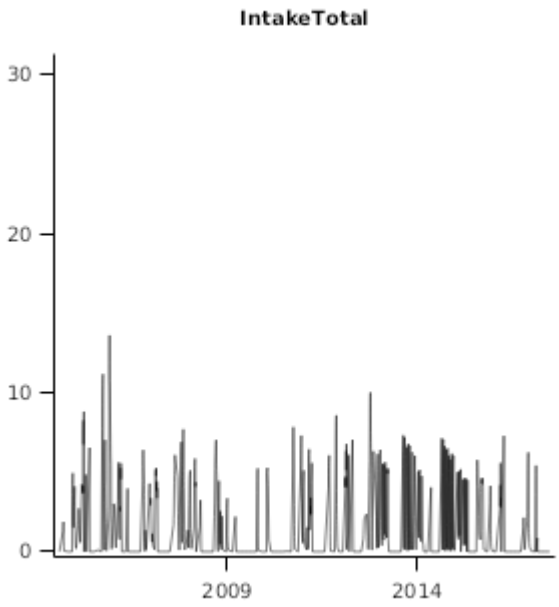
SLURP is a "crop" model that has been built using the Plant Modelling Framework to provide a very simple representation of crops and pastures with user-prescribed growth rather than an internal calculation of growth. The model does not predict crop growth, development or yields - these are supplied by the user. By continually setting the SLURP biomass and pasture quality at the start of the day to the pre-grazing values measured at LUDF, we can use SLURP to test the performance of Stock without adding in other sources of error.

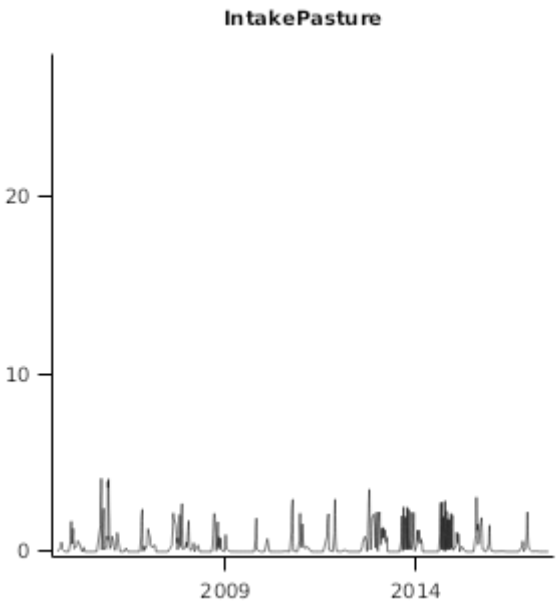
In this simulation, the pre-grazing pasture characteristics of SLURP are set in the SlurpPreGrazingSet component under the Paddock node. Here the relevant Excel file and worksheet are specified; within which PreGrazingCover, GrazedArea, PastureMEConc, PastureDigestibility, and SupplementOffered are specified. In addition to the Stock model itself, at its core the simulation has the ChangeLactatingCowNumber component, Stock Operations and SLURP. Stock Operations continually resets the number of dairy cows with the values taken from the LUDF dataset (see the Validation memo for details). In the Stock Operations, data on actual LUDF weekly changes in cow number have been collated from 1-Aug-2004 to 30-Jun-2017. Positive numbers denote lactating cows have been bought and moved to the feedlot while negative numbers are when cows are dried off and removed from the paddock. It was assumed that cows were bought at weekly intervals. For SLURP, the LUDF dataset is based on weekly reports - the pasture and supplement supplied in the SLURP Excel spreadsheet was extrapolated to the intervening days.

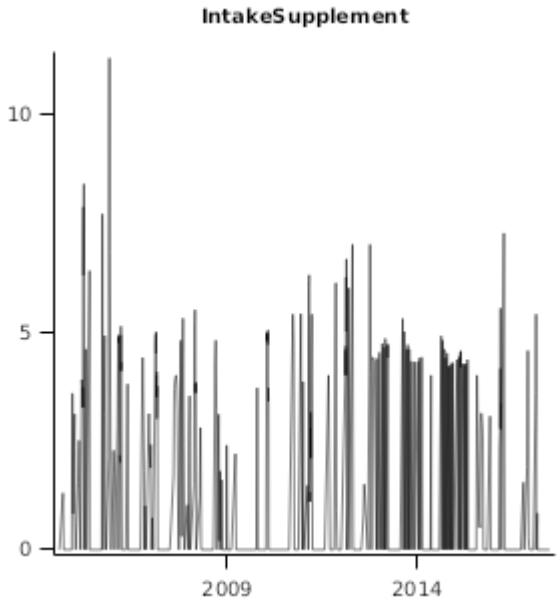












2.4 FeedlotTests

2.4.1 List of experiments

Experiment Name	Design (Number of Treatments)
Hersom1	(3)
Hersom2	(3)
Sharman1	(4)
Sharman2	(4)
Coleman	(2)
Paco	(5)

3 Sensibility

This sensibility test explores dual-purpose wheat in a high rainfall livestock system in south-eastern Australia.

It is based on Sprague et al., 2015

Sheep grazing wheat and fed in feedlot

In this example simulation sheep are bought and sold on specified date. They are fed supplement in a feedlot at a set rate, but graze a wheat crop when crop biomass >= 2.4 t/ha. Sheep are moved from the wheat crop and back to feedlot when crop biomass reaches 0.5 t/ha or crop zadok = 31.

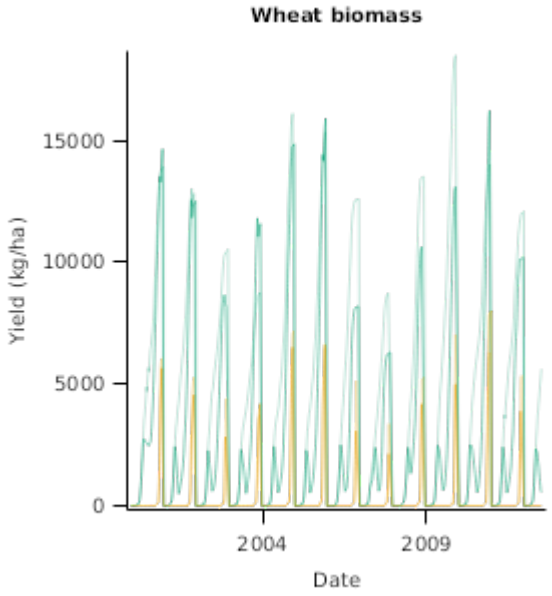
Activities in this manager:

1. Buy animals at start of year & put in feedlot
2. Move animals from feedlot to crop when ready to graze
3. Move animals from crop to feedlot
4. Shear all animals on specified date
5. Sell all animals at end of year

NOTES

1. When the animals are in the feedlot and an animal dies during the day, the supplement has already been fed into the feedlot based on the number of animals in the feedlot at the start of the day. This means the remaining animals have access to slightly more supplement and causes a spike in supp intake graph.
2. When sheep are culled for age + purchased to maintain stocking rate, several groups of sheep are created. This causes irregular amounts of supplement to be fed.

3.1 WheatSheepFeedlot



3.2 TemperatureResponse

3.2.1 List of experiments

Experiment Name	Design (Number of Treatments)
BeefCattleTemperatureResponse	(5)
SheepTemperatureResponse	(5)

4 MassBalanceCheck

4.1 MassBalanceForage

This simulation checks the mass balance of animal / **plant** interactions. The plant model used is a PMF Slurp model that doesn't simulate plant growth. This makes it much easier to check mass balance.

The checks are done in the MassBalanceCalculations manager script.

The script first calculates the amount of pasture removed:

```
PastureRemoved = StartOfDayPasture - EndOfDayPasture;
```

and the weight gain of the animals on a day.

```
LiveWeightGain = EndOfDayAnimal - StartOfDayAnimal;
```

It then calculates two balance terms that should be zero. The first checks that pasture removed = animal intake.

```
LostWt = PastureRemoved.Wt - Intake.Wt;
```

The second checks that N in the pasture removed = excreta N + the N retained by the animal.

```
LostN = PastureRemoved.N - (Excreta.N + LiveWeightGain.N);
```

A check is then made (on day 1 of the simulation) that the faeces from the animal is added to the surface organic matter (som) model:

```
if (!MathUtilities.FloatsAreEqual(animals.FaecesAll.Weight, som.Wt))
      throw new Exception("Mass balance error: The animal faeces weight on day 1 is not
equal to surface organic matter weight.");
```

A check is then made (on day 1 of the simulation) that the urine from the animal is added to the soil urea pool:

```
if (!MathUtilities.FloatsAreEqual(animals.UrineNAll, MathUtilities.Sum(urea.kgha)))
      throw new Exception("Mass balance error: The animal urine on day 1 is not equal to
soil urea amount.");
```

These last checks are only done on day 1 because flows in and out of SOM and Urea pools make it difficult to calculate mass balance. The assumption is that if it works on day 1 it works for all other days.

4.2 MassBalanceFeedlot

This simulation checks the mass balance of animal / **supplement** interactions. The checks are done in the MassBalanceCalculations manager script.

The script first calculates the amount of pasture removed:

```
PastureRemoved = StartOfDayPasture - EndOfDayPasture;
```

and the weight gain of the animals on a day.

```
LiveWeightGain = EndOfDayAnimal - StartOfDayAnimal;
```

It then calculates two balance terms that should be zero. The first checks that supplement removed = animal intake.

```
LostWt = SupplementRemoved.Wt - Intake.Wt;
```

The second checks that N in the pasture removed = excreta N + the N retained by the animal.

```
LostN = PastureRemoved.N - (Excreta.N + LiveWeightGain.N);
```

A check is then made (on day 1 of the simulation) that the faeces from the animal is added to the surface organic matter (som) model:

```
if (!MathUtilities.FloatsAreEqual(animals.FaecesAll.Weight, som.Wt))
      throw new Exception("Mass balance error: The animal faeces weight on day 1 is not
equal to surface organic matter weight.");
```

A check is then made (on day 1 of the simulation) that the urine from the animal is added to the soil urea pool:

```
if (!MathUtilities.FloatsAreEqual(animals.UrineNAll, MathUtilities.Sum(urea.kgha)))
      throw new Exception("Mass balance error: The animal urine on day 1 is not equal to
soil urea amount.");
```

These last checks are only done on day 1 because flows in and out of SOM and Urea pools make it difficult to calculate mass balance. The assumption is that if it works on day 1 it works for all other days.

5 References

M Freer, A.D Moore, J.R Donnelly, 1997. GRAZPLAN: Decision support systems for Australian grazing enterprises—II. The animal biology model for feed intake, production and reproduction and the GrazFeed DSS. Agricultural Systems 54 (1), 77 - 126.

Sprague, S. J., Kirkegaard, J. A., Dove, H., Graham, J. M., McDonald, S. E., Kelman W. M., 2015. Integrating dual-purpose wheat and canola into high-rainfall livestock systems in south-eastern Australia. 1. Crop forage and grain yield. Crop and Pasture Science 66, 365-376.